BROS: A Pre-trained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents

Teakgyu Hong\textsuperscript{1}, Donghyun Kim\textsuperscript{1}, Mingi Ji\textsuperscript{2}, Wonseok Hwang\textsuperscript{3}, Daehyun Nam\textsuperscript{4}, Sungrae Park\textsuperscript{4}

\textsuperscript{1}NAVER CLOVA, \textsuperscript{2}KAIST, \textsuperscript{3}LBox, \textsuperscript{4}Upstage AI Research, Upstage AI
teakgyu.hong@navercorp.com, dong.hyun@navercorp.com, qwertgfdcvb@kaist.ac.kr, wonseok.huang@lbox.kr, daehyun.nam@upstage.ai, sungrae.park@upstage.ai

Abstract

Key information extraction (KIE) from document images requires understanding the contextual and spatial semantics of texts in two-dimensional (2D) space. Many recent studies try to solve the problem by developing pre-trained language models focusing on combining visual features from document images with texts and their layout. On the other hand, this paper tackles the problem by going back to the basic: effective combination of text and layout. Specifically, we propose a pre-trained language model, named BROS (BERT Relying On Spatiality), that encodes relative positions of texts in 2D space and learns from unlabeled documents with area-masking strategy. With this optimized training scheme for understanding texts in 2D space, BROS shows comparable or better performance compared to previous methods on four KIE benchmarks (FUNSD, SROIE*, CORD, and SciTSR) without relying on visual features. This paper also reveals two real-world challenges in KIE tasks—(1) minimizing the error from incorrect text ordering and (2) efficient learning from fewer downstream examples—and demonstrates the superiority of BROS over previous methods.

Introduction

Automatic key information extraction (KIE) from industrial documents is an essential task in robotic process automation (RPA). Extracting an ordered item list from receipts (Park et al. 2019), prices and taxes from invoices (Liu et al. 2019), and paired key-values from form-like documents (Jaume et al. 2019) are representative examples. Since the task requires understanding texts in various layouts, the combination of multiple technical components from both computer vision and natural language processing is required.

Figure 1 describes a schematic illustration of pipeline for the document KIE tasks (Hwang et al. 2019; Denk and Reisswig 2019). First, given a document image, optical character recognition (OCR) detects the texts in the image and recognizes the content to generate a set of text blocks. Next, a serializer identifies a reading order of text blocks distributed in 2D image space and converts them into text sequence in 1D text space to apply NLP technology (Hwang et al. 2019; Denk and Reisswig 2019). The linguistic understanding of the pre-trained language model leads to superior performance than rule-based extractions. However, the conversion of texts in 2D space

Figure 1: Schematic illustrations of document KIE pipeline.

| Model                  | Img # | Params (O) | (P) | (F)   |
|------------------------|-------|------------|-----|-------|
| LayoutLM\textsubscript{BASE}  | 113M  | 78.66      | 33.89 | 62.50 |
| LayoutLM\textsubscript{MV2}\textsubscript{BASE} | ○      | 200M       | 82.76 | 40.77 | 69.92 |
| BROS\textsubscript{BASE}     | 110M  | 83.05      | 76.94 | 72.60 |
| LayoutLM\textsubscript{LARGE} | 343M  | 78.95      | 33.11 | 61.00 |
| LayoutLM\textsubscript{MV2}\textsubscript{LARGE} | ○      | 426M       | 84.20 | 62.53 | 72.12 |
| BROS\textsubscript{LARGE}    | 340M  | 84.52      | 79.42 | 74.42 |

Table 1: Performance comparison of pre-trained language models on (O)riginal, (P)ermuted, and (F)ew training samples FUNSD KIE tasks. In (F), 10 samples are used.

The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
into a text sequence in 1D space leads to the loss of layout information that is critical in KIE tasks.

