Position Masking for Improved Layout-Aware Document Understanding

Anik Saha
Rensselaer Polytechnic Institute
Troy, NY, USA
sahaa@rpi.edu

Catherine Finegan-Dollak
IBM Research
Yorktown Heights, NY, USA
cfd@ibm.com

Ashish Verma
IBM Research
Yorktown Heights, NY, USA
ashish.verma1@ibm.com

ABSTRACT

Natural language processing for document scans and PDFs has the potential to enormously improve the efficiency of business processes. Layout-aware word embeddings such as LayoutLM [17] have shown promise for classification of and information extraction from such documents. This paper proposes a new pre-training task called *position masking* that can improve performance of layout-aware word embeddings that incorporate 2-D position embeddings. We compare models pre-trained with only language masking against models pre-trained with both language masking and position masking, and we find that position masking improves performance by over 5% on a form understanding task.

KEYWORDS

structured document understanding, pre-trained language model, position embedding

1 INTRODUCTION

For many real-world documents, layout—how the text is positioned on a two-dimensional page—is essential to meaning. For example, the layout of addresses on an envelop tells us who is the sender and who the receiver. Similarly, in the invoice in Figure 1, a human can easily tell from the layout that the total due is $8.75. Recent systems that combine information about document layout with natural language processing techniques have shown improved performance on tasks such as information extraction [1, 7, 11, 17]. Such techniques hold great promise for many business processes that must be performed on scanned or native PDF formatted document images.

Information extraction (IE) from forms is essential to many business processes, from identifying the amount due on an invoice to identifying the shipping address on a purchase order to getting the name of a borrower on a loan application. Text-based IE approaches [13] that are successful in other domains, such as news articles and Wikipedia, don't take advantage of the essential nature of forms: keys and their values are often spatially grouped. Systems such as LayoutLM [17] that leverage this information have an advantage over text-only systems like BERT [2].

This paper's contribution is a new pre-training task for layout-aware text embeddings: position masking. A masked language model (MLM) pre-training task, like that used by LayoutLM [17], masks token embeddings and tries to predict the correct token based on context—the unmasked components of the token, such as its position, and the other tokens in the input sequence. Our model additionally masks 2-D position embeddings and attempts to predict the correct position based on context. We show that adding the position-masking pre-training task to LayoutLM results in an absolute improvement of 5% on a form-understanding task. The proposed pre-training task would be compatible with any text embeddings that incorporate 2-D position embeddings, such as the new LayoutLMv2 [18].

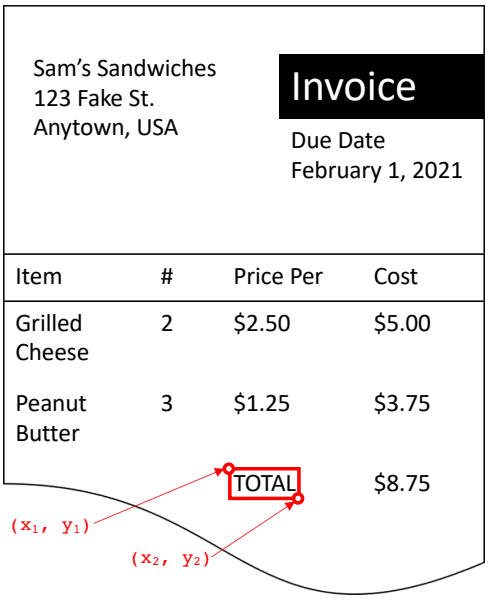

Figure 1: An example invoice, with one bounding box and coordinates shown. A human can use the positions of text to help determine the total cost.

2 RELATED WORK

Several approaches have recently emerged for layout-aware document representations, particularly aimed at information extraction (IE). [5] engineered a token representation that included word shape and positional information. [12]'s token representation incorporated a 2-D position embedding for both the token itself and a neighborhood of nearby tokens. [16] sought to extract a hierarchical structure among text fragments on form pages using a feature fusion model to combine text, layout, and visual features. [10] and [11] both constructed document graphs where nodes were text boxes and edge embeddings incorporated spatial information.

Some works transform a 2-D page into a 3-D tensor. They assign each pixel on a page a vector , generating a $vectorsize \times height \times$

width tensor. This representation is then passed to a CNN encoder-decoder model for closed-class IE. In Chargrid [7], each character is assigned a one-hot encoding. BERTgrid [1] replaces the one-hot character vectors with BERT embeddings for tokens. [8] incorporated the RGB values for pixels into Chargrid. A downside to these techniques is the size of the tensors that result, making scalability a problem.

