Understanding Long Document with Different Position-Aware Attentions

Hai Pham[†][*], **Guoxin Wang**[‡], **Yijuan Lu**[‡], **Dinei Florencio**[‡], **Cha Zhang**[‡]
[†]*Language Technologies Institute, Carnegie Mellon University*
[‡]*Microsoft Azure AI*
htpham@cs.cmu.edu {guow,yijlu,dinei,chazhang}@microsoft.com

Abstract

Despite several successes in document understanding, the practical task for long document understanding is largely under-explored due to several challenges in computation and how to efficiently absorb long multimodal input. Most current transformer-based approaches only deal with short documents and employ solely textual information for attention due to its prohibitive computation and memory limit. To address those issues in long document understanding, we explore different approaches in handling 1D and new 2D position-aware attention with essentially shortened context. Experimental results show that our proposed models have the advantages for this task based on various evaluation metrics. Furthermore, our model makes changes only to the attention and thus can be easily used for any transformer-based architecture.

1 Introduction

The task of document understanding has recently gleaned many successes (Xu et al., 2020, 2021b; Appalaraju et al., 2021). This task requires multimodal input that makes it heavier than the text-only ones, resulting in most models only being capable of dealing with short documents, i.e. having up to 512 tokens. However, there exist long documents almost everywhere, e.g. contracts, scientific papers, newsletters, or Wikipedia articles, which are typically longer than 1,000 words. To automatically summarize and understand such long documents urges long document understanding to become an important task in both NLP and AI.

Long document understanding faces several big challenges. 1) Recent document understanding approaches heavily rely on transformer (Vaswani et al., 2017). However, transformer suffers from the quadratic attention that usually limits the input to 512 words. Therefore, the correlation across long paragraphs/pages is yet to be learned. 2) Understanding long documents requires power to model all long information available, not only just in text but also in other modalities such as spatial information. For example, LayoutLM (Xu et al., 2020) showed that short document understanding is largely improved by additionally embedding spatial into text information. How to efficiently make use of spatial information for long document understanding, however, is still an open and challenging problem regarding computation cost and adaptability.

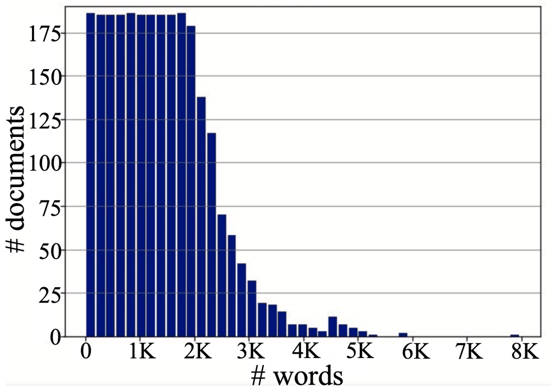


Figure 1: Distribution of document length in RVL-CDIP (Harley et al., 2015), a subset of IIT-CDIP used predominantly in the document understanding pretraining task. Most of them are longer than 1,000 words.

For long documents as shown in Figure 1, it is reasonable to assume that useful information is spanned across their lengths. Especially current OCR technology, which is essential for data preprocessing, only supports extracting spatial information on every page basis, without the knowledge of other pages. This behavior poses yet another big challenge in dealing with long documents, which requires a proper method to connect information across pages for all input modalities given.

In this paper, we discover new approaches in dealing with long document understanding, which

[*] Work done during an internship at Microsoft.

address the aforementioned challenges. We carefully preprocess OCR data to establish the proper linkages across pages. Then we explore approaches for directly reducing the heavy attention cost while achieving high performance, flexibly using the typical 1D (textual) and/or novelly, 2D (spatial) reduced contextual information, without adding more components into the already-heavy transformer (Appalaraju et al., 2021; Nguyen et al., 2021). Despite being simple, we show through experiments that both 1D and 2D information can enhance the practicality of transformer-based models while achieving the needed power of handling long documents.

Our contributions 1) We newly motivate the simplistic, flexible use of spatial input in attention, making it plug-able to transformer. 2) We are able to tackle the document understanding task with input data up to 4096 words. 3) Experimental results prove the advantages of our approaches on various long-document datasets in comparison to short models for both 1D and 2D contextual information.

