A Comparison of Image-based, Text-based and Multimodal Models in the Table Structure Recognition Task

Anonymous ACL submission

Abstract

Table Structure Recognition (TSR) aims to convert table images into machine readable formats such as HTML. The latest approach uses image-encoder-text-decoder model, in which image encoder extracts image features and a text decoder generates HTML tokens. Furthermore, a new approach uses multimodal-encoder, in which encoder extracts textual and visual features, and outperforms other image-encoder models. However, these models have not been compared under the same conditions. Given this background, it is necessary for future development of TSR to investigate the effects of image and text features on the TSR. In this research, we constructed an encoder-decoder model and used three different encoders: image-based, text-based, and multimodal. By comparing the TSR scores, we evaluated which model performs better. Experimental results suggested that an image-based approach is the most effective.

1 Introduction

Table Structure Recognition (TSR) is the task of extracting table structural elements (rows, columns, headers) from a table image and converting them into the corresponding HTML. Since tables appear in various media such as scientific papers, websites, and newspapers, analyzing tables by TSR is important for managing large amounts of documents (Hiroyuki Oka, 2021). Early research on TSR (Hassan and Baumgartner, 2007; Oro and Ruffolo, 2009) analyzed tables using rule-based methods, but in recent years, various TSR models have adopted methods of deep learning. Among the many models, the most popular is an image-to-text model (Nassar et al., 2022; Ye et al., 2021; Zhong et al., 2020; Li et al., 2022). These consist of an image encoder and a text decoder, and the image encoder extracts features and the text decoder generates HTML tags. On the other hand, a model (Chen et al., 2023) has emerged that consists of

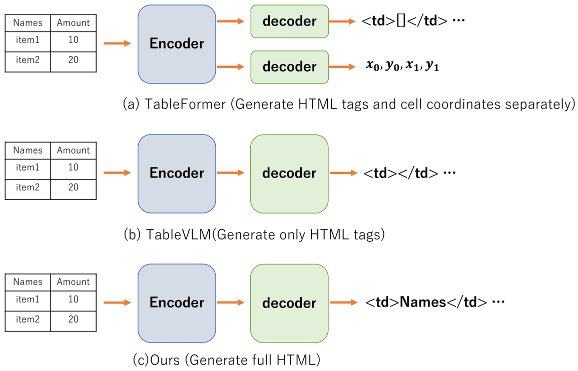

Figure 1: Comparing with other methods. (a) is encoder-dual-decoder models that generate HTML tags and cell coordinates. (b) is encoder-decoder models that generate HTML tag only. (c) is encoder-decoder models that generate full HTML.

a multimodal encoder, which takes both images and text as input and outperforms other image-encoder models. However, previous works are not comparing under the same experimental conditions considering differences with structures and generation methods. For example, Figure 1 (a) is an image-based Tableformer which consists of image-encoder-dual-decoder and outputs the HTML tags and its bounding boxes separately, while Figure 1 (b) is a multimodal TableVLM which consists of multimodal-encoder-single-decoder and outputs only HTML tags. Thus, it is necessary to further explore the optimal methodology for the TSR task in terms of generation method and modality.

In this research, we propose a method of generating complete HTML, which contain tags and cell texts as shown in Figure 1 (c). Under this condition, we analyze which model is superior by comparing the accuracy on the benchmark for TSR among three models: text-based, image-based, and multimodal models. Our contributions are summarized as follows:

- A method of generating complete HTML (tags and cell contents) is better than other methods

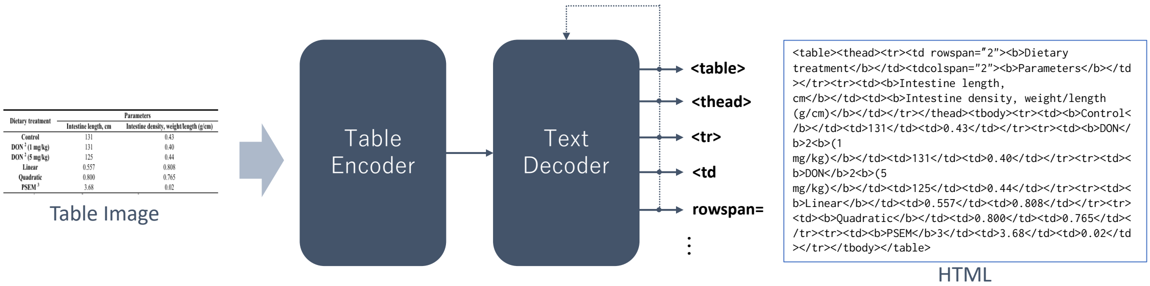

Figure 2: Model Architecture is simple encoder-decoder model that generates HTML from a table image. The encoder outputs a latent representation of the table and the decoder generates HTML tokens autoregressively.

of generation.