LayoutLM [17] is a transformer-based MLM that adds 2-D position embeddings to the input text embeddings for a BERT-like model. Our work is complementary to LayoutLM, adding a new pre-training task to enhance its performance. In work concurrent with ours, [18] improved on LayoutLM by using a multi-modal transformer and adding new pre-training tasks. Our technique could also be combined with theirs.

[15] compares learned position embeddings from BERT, RoBERTa, and GPT-2, with sinusoidal position embeddings. Its focus was on 1-D position embeddings, though; to our knowledge, no similar study has been conducted on 2-D position embeddings.

3 APPROACH

Our approach, illustrated in Figure 2, builds on LayoutLM [17], which we briefly review before describing the proposed improvement.

3.1 Background: LayoutLM

LayoutLM passes input text embeddings through transformer layers. Each of LayoutLM's input text embeddings is the sum of a token embedding, a 1-D position embedding, a segment embedding,[1] and 2-D position embeddings.

More formally, suppose we have a vocabulary V and a document image D comprised of tokens $t_0, t_1 ... t_N$ ($t_i \in V$) with bounding boxes $b_0, b_1 ... b_N$ and segments $s_0, s_1 ... s_N$. Token t_i's 1-D position is i—that is, its position in the input sequence. Bounding box b_i is defined by its top left corner at (x_1^i, y_1^i) and its bottom right corner at (x_2^i, y_2^i), as illustrated in Figure 1. Its width is $w_i = x_2^i - x_1^i$ and height is $h_i = y_2^i - y_1^i$. Let $\mathrm{E}(x)$ be an embedding function, with the subscript t for token embeddings, p for 1-D position embeddings, s for sequence embeddings, x for x-coordinate embeddings, y for y-coordinate embeddings, w for width embeddings, and h for height embeddings. The input text embedding for the i-th token is

$$\begin{aligned} E_i = \mathrm{E}_t(t_i) + \mathrm{E}_p(i) + \mathrm{E}_s(s_i) + \mathrm{E}_x(x_1^i) + \mathrm{E}_y(y_1^i) \\ + \mathrm{E}_x(x_2^i) + \mathrm{E}_y(y_2^i) + \mathrm{E}_w(w_i) + \mathrm{E}_h(h_i) \end{aligned} \tag{1}$$

LayoutLM uses a masked language model (MLM) pre-training task. In the MLM task, for some randomly selected subset of input tokens J, for each $j \in J$, t_j is replaced by `[MASK]`, so the first term of Equation 1 becomes $\mathrm{E}_t(\texttt{[MASK]})$. The token is the only component of the input text embedding that is masked, as illustrated in Figure 2(b). The transformer model is trained to use context—all of the unmasked components of the embedding E_j and the surrounding embeddings (E_i for all $i \neq j$)—to predict the missing token, $\hat{t}_j \in V$. MLM loss is

$$\mathcal{L}_{MLM} = \sum_{j \in J} \mathrm{CrossEntropy}(t_j, \hat{t}_j) \tag{2}$$

3.2 Position Masking

This work introduces *position masking*, as shown in Figure 2. We replace a fraction of *positions* with the coordinate assigned to `[MASK]` (the maximum x and y coordinate in the 2-D matrix) and train the model to predict what the true position was. Similar to the MLM task, the model bases its predictions on the context. Predicting the 2-D position of a token on a page will, we believe, force the model to better learn the relationship of layout and text.

We randomly choose a set K of tokens for position masking. Since J (the set of masked *tokens*) and K are selected randomly and independently, a small fraction may overlap, as in Figure 2(d). In these cases, the model is forced to rely more heavily on the surrounding text for context, rather than on unmasked components of the same input text embedding. This encourages the model to learn inter-token relationships of layout and text.

Since bounding boxes consist of multiple elements, position masking may be full or partial. Full position masking (Figure 2(c)) selects a fraction of tokens and replaces all the coordinates with their `[MASK]` values. Partial position masking (Figure 2(d)-(f)) replaces only some coordinates.

Position masking can be framed as either regression or classification. As a classification problem, the labels are the integers from $[0, m]$, where m is the maximum value of a pixel on the axis being predicted. Classification has the advantage of more closely mirroring MLM. However, regression may be more logical where we are trying to predict a measurable value rather than a categorical one. We define $g(x, \hat{x})$ to be a cross entropy loss function when we use a classifier and a smooth-L1 loss [3] when we use regression. If we mask only x_1, then position-masking loss is

$$\mathcal{L}_{PM} = \sum_{k \in K} g(x_k^1, \hat{x}_k^1) \tag{3}$$

When we do full position masking, we average the equivalent losses for all position embeddings.

