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 a complex, hierarchical structure. The primary 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 boundaries, cell, row, and column bounding box extraction. The area of fine-grained table structure remains relatively unexplored. This study presents a dataset of 657 tables, manually labeled for cell types and column hierarchy relations. The tables are selected from IBM FinTabNet. The selection of these 657 tables is performed using heuristics, resulting in a much larger proportion, roughly half, of the selected tables having a complex hierarchical structure than a random sample from FinTabNet. Further, we fine-tune models based on LayoutLM on the cell-type classification task and identify hierarchical relations among column headers. We achieve F1 scores of 97% and 73% on the respective tasks. Finally, we use the trained model to create soft labels for the entirety of FinTabNet.

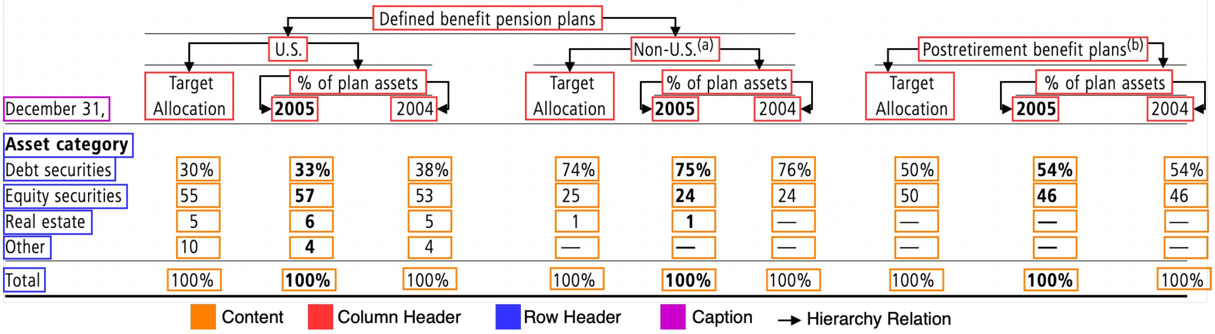
1 Introduction

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.

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 (“Target Allocation”, “% of plan assets”) has its meaning modified by the “parent” cell (“U.S”, “Non-U.S”). The caption “December 31" provides a temporal context to the information in table. 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



December 31,	Defined benefit pension plans						Postretirement benefit plans[b]		
	U.S.			Non-U.S.[a]					
	Target Allocation	% of plan assets		Target Allocation	% of plan assets		Target Allocation	% of plan assets	
		2005	2004		**2005**	2004		**2005**	2004
Asset category									
Debt securities	30%	**33%**	38%	74%	**75%**	76%	50%	**54%**	54%
Equity securities	55	**57**	53	25	**24**	24	50	**46**	46
Real estate	5	**6**	5	1	**1**	—	—	**—**	—
Other	10	**4**	4	—	**—**	—	—	**—**	—
Total	100%	**100%**	100%	100%	**100%**	100%	100%	**100%**	100%

Figure 1: Financial table annotated with fine structure.

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., 2020b), 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 700 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., 2020b) model fine-tuned on the first dataset.

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. 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).

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.

Data & Code: Anonymous version of our manually annotated data is available at https://doi.org/10.5281/zenodo.5855687

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(Paliwal et al., 2019; Prasad et al., 2020; Zheng et al., 2021; Hashmi et al., 2021) 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.

Downstream tasks. 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).

We now focus on the existing work which focuses on understanding the fine grained table structure.

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. (Wang et al., 2021) develops neural representations of tables for use in downstream tasks, but relies on heuristics to elucidate the hierarchical structure as opposed to our approach to classify cell types and identifying hierarchical relationships without using any heuristics.

Hybrid approach. The approach taken in (Sun et al., 2021) to reconstruct table structure uses pretrained networks to embed cells and rules enforced via PSL. (Chi et al., 2019) also use hand crafted features with graph neural networks for predicting the horizontal and vertical relations between cells while we fine-tune all weights of LayoutLM.

Neural Approaches. While there are a few completely neural approaches for 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. 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 cannot be handled by their system, whereas our framework handles such cases naturally.

3 Dataset Creation

This section discusses 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. 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

We annoated 657 tables sourced from IBM FinTabnet (Zheng et al., 2021). 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. Table 1 provides label level information about our annotated dataset.

Table 1: Details of manually annotated dataset.