2 Related Work

Transformer Attention For Long Documents There are several methods that address the quadratic cost transformer attention. Longformer (Beltagy et al., 2020) uses sliding window to reduce the context, only retrains some sparse global connections. Similarly, ETC (Ainslie et al., 2020) embeds relative positions and adds contrastive predictive encoding. Bigbird (Zaheer et al., 2020) optimizes Longformer's sliding window by adding random connections. Our model similarly uses sliding window but differs in that it exploits layout input along with the typical text input flexibly and directly into attention.

Multimodal Document Pretraining Document understanding largely inherits from multimodal pretraining (Li et al., 2020; Chen et al., 2020; Luo et al., 2020) with the successes from LayoutLM (Xu et al., 2020, 2021a). Recently, Docformer (Appalaraju et al., 2021) and StrucuturalLM (Li et al., 2021) introduced a two-pronged approach: having new pretrain tasks and suitable changes to the processing or embedding. Probably Skim-Attention (Nguyen et al., 2021) has the most related motivation for long documents, although we have a more memory-efficient, and faster way of handling layout input directly into attention and not from after the embedding like theirs, and consequently support longer input (4096 vs. 2048).

3 Our Model

3.1 Pretrain Model Architecture

We employ Masked Lanuage Model (MLM) architecture as in other document intelligence work, e.g. Xu et al. (2020, 2021a); Appalaraju et al. (2021), but make proper changes to enable the capability of long documents. Different from a typical MLM, we have multimodal–instead of text-only input–which makes the model heavier and thus cannot deal with long documents without proper changes, as described in our model shown in Figure 2.

First, we use the sliding-window inspired from Beltagy et al. (2020), given its lightweight and elegance in limiting the context window, making it significantly more memory friendly. Second, we introduce new spatial-based attention masks, in which each context window to a bounding box is determined by calculating its spatial neighbors, instead of the given neighboring words. Likewise, our model does not only use spatial input in the embedding but also in attention directly with preserved spatial correlation. Section 3.3 will elaborate on the establishment and usage of these new distance masks in comparison with others.

3.2 Post-OCR Processing

This processing is crucially important for long documents that usually have multiple pages because current OCR engines only generate single-page results, without any connections among pages. More current models are short models that support up to 512 tokens, and thus discard the rest of valuable information. Consequently, to make our model capable of long documents, we process the post-OCR data to establish the connections for all input components among the pages. For example, the bounding boxes on page n need adjusting the coordinates to include the previous $n-1$ pages.

3.3 Different Attention Masks

We begin to describe the original transformer attention and our different approaches for long documents, using 1D and 2D input data.

Original Attention Masks (Vaswani et al., 2017) (also known as full attention masks) For each layer, it is calculated by Equations (1) and (2),

$$\text{score}(\mathbf{Q}, \mathbf{K}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \tag{1}$$

$$\text{attn\_score}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{score}(\mathbf{Q}, \mathbf{K}) \cdot \mathbf{V}, \tag{2}$$