We train the MLM and masked position model simultaneously, using the loss function

$$\mathcal{L} = \mathcal{L}_{MLM} + \lambda \mathcal{L}_{PM} \tag{4}$$

where λ is a hyperparameter to weight the position masking loss.

4 EXPERIMENTS

4.1 Systems

We compare LayoutLM with no position masking against LayoutLM with several varieties of position masking. We used the publicly available implementation of LayoutLM.[2] All experiments use the BASE-sized model with the Masked Visual-Language Model (MVLM) loss but not Multi-label Document Classification (MDC) loss function.

Our position masking variations are as follows: PosMaskx1 uses partial position masking of the x_1 embedding, while PosMaskFull

[1] BERT uses segment IDs when the input text includes two parts; *e.g.*, for detecting semantic similarity between two sentences, the segment marks which tokens are part of which sentence. Here, all segment IDs are 0.

[2] https://github.com/microsoft/unilm/tree/master/layoutlm. We implemented the pre-training script ourselves since the public release did not include it.

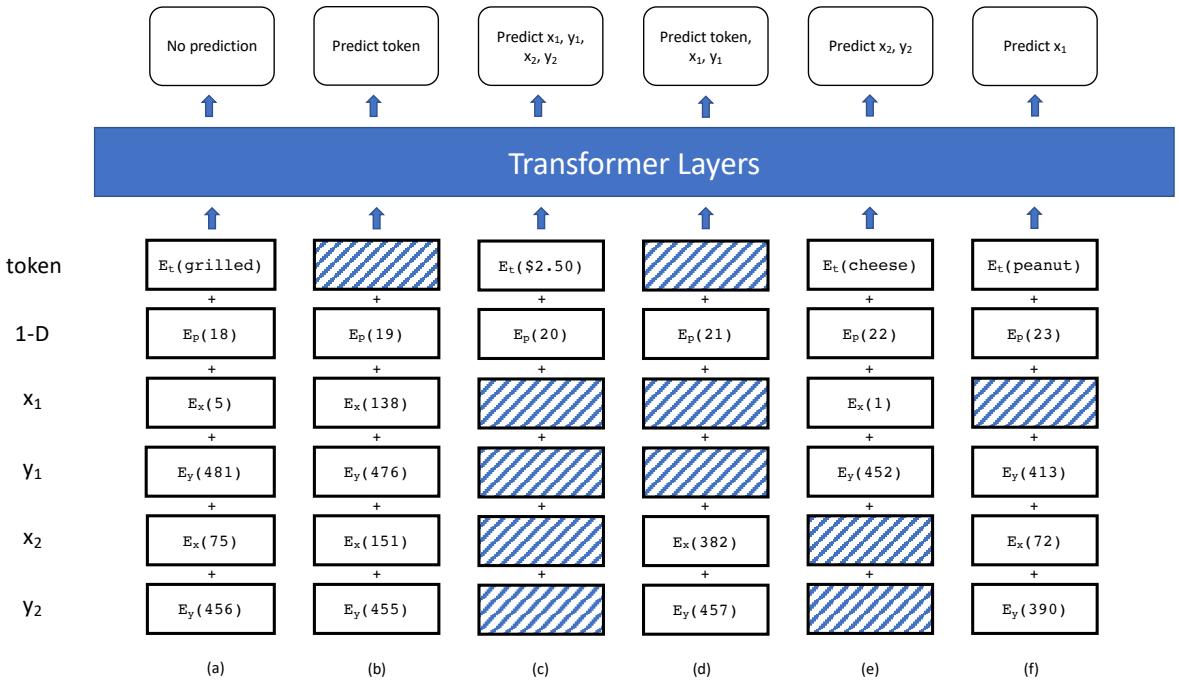

Figure 2: Position masking and language masking applied to a subset of tokens from the invoice in Figure 1. Shaded boxes indicate embeddings that have been masked. Segment embeddings omitted for brevity. Column a: No masking applied. Column b: Token masking only; can be used in BERT, LayoutLM, or our model. Column c: Full position masking. Column d: Partial position masking of x_1 and y_1 and token masking. Column e: Partial position masking of x_2 and y_2. Column f: Partial position masking of x_1 only.

masks all four coordinate embeddings. CELoss trains a classifier and RegLoss a regressor.