- Image-based model has the best performance in situations where large amounts of data are available.
- Text-based and multimodal models are efficient in terms of data and can provide accuracy even with a small amount of data.

2 Methodology

We evaluate and compare image-based, text-based, and multimodal (combining text with image) models on TSR datasets.

Model Architecture We use a simple encoder-decoder model, as shown in Figure 2. We use BART-decoder as a text decoder, LayoutLMv3-L as a text-based encoder, Swin Transformer as an image-based encoder, and LayoutLMv3 as a multimodal encoder.

Swin Transformer Encoder Swin Transformer (Liu et al., 2021) is an image-based model. Swin Transformer converts the table image $x \in \mathbb{R}^{(3 \times W_0 \times H_0)}$ into a fixed rectangle $(3, H, W)$. The transformed image is divided into patches and are input into model. The input patches are merged repeatedly and finally converted into a latent representation $z \in \mathbb{R}^{(N \times d)}$, where N is the final number of patches, d is the dimension of the latent representation.

LayoutLMv3 Encoder LayoutLMv3 (Huang et al., 2022) is a multimodal model which handles text, images, and coordinates. LayoutLMv3 receives tokens $t_i (0 \leq i < L)$ that have been split by WordPiece (Wu et al., 2016) from text obtained via OCR from table images, their bounding boxes $b_i \in (x_0, y_0, x_1, y_1)(0 \leq i < L)$, and the table image $x \in \mathbb{R}^{3 \times W_0 \times H_0}$ transformed into a fixed size $(3, H, W)$. The model captures layout relationships and finally outputs a latent representation of each tokens and image $z \in \mathbb{R}^{(L+N \times d)}$.

LayoutLMv3-L Encoder LayoutLMv3-L is a text-based model that handles text and coordinates. The difference with LayoutLMv3-L and LayoutLMv3 is not using image as input. In other words, the model receives tokens $t_i (0 \leq i < L)$ that have been split by WordPiece from text obtained via OCR, their bounding boxes $b_i \in (x_0, y_0, x_1, y_1)(0 \leq i < L)$ only. Therefore, the model finally outputs $z \in \mathbb{R}^{(L,d)}$.

BART Decoder BART decoder (Lewis et al., 2020) receives the latent representation z obtained from the encoder and decode z into corresponding HTML tokens. The decoder generates HTML tokens autoregressively using the itself Self-Attention and Cross-Attention.

3 Experiments

Datasets We use two datasets in our research: PubTabNet (Zhong et al., 2020) which contains 509K tables from scientific papers, and FinTabNet (Zheng et al., 2020), which contains 112K tables derived from annual reports of S&P 500 companies. Both datasets contain HTML corresponding to table image. The PubTabNet dataset is divided into 97% for training and 3% for validation, while FinTabNet is allocated to 81% for training, 9.5% for validation, and 9.5% for testing.

Evaluation Metric We evaluate the generated HTML by Tree-Edit-Distance-Similarity (TEDS) (Zhong et al., 2020). TEDS is given by the following formula.

$$\text{TEDS}(T_a, T_b) = 1 - \frac{\text{EditDist}(T_a, T_b)}{\max(|T_a|, |T_b|)} \quad (1)$$

T_a and T_b represent the HTML tree structure, and $EditDist()$ calculates the edit distance between

			FinTabNet		PubTabNet	
Model	Modality	OCR	TEDS-Struc(%)	TEDS(%)	TEDS-Struc(%)	TEDS(%)
TableFormer (Nassar et al., 2022)	V	✓	96.80	-	**96.75**	93.60
Swin Transformer-BART	V	-	95.60	88.93	96.29	**95.12**
PaddleOCR + LayoutLMv3-L-BART	L	✓	97.21	94.77	95.06	90.80
TesseractOCR + LayoutLMv3-L-BART	L	✓	95.97	91.79	93.50	83.62
PaddleOCR + LayoutLMv3-BART	VL	✓	**97.56**	**95.23**	96.25	93.69
TesseractOCR + LayoutLMv3-BART	VL	✓	95.72	91.59	95.59	91.32

Table 1: The TEDS on FinTabNet and PubTabNet.