To avoid the loss of layout information, a new type of language model, LayoutLM (Xu et al. 2020) expands a 1D positional encoding of BERT to 2D and is trained over a large corpus of industrial documents to understand spatial dependencies between text blocks. Its fine-tuning has shown breakthrough performances on multiple KIE tasks and becomes a strong baseline. After the rise of LayoutLM, several studies try to develop pre-trained language models by combining additional visual features (Xu et al. 2021; Powalski et al. 2021; Li et al. 2021b; Appalaraju et al. 2021; Li et al. 2021c) (e.g. image patches identified by an object detection) and show further performance improvements. However, the extensions using visual features require additional computational costs and they still demand more effective combinations of texts and their spatial information.

In this paper, we introduce a new pre-trained language model, named BROS, by re-focusing on the combinations of texts and their spatial information without relying on visual features. Specifically, we propose an effective spatial encoding method by utilizing relative positions between text blocks, while most of previous works employ absolute 2D positions. Additionally, we introduce a novel self-supervision method, named area-masked language model, that hides texts in an area of a document and supervises the masked texts. With these two approaches for encoding of spatial information, BROS shows superior or comparable performances compared to previous methods using additional visual features.

Aside from improving KIE performances, BROS also addresses two important real-world challenges in KIE tasks: minimizing dependency on the order of text blocks and learning from a few training examples of downstream tasks. The first challenge indicates the robustness on the serialization followed by the OCR process in Figure 1. In real scenario, document images are usually irregular (i.e. rotated or distorted documents) and the serializer might fail to identify a proper order of text blocks. In addition, when serialization fails, the performance of sequence tagging approaches (e.g. BIO tagging), which most previous works employ, drops dramatically. To circumvent the difficulty, we apply SPADE (Hwang et al. 2021) decoder that extracts key text blocks without any order information to the pre-trained models and evaluates them on the new benchmarks where the order of text blocks are permuted. As a result, BROS shows better robustness on the serializers compared to LayoutLM (Xu et al. 2020) and LayoutLMv2 (Xu et al. 2021).

The second challenge is related to the required number of labeled examples to understand the target key contents. Since a single KIE example consists of hundreds of text blocks that should be categorized, the annotation is expensive. Most public benchmarks consist of less than 1,000 samples, even though the target documents contain hundreds of layouts and diverse contexts. In this paper, we analyze KIE performances over the number of training examples and compare the pre-trained models. As a result, BROS performs better on FUNSD KIE tasks, and also BROS only with 20~30% of FUNSD examples achieves better performance than LayoutLM with 100% of them. Summarized results for these experiments are shown in Table 1.

Our contributions can be summarized as follows:

- We propose an effective spatial layout encoding method by accounting for relative positions of text blocks.
- We also propose a novel area-masking self-supervision strategy that reflects 2D natures of text blocks.
- The proposed model achieves comparable performance to the state-of-the-art without relying on visual features.
- We compare existing pre-trained models on permuted KIE datasets that lost the orders of text blocks.
- We compare the fine-tuning efficiency of various pre-trained models under a data-scarce environment.

**Related Work**

**Pre-trained Language Models for 2D Text Blocks**

Unlike the pre-trained models for conventional NLP tasks, such as BERT (Devlin et al. 2019), LayoutLM (Xu et al. 2020) is first proposed to jointly model interaction between text and layout information for the document KIE task. It encodes the absolute position of text blocks with axis-wise embedding tables and learns a token-level masked language model that hides tokens randomly and estimates the origins. After the publication of LayoutLM, several pre-trained models have been tried to additionally integrate visual features, such as visual feature maps from raw images (Xu et al. 2021; Appalaraju et al. 2021), image patches identified by an object detection module (Li et al. 2021b), and visual representations of text blocks (Powalski et al. 2021; Li et al. 2021c).

Although the extensions imposing multi-modalities of visual and textual features provide additional performance gains in KIE tasks, they spend additional computations to process raw document images. Additionally, an effective combination of text and layout is still required as the major component of the multi-modalities.