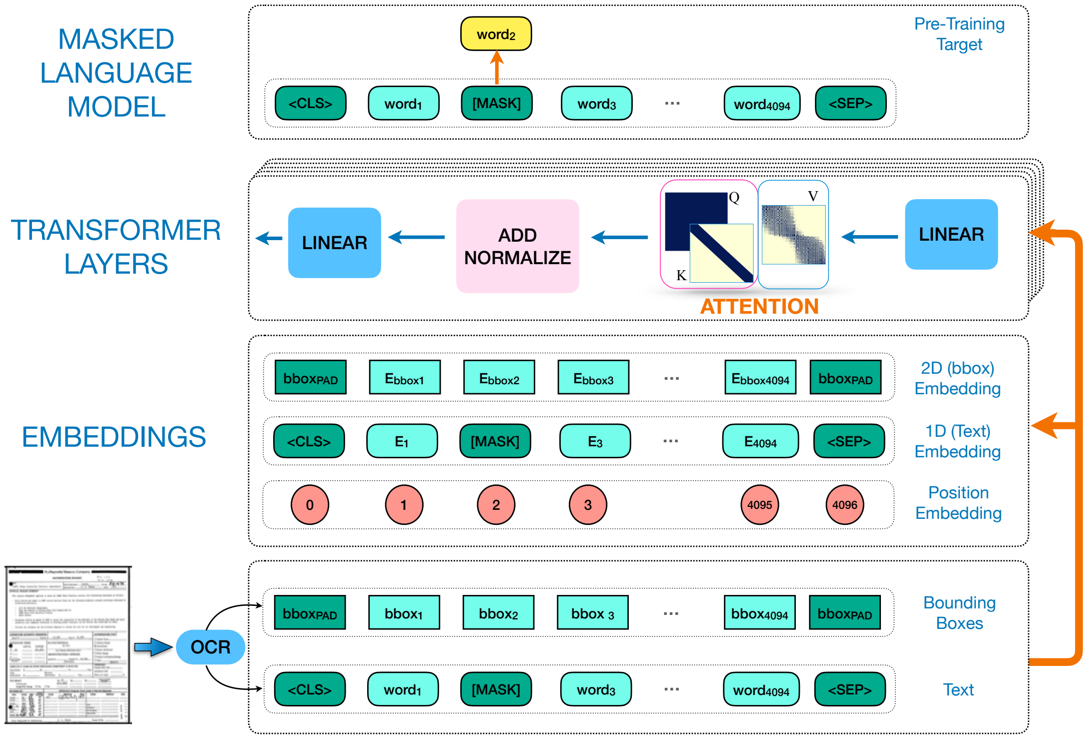


Figure 2: Our MLM pretrain model architecture. Unlike LayoutLM, we use 1D and 2D input for not only in embeddings but also in transformer attention.

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ stand for the learnable Query, Key, and Value matrices respectively. Given the lengths of these three matrices are all N (input length), the complexity of each step is $\mathcal{O}(N^2)$.

Sliding-Window Masks (Figure 3a) We use the sliding-window approach as inspired from Beltagy et al. (2020), which limits the context for each token from N down to a smaller M, e.g. $N = 4096, M = 512$, and so the complexity is reduced to $\mathcal{O}(NM)$.

$$\mathbf{K}_w = \text{get\_window}(\mathbf{K}) \tag{3}$$

$$\text{score}(\mathbf{Q}, \mathbf{K}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}_w^T}{\sqrt{d_k}}\right) \tag{4}$$

$$\mathbf{V}_w = \text{get\_window}(\mathbf{V}) \tag{5}$$

$$\text{attn\_score}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{score}(\mathbf{Q}, \mathbf{K}) \cdot \mathbf{V}_w \tag{6}$$

Using that intuition, the calculations are now changed to Equations (3–6), with the added `get_window` steps in Equations (3) and (5) [1].

Sliding-Window plus Random Token Masks (Figure 3b) On top of sliding windows, we add a few random tokens to establish more connections to the attention, similarly to Zaheer et al. (2020). This operation essentially makes changes only to Equations 3 and 5, with extraction of random tokens.

$$\mathbf{K}_w = \text{get\_2D\_window}(\mathbf{K}) \tag{7}$$

$$\mathbf{V}_w = \text{get\_2D\_window}(\mathbf{V}) \tag{8}$$

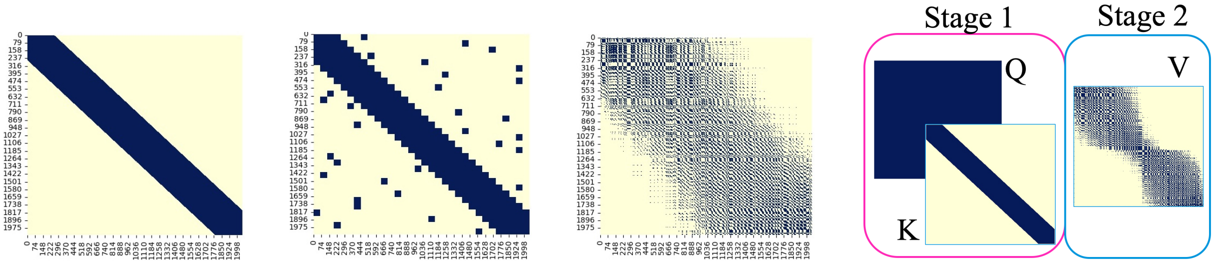
Spatial Distance Masks (Figure 3c) Different from previous attention types, the M context tokens for each token is not decided by textual (1D) but instead by spatial (2D) input with some steps. First, we calculate the centers of all bounding boxes. Second, we fit the kNN algorithm to the sequence of those points based on L2 distance, resulting in a 2D distance matrix (having the same shape NM as sliding window). In summary, we replace Equations 3 and 5 with Equations 7 and 8.