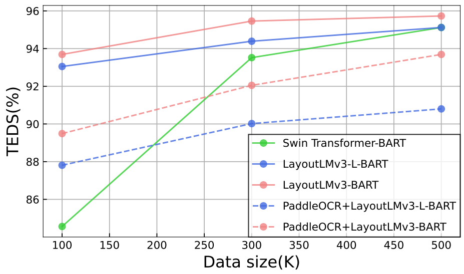

Figure 3: The TEDS when changing the number of training data in PubTabNet.

Train data	TEDS-Struc(%)	TEDS(%)
FinTabNet	95.60	88.93
FinTabNet+PubTabNet	**97.06**	**95.95**

Table 2: The TEDS of Swin Transformer-BART on FinTabNet when training data size increase.

the two tree structures. Also, $|T|$ represents the number of nodes in T. We also evaluate by TEDS-Struc, which ignore cell content and only consider logical structure of HTML as T.

Implementation Details We chose Swin Transformer with image size $(H, W) = (448, 896)$ as inputs, window size=7, layers [2, 2, 14, 2] and number of parameters $77M$. We also use LayoutLMv3 encoder that consist of a 6-layer model with image size $(H, W) = (224, 224)$ as inputs, $d = 768$, maximum sequence length $L = 512$, and number of parameters $83M$. Also, We set the LayoutLMv3-L encoder in the same way as the LayoutLMv3 encoder and this parameters is $82M$. Note that the number of parameters was set close to each other in order to compare the three models. We use BART decoder that consist of 4-layer, with $d = 1024$ and $L = 1024$. Each model was initialized with pre-trained weights. The model was trained using the AdamW (Loshchilov and Hutter, 2019) optimization method, with a learning rate of 0.0001, a weight decay of 0.02, and $(\beta_1, \beta_2) = (0.9, 0.99)$. The batch size was set to 192, and the training was conducted over 20 epochs. Additionally, there was a warm-up period covering 5% of the total training duration, during which the learning rate was linearly increased to 0.0001. Furthermore, We truncate the sequence of HTML and inputs over maximum length L. As inputs to LayoutLMv3 and LayoutLMv3-L, We use PaddleOCR[1] and TesseractOCR[2]. During inference the HTML tokens is generated using greedy search.

4 Results and Discussion

Image-based vs Text-Based vs Multimodal As shown in Table 1, TableFormer (Nassar et al., 2022) was added as baselines (Figure 1 (a)). This model has an image-encoder-dual-decoder structure, and the two decoders output HTML tags and cell bounding boxes. Finally, We obtain HTML by extracting cell texts from generated cell bounding boxes. The TEDS of Swin Transformer-BART achieved 1.5% increase over baseline on PubTabNet. This suggests that our approach that generates complete HTML is better than generating cell coordinates and later obtaining the cell texts by a separate OCR. Next, comparing the overall results, Swin Transformer-BART has the highest TEDS in PubTabNet. On the other hand, in FinTabNet Swin Transformer-BART has the lowest TEDS, while PaddleOCR+LayoutLMv3-BART has the highest TEDS. We believe that this is due to the difference in the number of training data between FinTabNet and PubTabNet. Figure 3 shows the TEDS of each model when the train data size of PubTabNet is changed. This shows that Swin Transformer-BART has low TEDS when the amount of data is small. On the other hand, when the number of data is increased, the TEDS becomes about the same as other models. The trend indicates that image-based models require a lot of training data. In contrast, text-

[1] `https://github.com/PaddlePaddle/PaddleOCR`
[2] `https://github.com/tesseract-ocr/tesseract`

Model	Modality	FinTabNet TEDS-Struc(%)	FinTabNet TEDS(%)	PubTabNet TEDS-Struc(%)	PubTabNet TEDS(%)
TableVLM (Chen et al., 2023)	VL	-	-	96.92	-
LayoutLMv3-L-BART	L	98.34	97.31	96.82	95.12
LayoutLMv3-BART	VL	**98.60**	**97.65**	**97.11**	**95.73**

Table 3: The evaluation in TEDS when these models receive the cell texts and its bounding boxes obtained from annotations, not using OCR. This represents the performance of the model under the condition of using an OCR with 100% accuracy.