Aside from incorporating visual features, StructuralLM (Li et al. 2021a) utilizes cell information, a group of ordered text blocks, and shows promising performance improvements. However, the local orders of text blocks might not be available depending on the KIE tasks and the OCR engines. Therefore, this paper focuses on the original granularity of text blocks identified by OCR engines and improves the combination of text and layout by an effective spatial encoding method and an area-based pre-training strategy.

**Parsers for Document Key Information Extraction**

BIO tagger, which is a representative parser for entity extraction from the text sequences, extracts key information by identifying spans with the beginning (B) and inside (I) points. Though BIO tagger has been used as a conventional method, it has two limitations for applying to document KIE. One is that the correct order of text blocks is required for extracting key information when post-processing each classified token class (i.e. B- and I- classes). For example, if the text blocks are not ordered properly, such as “recognition, optical, character”, the correct answer can not be made. The
BERT Relying on Spatiality (BROS)

The main structure of BROS follows LayoutLM (Xu et al. 2020), but there are two critical advances: (1) a use of spatial encoding metric that describes spatial relations between text blocks and (2) a use of 2D pre-training objective designed for text blocks on 2D space. Figure 2 shows a visual description of BROS for document KIE tasks.

Encoding Spatial Information into BERT

The way to encode spatial information of text blocks decides how text blocks be aware of their spatial relations. LayoutLM (Xu et al. 2020) simply encodes absolute x- and y-axis positions to each text blocks but the specific-point encoding is not robust on the minor position changes of text blocks. In this paper, we slightly modify the SPADE decoder to hold co-modality to represent positions of their semantically coupled text blocks. Specifically, we apply the SPADE decoder for entity linking tasks of KIE benchmarks and also for all tasks lost perfect order information of text blocks. Figure 3 shows a directed relation graph of tokens to represent key information of KIE tasks regardless of the order of text blocks. For formal description, we use $p = (x, y)$ to denote a point on 2D space and a bounding box of a text block consists of four vertices, such as $p^t, p^r, p^b, p^l$, that indicate top-left, top-right, bottom-right, and bottom-left points, respectively. BROS first normalizes all the 2D points of the other bounding boxes of text blocks and applies sinusoidal functions as $\tilde{p}_{i,j} = [f_{\text{sinu}}(x_i - x_j); f_{\text{sinu}}(y_i - y_j)]$. Here, $f_{\text{sinu}}: \mathbb{R} \rightarrow \mathbb{R}^{D_f}$ indicates a sinusoidal function, which is used in Vaswani et al. (2017). $D_f$ is the dimensions of sinusoid embedding and the semicolon (;) indicates concatenation. Through the calculations, the relative positions of $j$th bounding box based on the $i$th bounding box are represented with the four vectors, such as $\tilde{p}_{i,j}, \tilde{p}_{i,j}^t, \tilde{p}_{i,j}^r, \tilde{p}_{i,j}^b$. Finally, BROS combines the four relative positions by applying a linear transformation, $b_{h_{i,j}} = W^t_{h_{i,j}} \tilde{p}_{i,j}^t + W^r_{h_{i,j}} \tilde{p}_{i,j}^r + W^b_{h_{i,j}} \tilde{p}_{i,j}^b + W^l_{h_{i,j}} \tilde{p}_{i,j}^l$, (1) where $W^t, W^r, W^b, W^l \in \mathbb{R}^{(H/A) \times 2D_f}$ are linear transition matrices, $H$ is a hidden size of BERT, and $A$ is the number of self-attention heads.
Area-masked Language Model

Pre-training diverse layouts from unlabeled documents is a key factor for document KIE tasks. BROS utilizes two pre-training objectives: one is a token-masked LM (TMLM) used in BERT and the other is a novel area-masked LM (AMLM) introduced in this paper. The area-masked LM, inspired by SpanBERT (Joshi et al. 2020), captures consecutive text blocks based on a 2D area in a document.