3.4 Pretrain Model Variants

We build out MLM pretrain architecture with various attention mechanisms for long documents as described in Section 3.3 and compare their performances in several tasks. Since this change is only made directly to the attention, our method can be used off-the-shelf for transformer-based architecture with multimodal input.

SW Model This model directly uses Sliding-Window (SW) masks for attention, which significantly reduces the computation and was shown to be effective for long documents in text-based tasks.

[1] To enable fast calculations in Equations (4) and (6) with now-changed matrix shapes, one has to extract and chunk the contexts for all tokens in a way that can exploit fast matrix multiplication (e.g. by using `einsum`)



(a) SW (b) SW+RAND (c) DISTANCE (d) DISTANCE+SW

Figure 3: Visualization of our models' different types of attention mask for real samples from RVL-CDIP dataset (Harley et al., 2015) with limit length of 2048 and context size 512 (for both textual and spatial cases). Fig 3a is sliding window (SW), Fig 3b is sliding window in blocks with 1-per-block random blocks (SW+RAND), Fig 3c is spatial-based distance mask, and Fig 3d is the combination of sliding window and distance modes. *Legend*: Attention mask may only have values of 0 and 1, which are represented as the light-yellow background and dark-blue foreground colors, respectively.

SW+RAND Model This model uses Sliding-Window plus Random Token Masks.

DISTANCE Model This model uses Spatial Distance Masks, with all neighboring contexts being preemptively computed using kNN, and is implemented in the data loading instead of transformer encoding phase, not to slow down the main process.

DISTANCE+SW Model. In this model, we combine the spatial and textual attention masks together in a single attention pass, with the motivation of combining the benefits of those two. In detail, it is done by Equations (3–6), with Equation (3) now being replaced by Equation (7).

4 Experiments

4.1 Tasks and Datasets

Pretraining We use **IIT-CDIP Test Collection 1.0** [2] dataset for our MLM pretraining task. This is a large-scale dataset with over 6M multi-page documents and around 11M pages in total (each page is stored as an image and is preprocessed by an OCR engine).

Document Classification This document classification task uses **RVL-CDIP** (Harley et al., 2015) dataset, which is a subset of the pretraining dataset IIT-CDIP. It comprises 16 classes and each class equally has 25K grayscale images. The document length distribution is shown in Figure 1.

Sequence Labeling There are two datasets for this task, namely Kleister-NDA and FunSD.

1) FunSD (Guillaume Jaume, 2019)[3] This is a lightweight dataset that has 199 noisy scanned forms, which contain around 31K words and 9.7K entities with 7 given token classes. Though it is not a long-document dataset (all documents have < 512 words), it is useful for ablation studies.

2) Kleister-NDA (Graliński et al., 2020; Stanisławek et al., 2021)[4] This dataset has 540 documents in total (254 train, 83 val, and 203 test) with 2,160 entities annotated and an average of 2,540 words per document. Due to the difficulty in reproducibility with unclear results post-processing, this task is cast similarly to FunSD with 4 classes.

4.2 Baselines

We compare our 4 model variants (Figure 3) with the following baselines:

Text: This group consists of models that only accept text input including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and other long models including Bigbird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020) [5].

Text+Layout: This group contains models that accept both text and layout information, including LayoutLM (Xu et al., 2020) variants.

4.3 Results and Discussions

Document Classification As shown in Table 1, long models (SeqLen 4096) clearly outperform short ones in both baseline groups, with or without layout information added to the input. Furthermore, all our 4 model variants outperform all the baselines.