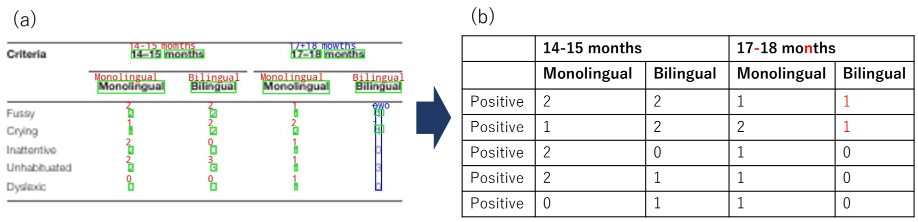

Figure 4: Case Study: (a) displays the texts and bounding boxes obtained by TesseractOCR. (b) shows the table generated by LayoutLMv3-BART, which receives the output from TesseractOCR (a).

based and multimodal models are efficient in terms of data. Threforer, we carried out additional evaluation when Swin Transformer-BART was trained with PubTabNet and then finetune with FinTabNet as shown in Table 2. Increasing training data yields a notable improvement of 7% TEDS and 1.4% TEDS-struc. Comparing the results in Table 1 and Table 2, It can be seen that when there is a large amount of training data, Swin Transformer-BART has highest TEDS in FinTabNet and PubTabNet. This results suggest that image-based approaches are most suitable because large-scale data is easily available in recent years.

The TEDS of text-based and multimodal models when inputs is perfect Table 3 shows the TEDS when using the cell texts and the bounding boxes obtained from the annotations. This represents the performance of the model in an ideal situation when using an OCR with an accuracy of 100%. TableVLM (Chen et al., 2023) has a similar structure to LayoutLMv3-BART, but only generates HTML tags. LayoutLMv3-BART outperforms TableVLM by improving 0.2% TEDS-Struc on PubTabNet. This suggests that generating full HTML is better than generating only HTML tags. Comparing the results of Table 1 and Table 3, LayoutLMv3-BART and LayoutLMv3-L-BART using perfect inputs also show better TEDS and TEDS-Struc than when using PaddleOCR or TesseractOCR as inputs. Furthermore, both models outperform Swin Transformer-BART. Therefore, multimodal or text-based model would be better in an environment where very accurate OCR is available, but it is currently difficult to obtain OCR with such high accuracy, suggesting that image-based solutions is still the better choice.

Case Study Figure 4 shows the characters and coordinates obtained from TesseractOCR, and (b) shows the outputs by LayoutLMv3-BART that receives them. As shown in (a), the text obtained from TesseractOCR not only contains errors of characters, but also undetected characters and incorrect bounding boxes. However, even after inputting these, a somewhat correct table is generated. This may be because the model corrects errors internally or maintains rules for the table structure. Therefore, it can be seen that the method of generating complete HTML is better than obtaining the cell texts later using OCR, as shown in Figure 1.

5 Conclusion

In this study, we constructed an encoder-decoder model that generates complete HTML with a single decoder in order to solve the TSR task. Under this condition, we analyze which model is superior by comparing the accuracy on the benchmark for TSR among three models: text-based, image-based, and multimodal models. As a result, an image-based approach is suitable for this task. It is also suggested that the method that generates complete HTML is superior to other generation methods.

6 Limitations

We use only two open-source OCR, not paid OCR that are highly accurate. Therefore, we need to research the detailed differences in performance in the TSR task, using various OCR. Furthermore, the approach of generating full HTML leads to extremely long sequence lengths and has limitations for large tables or tables with many characters.

References

Leiyuan Chen, Chengsong Huang, Xiaoqing Zheng, Jinshu Lin, and Xuanjing Huang. 2023. TableVLM: Multi-modal pre-training for table structure recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2437–2449, Toronto, Canada. Association for Computational Linguistics.

T. Hassan and R. Baumgartner. 2007. Table recognition and understanding from pdf files. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1143–1147.

Hiroyuki Shindo Yuji Matsumoto Masashi Ishii Hiroyuki Oka, Atsushi Yoshizawa. 2021. Machine extraction of polymer data from tables using xml versions of scientific articles, science and technology of advanced materials: Methods. 1.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chenxia Li, Ruoyu Guo, Jun Zhou, Mengtao An, Yuning Du, Lingfeng Zhu, Yi Liu, Xiaoguang Hu, and Dianhai Yu. 2022. Pp-structurev2: A stronger document analysis system.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter Staar. 2022. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4614–4623.

Ermelinda Oro and Massimo Ruffolo. 2009. Pdf-trex: An approach for recognizing and extracting tables from pdf documents. In *2009 10th International Conference on Document Analysis and Recognition*, pages 906–910.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

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.

Xinyi Zheng, Doug Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 2020. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation.