# of table		657	
# of table with hierarchy		339	
Cell Type	**50th**	**75th**	**100th**
Column Header	4	6	20
Row Header	7	12	63
Content	20	40	241

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., 2020b). We detected the largest consecutive group of numeric values for the heuristic model and marked those as content cells. Cells above and left of the content block are marked as column and row headers. Keyword matching against a hand-curated list is used to detect captions. First column headers are sorted into different levels for heuristic-based hierarchy detection based on the vertical positional information. Then, each cell in level N is assigned a child to the closest cell in level $N-1$.

In the 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. A prediction is done for every polled cell embedding. Column hierarchy prediction is modeled as a binary classification task. Cell embeddings are concatenated and passed onto a non-linear classifier for all possible column header pairs. All models are trained end-to-end.[1]

LayoutLM achieves an F1 score of 96.9 and 72.4 on cell label prediction and relation prediction, respectively. Table 2 shows the complete results for both tasks. Finally, the model creates soft labels for the entire IBM FinTabNet dataset.

4 Discussion

Data Description: In the distribution of four cell type classes, we naturally see an imbalance with the number of content cells as the majority class. As shown in Table 1 the number of content cells per table also varies highly, indicating a variety of table sizes available in our data. Approximately 58% of tables have a caption cell.

About half of the tables in our dataset have column hierarchy present. Though most of these

[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.

Table 2: Baseline Results. H: Column header, R: Row header, C: Content Cell, Ca: Caption.

	Accuracy	**Macro F1**	**Precision**				**Recall**				**F1**			
Cell label prediction			**H**	**R**	**C**	**Ca**	**H**	**R**	**C**	**Ca**	**H**	**R**	**C**	**Ca**
Heuristic	88.5	72.6	50.9	95.6	98.0	40.8	85.9	75.7	93.7	52.9	64.0	84.5	95.8	46.12
BERT	95.0	88.5	82.4	90.3	98.4	85.7	86.1	86.7	98.9	79.5	84.2	88.5	98.7	82.5
LayoutLM	**99.2**	**96.9**	**97.3**	**98.8**	**99.8**	**87.9**	**96.7**	**99.6**	**99.5**	**96.3**	**97.0**	**99.2**	**99.7**	**91.9**
Cell relation prediction			**True**		**False**		**True**		**False**		**True**		**False**	
Heuristic	66.7	65.3	43.9		**92.8**		**87.4**		59.1		**58.4**		72.2	
BERT	80.2	72.4	62.5		85.1		53.8		89.1		57.8		87.0	
LayoutLM	**81.8**	**72.4**	**71.3**		83.8		46.6		**93.6**		56.3		**88.5**	

H: Column header, R: Row header, C: Content Cell, Ca: Caption.

are 2 level hierarchies, about 10% of total tables ($n = 66$) have 3 levels of column headers. The maximum height of column hierarchy in our dataset is 4, including complex examples of nested hierarchies as shown in Figure 1.

Challenges: Our heuristics perform well in detecting row headers and content cells but struggle with some column header and caption detection aspects. Precision for column header detection is low due to non-numeric tables. In the case of non-numeric tables, many content cells get marked as column headers leading to low precision. Poor performance of caption detection can be attributed to limitations of keywords list and false positives inherent to text matching. Simple rules assume that every cell on level N must have a parent on level $N-1$, which is not valid for complex tables. Hence hierarchy detection using heuristics gives low precision for the positive class and low recall to the negative category. Such effect is further boosted by trickled down errors from cell label detection algorithm. Though, these rules work well if a hierarchy exists, as indicated by the high recall of positive class.

BERT improves the performance of cell labeling tasks, especially in the case of non-numeric tables. The presence of textual context helps in differentiating between headers and content cells. However, the class level performance for hierarchy detection suffers from the model being biased towards negative class due to class imbalance. This is expected since BERT does not account for positional information, essential for hierarchy prediction tasks.

Adopting a positionally and contextually aware model like LayoutLM improves cell labeling performance. Our manual inspection revealed that a few errors still present are caused by minority tables in which differentiation between column header, captions, and top row headers is done using changes in fonts rather than positions. Shifting to a more visually aware architecture like LayoutLMv2 (Xu et al., 2020a) may help in improving performance for such cases. LayoutLM performance is much better than heuristics/BERT on hierarchy detection tasks. However, significant room for improvement is still available. It is common to have textually same and positionally close hierarchy pairs in complex financial tables. We observed in such cases the probability of LayoutLM predicting a false parent-child couple as true is high. Further, since each possible pair of hierarchy is fed independently to the model, at times, a single cell is assigned multiple parents, which leads to poor performance. These concerns can be addressed using rule-based post-processing and having models aware of both global and relative positional context.

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 to improve on the LayoutLM-based baseline.

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*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. 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)*.