This result concurs with our observation that long documents have valuable information spanned

[2] https://ir.nist.gov/cdip/

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

[4] https://github.com/applicaai/kleister-nda

[5] Our SW and SW+RAND models share the similarity with those last two ones, with difference of handling multimodal input for document intelligence.

Type	Model	SeqLen	Acc (%) ↑
Text	BERT-base	512	89.81
	RoBERTa-base	512	90.06
	BERT-large	512	89.92
	RoBERTa-large	512	90.11
	Bigbird-base	4096	93.48
	Longformer-base	4096	93.85
	Bigbird-large	4096	93.34
	Longformer-large	4096	93.73
Text+Layout	LayoutLM-base	512	91.88
	LayoutLM-large	512	91.90
	Ours SW	4096	94.50
	Ours SW+Rand	4096	**95.25**
	Ours Distance	4096	94.79
	Ours Distance+SW	4096	94.69

Table 1: Classification accuracy for RVL-CDIP. For this long-document dataset, the models capable of using 4096 words uniformly beat other models and layout information helps with the task compared with using Text input. All our long models show their advantages on this long dataset.

Type	Model	SeqLen	F1 ↑
Text	BERT-base	512	47.06
	BERT-large	512	52.66
	Longformer-base	4096	61.78
	Bigbird-base	4096	46.98
Text+Layout	LayoutLM-base	512	55.69
	LayoutLM-large	512	61.95
	Ours SW	4096	**64.06**
	Ours SW+Rand	4096	58.92
	Ours Distance	4096	57.01
	Ours Distance+SW	4096	44.70

Table 2: Results on Kleister-NDA. Although this dataset is challenging, long models still show advantages over short ones.

across the length. And importantly, our models show advantages of handling long multimodal input, and hence are more practical with real data that are usually longer than 512 tokens.

Sequence Labeling with Kleister-NDA[6] Comparing the "base" versions (separated from their "large" counterparts), Table 2 shows that most of our models, which are also the "base" ones, clearly have better scores. Particularly, our SW model is the best performer.

Furthermore, our Distance+SW is not performing equally well. Our hypothesis is that the OCR engine cannot understand the decoying annotation in this dataset, and thus generates spatial results that do not correlate well with the text. Consequently, the combination of textual and spatial information does not result in the benefits of those two.

4.4 Ablation: Long Models on Short Dataset

The purpose of this study is to explore how long models perform on short documents, which also appear in practice, to see whether they can generalize their performance to shorter data.

Type	Model	SeqLen	F1 ↑
Text	BERT-base	512	60.3
	RoBERTa-base	512	66.5
	BERT-large	512	65.6
	RoBERTa-large	512	70.7
	Bigbird-base	4096	45.8
	Longformer-base	4096	71.4
	Bigbird-large	4096	46.8
	Longformer-large	4096	73.5
Text+Layout	LayoutLM-base	512	78.7
	LayoutLM-large	512	**79.0**
	Ours SW	4096	69.9
	Ours SW+Rand	4096	77.1
	Ours Distance	4096	64.0
	Ours Distance+SW	4096	61.8

Table 3: Results on FunSD dataset. As usual, layout information is helpful in boosting performance. However, long models do not perform well compared with short models on this small, short-document dataset.

Table 3 shows that on FunSD, we see again that layout information generally helps in the case of multimodal input. However, long models do not perform well compared to short ones, although the gap between the best of ours and the baselines are not very far away (77.1 vs. 79.0). The main reason is that long models essentially have much more parameters than short ones. And not only is FunSD short, it is also very small. As a result, the limited phase of fine-tuning on only 199 samples can hardly tune parameters well for good results. Especially, since all documents are short, most long input to the model is zero padding and thus not enough for contributing for better scores.

Another reason is that long models have their embedding representations trained for the length of 4096 tokens and hence are hard to adapt to 512-token input with just a few fine-tuning steps. As a result, analyzing the data well to design suitable pretraining and fine-tuning models is very important.

The next 2 studies will explore the implications of the newly-added spatial attention masks in our models.

4.5 Ablation: Different-Length Documents

This study aims to explore how the models work if we do not cut any information from documents (the models take input up to their maximum length

[6] The results are from the validation split due to no annotation for the test split provided in the dataset.