Position masking carries a risk that the model will learn the relationship of coordinates to height and width, rather than of positions to tokens. We therefore excluded the height and width embeddings from the PosMask conditions.

4.2 Pre-training

We pre-train all models on 500,000 document images from tobacco litigation settlement documents.[3] Three hundred twenty thousand of these are from the train set of RVL-CDIP[4] [4], with labels removed. We collected the remaining 180,000 from the University of California San Francisco Library's Truth Tobacco Industry Documents.[5] We use Tesseract OCR v.4.1[6] to obtain text and bounding boxes for all scanned document images. Additional pre-training settings are in Appendix B.

4.3 Evaluation

We compare the performance of the pre-trained systems on the downstream task of form understanding. Form understanding aims to identify keys (sometimes called questions) and values (sometimes called answers) on forms. For instance, in Figure 1, "Due Date" is a key with the value "February 1, 2021." We evaluate on the FUNSD dataset[7] [6]. It includes 199 (149 train/50 test) scans of forms, with gold labels for keys, values, headers, and "other" entity types.

We fine tune for 100 epochs on the train set, then predict an entity tag for each token in the test set. We report precision, recall, and F_1 for entity identification. Following [17], we leave linking for future work. We report the mean over 5 fine-tuning runs for each system. We use a one-way ANOVA and a Tukey test to check statistical significance.

[3]The original LayoutLM paper pre-trained on tobacco litigation documents from IIT-CDIP [9] IIT-CDIP was unavailable to the public during much of 2020-21 (See discussions such as https://github.com/microsoft/unilm/issues/250), so we collected comparable documents. Moreover, in light of concerns about the environmental impact of enormous experiments [14], and since the aim of the current work is to compare models with and without position masking, rather than to exceed SOTA, we do not use 11 million pages, as LayoutLM's largest models did. Our pre-training size of 500,000 pages was the smallest pre-training size reported in [17], which outperformed the corresponding BERT model.

[4]https://www.cs.cmu.edu/~aharley/rvl-cdip/

[5]https://www.industrydocuments.ucsf.edu/tobacco/ Details of our collection method are in Appendix A. The IDs and page numbers for document images included in our dataset are available at https://github.com/aniksh/tobacco_documents.

[6]https://github.com/tesseract-ocr/tesseract

[7]https://guillaumejaume.github.io/FUNSD/

Model	Precision	Recall	F1
LayoutLM	61.1 (0.8)	69.2 (0.2)	64.9 (0.5)
PosMaskx1			
with CE Loss	66.8 (0.6)	**74.0** (0.5)	**70.2** (0.4)
with Reg Loss	65.5 (1.0)	73.6 (0.6)	69.3 (0.8)
PosMaskFull			
with CE Loss	**67.1** (1.0)	73.6 (0.6)	**70.2** (0.8)
with Reg Loss	64.9 (1.0)	72.9 (1.0)	68.7 (0.9)

Table 1: Mean (standard deviation) Precision, Recall, and F1 scores on FUNSD. PosMaskx1: partial position masking of the x1 coordinate. PosMaskFull: full position masking. CE Loss: cross entropy. Reg Loss: regression loss. Results are not directly comparable with [17]'s Table 1 due to difference in training data.[3]

5 RESULTS & DISCUSSION

As shown in Table 1, all of the position-masking models outperformed the baseline. The differences are significant ($p < 0.001$). Two position-masking models outperformed LayoutLM without position masking by over 5%. Full position masking with regression loss was significantly worse than either of the cross entropy conditions ($p < 0.05$). No other differences were significant.

Since our position masking implementation did not include height or width embeddings, we performed an ablation to determine whether masking the positions or removing height and width was responsible for the difference. LayoutLM with no height or width embeddings and with no position masking achieved mean F_1 of 67.5, which is significantly better than the baseline ($p < 0.001$), although the improvement was not as large the improvement from position masking. The model with no height or width embeddings showed nearly the same improvement in recall as the position masking models (mean recall 72.7), but a much smaller increase in precision (mean precision 63.0). Both position masking systems with cross entropy loss had significantly better F_1 scores than the ablated model ($p < 0.001$), as did masking x1 with regression loss ($p < 0.05$).

6 CONCLUSION

This work has introduced a new pre-training task, position masking, to improve layout-aware text embeddings. We have shown that adding position masking to a LayoutLM baseline model improved performance on a form understanding task. Future work should explore how well this technique can generalize to other tasks.