TMLM randomly masks tokens while keeping their spatial information, and then the model predicts the masked tokens with the clues of spatial information and the other un-masked tokens. The process is identical to MLM of BERT and Masked Visual-Language Model (MVLM) of LayoutLM. Figure 4 (a) shows how TMLM masks tokens in a document. Since tokens in a text block can be masked partially, their estimation can be conducted by referring to other tokens in the same block or text blocks near the masked token.

AMLM masks all text blocks allocated in a randomly chosen area. It can be interpreted as a span masking for text blocks in 2D space. Specifically, AMLM consists of the following four steps: (1) randomly selects a text block, (2) identifies an area by expanding the region of the text block, (3) determines text blocks allocated in the area, and (4) masks all tokens of the text blocks and predicts them. At the second step, the degree of expansion is identified by sampling a value from an exponential distribution with a hyper-parameter, $\lambda$. The rationale behind using exponential distribution is to convert the geometric distribution used in SpanBERT for a discrete domain into a distribution for a continuous domain. Thus, we set $\lambda = -\ln(1 - p)$ where $p = 0.2$ used in SpanBERT. Also, we truncated exponential distribution with 1 to prevent an infinity value covering all spaces of the document. It should be noted that the masking area is expanded from a randomly selected text block since the area should be related to the text sizes and locations to represent text spans in 2D space. Figure 4 compares token- and area-masking on text blocks. Because AMLM hides spatially close tokens together, their estimation requires more clues from text blocks far from the estimation targets.

Finally, BROS combines two masked LMs, TMLM and AMLM, to stimulate the model to learn both individual and consolidated token representations. It first masks 15% of tokens for AMLM and then masks 15% of tokens on the left text blocks for TMLM. Similar to BERT (Devlin et al. 2019), the masked tokens are replaced by [MASK] token for 80%, random token for 10%, and original token for the rest 10%.

Key Information Extraction Tasks

We solve two categories of KIE tasks, entity extraction (EE) and entity linking (EL). The EE task identifies sequences of text blocks that represent desired target texts. Figure 5 (a) is an example of the EE task: identifying header, question, and answer entities in the form-like document. The EL task connects key entities through their hierarchical or semantic relations. Figure 5 (b) is an example of the EL task: grouping menu entities, such as its name, unit price, amount, and price. Table 2 lists four KIE bench-
trained in our experimental setting shows comparable performances to the published LayoutLM.

The main Transformer structure of BROS is the same as BERT. We set the hidden size, the number of self-attention heads, the feed-forward/filter size, and the number of Transformer layers of BROS\textsubscript{BASE} to 768, 12, 3072, and 12, respectively and those of BROS\textsubscript{LARGE} to 1024, 24, 4096, and 24, respectively. The dimensions of sinusoid embedding $D^*$ is set to 24 for BROS\textsubscript{BASE} and 32 for BROS\textsubscript{LARGE}.

BROS is trained by using AdamW optimizer (Loshchilov and Hutter 2019) with a learning rate of 5e-5 with linear decay. The batch size is set to 64. During pre-training, the first 10% of the total epochs are used for a warm-up learning rate. We initialized weights of BROS with those of BERT and trained it for 5 epochs on the IIT-CDIP dataset using 8 NVIDIA Tesla V100 32GB GPUs.

During fine-tuning, the learning rate is set to 5e-5. The batch size is set to 16 for all tasks. The number of training epochs or steps is as follows: 100 epochs for FUNSD, 1K steps for SROIE* and CORD, and 7.5 epochs for SciTSR.

### Experiment Results

To evaluate the performance of the model, we first conduct experiments using the given order of text blocks in the dataset. Then, we verify the robustness of the model against two important challenges in the KIE tasks, which are the dependency about the order of text blocks and learning from a few training examples.

Over our experiments, we report the scores of LayoutLM and LayoutLM\textsubscript{v2} using the models published by the authors\textsuperscript{4} and denote them as LayoutLM* and LayoutLM\textsubscript{v2}*. We report the mean (and optionally the standard deviation) of the results using the 5 different random seeds.