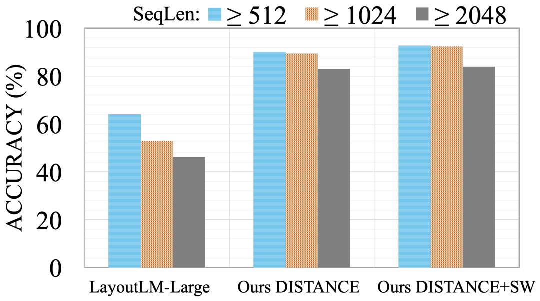


Figure 4: RVL-CDIP performance on different document types based on their original lengths (i.e. without purging) with LayoutLM (with the best "large" version) and our spatial models (DISTANCE and DISTANCE+SW). Our models are consistently better.

limit). Out of 40K test samples in RVL-CDIP, there are 9268 samples with length $\geq$ 512, 2312 with length $\geq$ 1024, and only 106 with length $\geq$ 2048.

Figure 4 shows the consistent observation that our models are much better than LayoutLM, and yet perform slightly worse as the original document length increases. There could be several possible reasons for this behavior: the models are not well pretrained and/or fine-tuned, many long documents have lots of confusing parts, or there are many noises in OCR results.

4.6 Ablation: Different Max Input Lengths

Given the pretrained models that can accept input up to 4096 tokens, we finetune them with the input of different maximum lengths, i.e. excess will be purged. As a result, we use all 40K test samples in RVL-CDIP for this study.

As shown in Figure 5, our models are better and better as more tokens are absorbed, thus once again confirming our intuition that valuable information is spanned across the length. As a result, if the model capacity permits, we should not limit the capacity to 512 tokens as in most current models in the literature.

4.7 Further Discussion on Spatial Masks

As seen in the above experimental results, direct usage of 2D layout context information in the transformer attention has some advantages. However, its performance does not match the typical usage of 1D textual information. This might be discouraging at first since introducing spatial masks brings heavier computation compared to textual ones. We hypothesize the drawbacks are due to some objective limitations. First, the kNN suffers some inaccuracy compared with normal (and slow) calculations. Second, the performance of the whole pipeline heavily depends on OCR quality, e.g. in Kleister-NDA with decoy design, OCR results are not well aligned with the text. Consequently, we conjecture that with future development in OCR technologies, the use of spatial masks would be more and more helpful in practice.

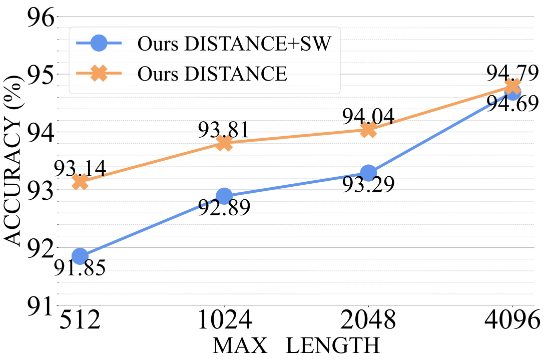


Figure 5: RVL-CDIP performance on different maximum lengths using our DISTANCE and DISTANCE+SW models. For each of lengths 512, 1024, 2048, and 4096, the test set contains the same 40K samples. A longer maximum length gives better results.

5 Conclusion and Discussion

We propose a versatile solution for long document understanding, in which the shortened context can be used in the form of textual and/or layout input for the attention mechanism in a flexibly pluggable manner. We keep our approach simple by not putting extra overhead on complicated embedding or encoding methods. Despite its simplicity, our solution has shown promising experimental results on document understanding tasks with long, multimodal input. In the future, we will further reduce the memory consumption of models with given multimodal input and speed up the pretraining. Similar to LayoutLM, pretraining usually takes 80 hrs/epoch with 8 V100 GPUs. Thus there are certainly lots of room for improvement to make these models more efficient and practical.

6 Acknowledgements

The authors would like to thank Microsoft for providing GPU training resources for the projects, the colleagues in Microsoft, and last but not least, the anonymous reviewers for helpful discussions and feedback.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. *ICCV*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*.

Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2020. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356*.

Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*.

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. *International Conference on Document Analysis and Recognition*.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. *ACL*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.