REFERENCES

[1] Timo I. Denk and Christian Reisswig. 2019. BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. In *Workshop on Document Intelligence at NeurIPS 2019*. arXiv:1909.04948 https://openreview.net/forum?id=H1gsGaq9US

[2] 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* (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[3] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, USA, 1440–1448. https://doi.org/10.1109/ICCV.2015.169 arXiv:1504.08083

[4] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. 991–995. https://doi.org/10.1109/ICDAR.2015.7333910 arXiv:1502.07058

[5] Xavier Holt and Andrew Chisholm. 2018. Extracting structured data from invoices. In *Proceedings of the Australasian Language Technology Association Workshop 2018*. Dunedin, New Zealand, 53–59. https://www.aclweb.org/anthology/U18-1006

[6] Guillaume Jaume, Hazım Kemal Ekenel, and Jean Philippe Thiran. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *ICDAR-OST*. https://doi.org/10.1109/icdarw.2019.10029 arXiv:1905.13538

[7] Anoop R. Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards Understanding 2D Documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4459–4469. https://doi.org/10.18653/v1/D18-1476 arXiv:1809.08799

[8] Mohamed Kerroumi, Othmane Sayem, and Aymen Shabou. 2020. VisualWordGrid: Information extraction from scanned documents using a multimodal approach. *arXiv Preprint* (2020). arXiv:2010.02358

[9] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Vol. 2006. 665–666. https://doi.org/10.1145/1148170.1148307

[10] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph Convolution for Multimodal Information Extraction from Visually Rich Documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 32–39. https://doi.org/10.18653/v1/N19-2005 arXiv:1903.11279

[11] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. 2020. ZeroShotCeres: Zero-Shot Relation Extraction from Semi-Structured Webpages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8105–8117. https://www.aclweb.org/anthology/2020.acl-main.721

[12] Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. Representation Learning for Information Extraction from Form-like Documents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6495–6504. https://doi.org/10.18653/v1/2020.acl-main.580

[13] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised Open Information Extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 885–895. https://doi.org/10.18653/v1/N18-1081

[14] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3645–3650. https://doi.org/10.18653/v1/P19-1355

[15] Yu-An Wang and Yun-Nung Chen. 2020. What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6840–6849. https://www.aclweb.org/anthology/2020.emnlp-main.555

[16] Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. 2020. DocStruct: A Multimodal Method to Extract Hierarchy Structure in Document for General Form Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 898–908. https://doi.org/10.18653/v1/2020.findings-emnlp.80 arXiv:2010.11685

[17] 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 (KDD '20)*. Association for Computing Machinery, New York, NY, USA, 1192–1200. https://doi.org/10.1145/3394486.3403172

[18] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2020. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. *arXiv Preprint* (2020), 1–16. arXiv:2012.14740 http://arxiv.org/abs/2012.14740

A SUPPLEMENTAL TOBACCO DOCUMENTS DATASET

Our pre-training data incorporates 180,000 document images we collected from the University of California San Francisco's Truth Tobacco Industry Documents database (formerly known as the

Legacy Tobacco Documents Library), the same source that [9] used to build IIT-CDIP originally.

We collected approximately 99,000 PDFs from the tobacco litigation documents using the search terms '(availability:public AND industry:tobacco)'. We iterated through the files in an arbitrary order, dividing them into their individual pages, until we had over 300,000 pages. After performing OCR on these pages using Tesseract, we manually inspected samples of the smallest hOCR output files. We determined that files under 1.8 kb in size typically did not include text except for whitespace, so we deleted pages that yielded such small output. From the remaining pages, we randomly sampled 180,000 to be our supplemental dataset.

To enable replication, we will provide a list of the specific pages in the supplemental dataset.

B DETAILED EXPERIMENTAL SETTINGS

Pre-training. We initialized the model parameters with the BERT-base language model. Distributed training was set up with 8 Tesla V100 GPUs, each with 32GB memory. Batch size is 25 per GPU, so the total batch size is 200. We use the same AdamW optimizer as LayoutLM with a learning rate of 5e-5 without weight decay. A linear learning rate schedule was used that goes from the initial value to zero at the end of all epochs. The gradient norm was clipped at 1. For position masking loss, the weight λ is set to 1 based on initial experiments on a small pre-training dataset.

Fine Tuning. Fine tuning was done in a single GPU as the dataset is small. So the batch size is 25. Same optimizer and learning rate scheduler as pre-training was used.