### With the Order Information of Text Blocks

Table 3 summarizes the results for the FUNSD EE task reported by the previous best. Interestingly, BROS provides better or similar performances compared to the multi-modal models incorporating additional visual (Image\textsuperscript{v}) or hierarchical (Cell\textsuperscript{h}) information. In other words, although BROS does not require extra computations and parameters to process additional features, BROS can achieve better or comparable performances. Table 4 shows the F1 scores on three EE and EL tasks with the order of text blocks given in the dataset. For EE tasks, all models utilize BIO tagger that captures spans of text blocks to represent key entities in documents. For EL tasks, SPADE decoder is used to identify relationships between entities not placed sequentially in a series of text blocks. In all cases, BERT performs the worst because those tasks require understanding texts in 2D space, but BERT only encodes 1D sequential information. LayoutLM*.

---

\textsuperscript{1}https://ir.nist.gov/cdip/
\textsuperscript{2}https://www.cs.cmu.edu/ aharley/rvl-cdip/
\textsuperscript{3}https://clova.ai/ocr

---

![Image](https://example.com/image1.png)

**Figure 5:** Examples of EE and EL tasks. In (a), the colored blocks represent key entities. In (b), the red arrows show the hierarchical relationships between the entities.

---

| Dataset | Types | Tasks | # Images |
|---------|-------|-------|----------|
| FUNSD   | Forms | EE, EL| Train 149, Test 50 |
| SROIE\textsuperscript{*} | Receipts | EE | Train 526, Test 100 |
| CORD    | Receipts | EE, EL | Train 900, Val 100, Test 100 |
| SciTSR  | Tables  | EL   | Train 12,000, Test 3,000 |

\textsuperscript{*} modified version of SROIE. See details in Appendix.

---

### Experiments

#### Experiment Settings

For pre-training, IIT-CDIP Test Collection 1.0\textsuperscript{1} (Lewis et al. 2006), which consists of approximately 11M document images, is used but 400K of RVL-CDIP dataset\textsuperscript{2} (Harley, Ufkes, and Derpanis 2015) are excluded following LayoutLM. To obtain text blocks from document images, CLOVA OCR API\textsuperscript{3} was applied. We observed no difference in performance depending on the OCR engine; LayoutLM
Table 3: Performance comparison on the FUNSD EE task. Bold indicates the best performance among models using only text and layout, and underline represents the best one. *Image* and *Cell* denote additional visual and hierarchical information, respectively. Our methods are repeatedly evaluated five times and the values of other methods are the reported scores.

Table 4: Performance comparisons on three EE and EL tasks with the order information of text blocks.

Table 5: Performance comparisons on three EE and EL tasks without the order information of text blocks.

Table 6: Comparison of FUNSD EE performances according to sorting methods.

Without the Order Information of Text Blocks As we mentioned in previous section, we introduce the permuted KIE benchmarks lost the orders of text blocks by shuffling the provided orders. To solve EE and EL tasks without the order information, we employ the SPADE decoder for all tasks. Table 5 shows the comparison results. p-F, p-S, p-C, and p-Sci refer to p-FUNSD, p-SROIE*, p-CORD, and p-SciTSR, respectively. BERT, which does not employ any spatial information of text blocks, shows the worst results on the orderless conditions. By being aware of the spatial-
Figure 6: Performance comparisons according to the amount of fine-tuning data. Each point represents the result of fine-tuning using from 10% to 100% of training data.

| Dataset | # Data | BERT   | LayoutLM* | LayoutLMv2* | BROS |
|---------|--------|--------|-----------|-------------|------|
| FUNSD   | 5      | 31.51  | 48.23     | 64.26       | 68.35|
| EE      | 10     | 40.46  | 62.50     | 69.92       | 72.60|
| FUNSD   | 5      | 14.65  | 21.38     | 7.32        | 31.11|
| EL      | 10     | 14.88  | 21.48     | 13.99       | 39.17|

Table 7: Results of training with 5 and 10 examples.