Laura Nguyen, Thomas Scialom, Jacopo Staiano, and Benjamin Piwowarski. 2021. Skim-attention: Learning to focus via document layout. *EMNLP Findings*.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. Kleister: Key information extraction datasets involving long documents with complex layouts. *arXiv preprint arXiv:2105.05796*.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2021a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *ACL*.

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.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021b. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

Parameter Name	Value
do_lower_case	true
fp16	true
fp16_backend	amp
gradient_accumulation_steps	4
max_seq_length	4096
max_2d_position_embeddings	1024
max_steps	1000000
model_name_or_path	allenai/longformer-base-4096
dataloader_num_workers	64
tasks	mask_lm
optimizer	transformers_AdamW
learning_rate	5e-5
warmup_ratio	0.1
weight_decay	0.01
whole_word_masking	false
add_prefix_space	true
attention_window	512

Table 4: Main pretrain hyperpameters on the MLM pretraining task for the ITT-CDIP large-scale dataset. There are 3 variants share this set of parameters that are Ours SW, Ours DISTANCE and Ours DISTANCE+SW models. All of them use the pretrained weights from Longformer-base (Beltagy et al., 2020) model.

A More Information on the Pretrain Task

Pretrain Data Preprocessing As described, for pretrain model we retain the same OCR engine for generating and aligning layout and text information from LayoutLM (Xu et al., 2020). The task is also the same, which is Masked Language Modeling (MLM). To deal with long documents, we have to implement the additional sliding-window, random-block and distance-based masks.

Pretrain Model Implementation Our solution only makes changes to the attention module, in which uses can choose to use any types of attention masks from the 4 variants illustrated in Figure 3.

For the SW and SW+RAND models which are also our new pretrain models, we implement the layout-related part on top of the original BigBird [7] and Longformer [8] implementations from Huggingface's transformers, respectively. Otherwise the distance-based masks, which are employed in DISTANCE and DISTANCE+SW models, are newly implemented as a pluggable module.

Training MLM We pre-train the task on the IIT-CDIP datasets, using a single-node multi-GPU mode. Each job was run on a server with 8 V100 Nvidia GPUs, each of which has 32GB memory and fast processors. For text-only models, please refer to LayoutLM's github [9].

For SW model, we use the public pretrained weights from Lomgformer (Beltagy et al., 2020). Other of our models employ the same set of parameters, except for the pretrained weights, in which SW+RAND model uses the weights from Bigbird (Zaheer et al., 2020) and the last two models having distance masks (DISTANCE and DISTANCE+SW models) use the same pretrained weights as SW model, as demonstrated in Table 4.

It is also worth noting that the pretrained weights from Longformer and Bigbird models are useful even for the models using distance masks because those two model families support documents with length 4096, so the position embeddings are helpful. For speed and memory tradeoff, we limit the context for distance masks to only 128 (vs. 512 in textual contexts), without sacrificing much performances, as reported in Section 4.3.

Training Notes Although not reported in the main content, we note some lessons learned from the pretraining task. As we observe, the Ours SW model consistently achieves the best results, while consuming the least GPU memory. For the base model, it only consumes about 7 GB GPU memory and Ours DISTANCE+SW that uses sliding-window attention on its half processing also consumes about 9 GB memory. Both models, as a result, can be deployed well on a broad range of GPUs in the market.

Unlike those conveniences, Ours SW+RAND and Ours DISTANCE do not share the same advantages. In fact, they consumes about more than 30GB GPU memory each, limiting their practicality. We hypothesize the main reason for such drawbacks is that they have random, inconsistent patterns, and hence there is no efficient way to take advantage of fast memory-efficient and fast matrix operations.

Finally, although showing promising practical behaviors, all baselines and our models, and almost any transformer-based ones are certainly not lightweight models. And although there are advancements in compressing those heavy models (e.g. (Touvron et al., 2021; Frankle and Carbin, 2019), there seems to be a considerable way to go for making these model run on mobile devices in the near future.