			FinTabNet†		PubTabNet	
Image size	Parameters	Window size	TEDS-Struc(%)	TEDS(%)	TEDS-Struc(%)	TEDS(%)
(448, 896)	77M	7	97.06	95.95	96.29	95.12
(864, 864)	82M	9	**98.24**	**97.51**	**96.67**	**95.77**

† Evaluation when the model was trained with PubTabNet and then finetune with FinTabNet.

Table 4: The TEDS of Swin Transformer-BART that handles different resolutions.

			FinTabNet		PubTabNet	
Model	Modality	OCR	TEDS-Struc(%)	TEDS(%)	TEDS-Struc(%)	TEDS(%)
TableFormer (Nassar et al., 2022)	V	✓	96.80	-	**96.75**	93.60
Swin Transformer-BART (448, 896)	V	-	97.06†	95.95†	96.29	95.12
Swin Transformer-BART (864, 864)	V	-	**98.24**†	**97.51**†	96.67	**95.77**
PaddleOCR + LayoutLMv3-L-BART	L	✓	97.21	94.77	95.06	90.80
TesseractOCR + LayoutLMv3-L-BART	L	✓	95.97	91.79	93.50	83.62
PaddleOCR + LayoutLMv3-BART	VL	✓	97.56	95.23	96.25	93.69
TesseractOCR + LayoutLMv3-BART	VL	✓	95.72	91.59	95.59	91.32
TableVLM (Chen et al., 2023)	VL	-‡	-	-	96.92	-
LayoutLMv3-L-BART	L	-‡	98.34	97.31	96.82	95.12
LayoutLMv3-BART	VL	-‡	**98.6**	**97.65**	**97.11**	95.73

† Evaluation when Swin Transformer-BART was trained with PubTabNet and then finetune with FinTabNet.
‡ Using cell texts and bounding boxes from annotations, not OCR.

Table 5: The all results.

A Additional Results and Discussion

A.1 TEDS of Swin Transformer-BART when input size change.

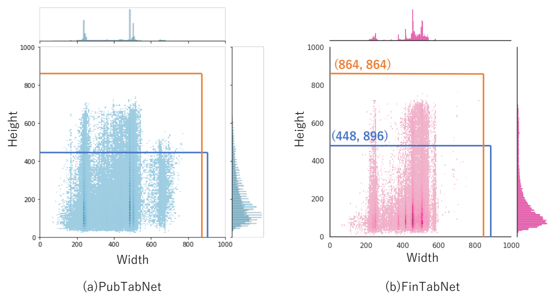

Figure 5: The distribution of the image size on PubTabNet (left) and FinTabNet (right).

Table 4 shows the TEDS of Swin Transformer-BART that handles different resolutions. Figure 5 is also a scatter plot of the resolution of table images of two datasets. Swin Transformer-BART(864, 864) outperforms Swin Transformer-BART(448, 896) on FinTabNet and PubTabNet. The improvement in score suggests that it is necessary to set the input size of model based on the original image size, as shown in Figure 5.

A.2 All results

Table 5 summarizes all the results. Swin Transformer-BART(864, 864) outperforms other models on PubTabnet and FinTabNet. Furthermore, Swin Transformer-BART(864, 864) outperforms or matches LayoutLMv3-BART and LayoutLMv3-L-BART which both receive complete cell texts and bounding boxes from annotations. Therefore, these results indicate that an image-based approach is most suitable for TSR.

A.3 Comparison of model inference speeds

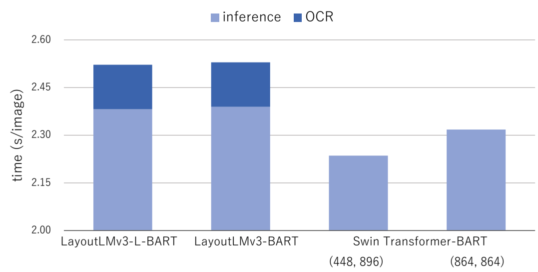

Figure 6: Comparison of model inference speeds. In the chart, light blue represents the inference speed of the model itself, while blue indicates the speed of Paddle OCR.

As shown in Figure 6, the inference speed of Swin Transformer-BART outperforms LayoutLMv3-L-BART and LayoutLMv3-BART. Thus, an image-based model is better than text-based and multi-modal models in terms of the inference speed.

B Licences

Name	License
Tesseract OCR	Apache-2.0
Paddle OCR	Apache-2.0
FinTabNet	CDLA-Permissiv-1.0
PubTabNet	CDLA-Permissive-1.0

Table 6: The licenses of used tools and datasets.