[7] https://huggingface.co/transformers/model_doc/bigbird.html

[8] https://huggingface.co/transformers/model_doc/longformer.html

[9] https://github.com/microsoft/unilm

B More information on Finetuning Tasks

As described in the main content, after pretraining, the saved models are the backbone for the respective fine-tuning model types. For that reason, the parameters are mostly shared with their pretrain counter-part models, e.g. Table 4 for Ours SW models. Generally, we keep the same optimizer and batch size of 32 (combined across all used parallel GPUs).

For **RVL-CDIP** in the document classification task, we use the `SequenceClassification` model type. On top of the pretrain skeleton, we add a small classifier with 2 fully-connected layers and a drop-out layer in between. The final output is the single class for the whole sequence/document.

For **FunSD** and **Kleister-NDA** datasets, we instead use the `TokenClassification` model type, which is designed to classify all-document entities. The similar classifier is added to the pretrained skeleton, now with a different usage in which each token/entity is to be classified into 1 of the number of given classes.

What's more, to preprocess these two datasets, we have to ingest all available document tokens. Likewise, with documents longer than the maximum lengths, we need to cut those documents, and recursively treat the overflowing parts in the same way. In terms of implementation, unlike FunSD that is lightweight, we always want to avoid loading the whole dataset into the memory but rather take advantage of the data buffering in feeding to the models. As a result, we pre-process all data first, save them to disks and only load the respective parts when needed.

Additional Information for Kleister-NDA It is worth recalling that the evaluation of it is tricky if using the provided official GEval evaluation script (Graliński et al., 2020)[10]. In detail, given the predited tokens, one has to retrieve the associated texts in a group. For example, the beginning of an entity group usually starts with a class beginning with "B-", followed by a series of "I-" tokens. However, there is no guarantee that the prediction will always return a group having this meaningful pattern, let alone many other complicated cases that can happen. Such complications make the post-processing of the prediction– before feeding to GEval–very difficult and importantly, not easily reproducible. In fact, amongst recent papers that report performance on this dataset (e.g. in Xu et al. (2021a); Appalaraju et al. (2021)), there is reference code with which for us to compare.

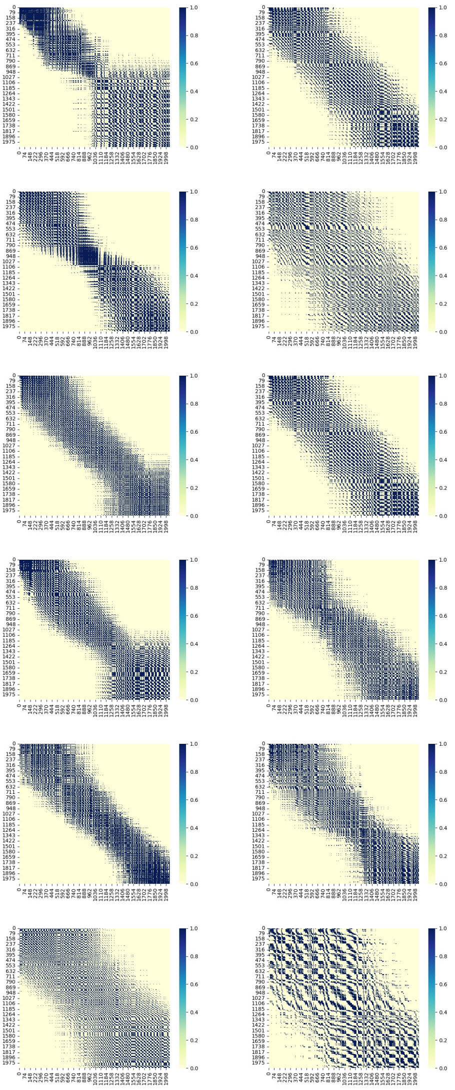


Figure 6: More distance masks from RVL-CDIP samples with the limit length of 2048 and 512 neighbors each.

Consequently, we treat this dataset the same as FunSD, given their similarity in annotation. In addition, because this dataset is larger and much more difficult (due to decoying texts) compared to FunSD, we analyze the train dataset and employ the weighted loss based on the distribution the given labels. As a result, our method is more transparent and reproducible.

C Additional Samples on Distance Masks

Complementary to Figures 3c and 3d, we present some more distance masks based on real samples taken from RVL-CDIP with the same setting in Figure 6.

[10] https://github.com/applicaai/kleister-nda