Learning from Few Training Examples One of the advantages of pre-trained models is that it shows effective transfer learning performance even with a few training examples (Devlin et al. 2019). Since collecting fine-tuning data requires a lot of resource, achieving high performance with a small number of training examples is important.

Figure 6 shows the results of the FUNSD KIE tasks by varying the amount of training examples from 10% to 100% during fine-tuning. In all models, performances tend to increase as the ratio of training data increased. In both tasks, BROS shows the best performances regardless of the number of training samples.

To further test extreme cases, we conduct experiments using only 5 and 10 training examples. Table 7 shows the results of the FUNSD KIE tasks. We fine-tune models for 100 epochs with a batch size of 4. In all cases, BROS shows the best performances. The results prove the generalization ability of BROS even when there are very few training examples.

Ablation Study

We conduct ablation studies to investigate which component contributes to the performance improvement. For the ablation studies, we utilize LayoutLM that is our own implementation of LayoutLM for fair comparisons under the same experimental settings. All models in these studies are pre-trained for 1 epoch.

Table 8 provides performance changes from adding our proposed components. When applying our proposed positional encoding to LayoutLM, the performances consistently increase with huge margins of 3.62pp on average over all tasks. Independently, our extension on pre-training objectives solely provides 1.14pp of performance improvement on average. By utilizing both, BROS provides the best performances with margins of 5.10pp on average. This ablation study proves that each component of BROS solely contributes to performance improvements as well as their combination provides better results.

Table 9 compares three positional encoding methods: absolute position in LayoutLM, relative position in LayoutLMv2, and ours. Relative position methods perform better than absolute one and the performance gap becomes larger in EL tasks. And among them, our method shows the best results.

Conclusion

We propose a pre-trained language model, BROS, which focuses on modeling text and layout features for effective key information extraction from documents. By encoding texts in 2D space with their relative positions and pre-training the model with the area-masking strategy, BROS shows superior performance without relying on any additional visual features. In addition, under the two real-world settings—imprecise text serialization and small amount of training examples—BROS shows robust performance while other models show significant performance degradation.
Acknowledgments

We thank many colleagues at NAVER CLOVA for their help, in particular Yoonsik Kim, Moonbin Yim, Han-cheol Cho, Bado Lee, Seunghyun Park, and Youngmin Baek for useful discussions.

References

Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Mannmatha, R. 2021. DocFormer: End-to-End Transformer for Document Understanding. arXiv preprint arXiv:2106.11539.

Chi, Z.; Huang, H.; Xu, H.-D.; Yu, H.; Yin, W.; and Mao, X.-L. 2019. Complicated table structure recognition. arXiv preprint arXiv:1908.04729.

Clausner, C.; Pletschacher, S.; and Antonacopoulos, A. 2013. The significance of reading order in document recognition and its evaluation. In 2013 12th International Conference on Document Analysis and Recognition (ICDAR), 688–692. IEEE.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J. G.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL).

Denk, T. I.; and Reisswig, C. 2019. BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. In Workshop on Document Intelligence at NeurIPS 2019.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long and Short Papers), 4171–4186.

Harley, A. W.; Uğkès, A.; and Derpanis, K. G. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), 991–995.

Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. ICDAR2019 competition on scanned receipt ocr and information extraction. In Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR), 1516–1520. IEEE.

Hwang, W.; Kim, S.; Seo, M.; Yim, J.; Park, S.; Park, S.; Lee, J.; Lee, B.; and Lee, H. 2019. Post-OCR parsing: building simple and robust parser via BIO tagging. In Workshop on Document Intelligence at NeurIPS 2019.

Hwang, W.; Yim, J.; Park, S.; Yang, S.; and Seo, M. 2021. Spatial Dependency Parsing for Semi-Structured Document Information Extraction. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 330–343.

Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. FUNSD: A dataset for form understanding in noisy scanned documents. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 2, 1–6. IEEE.

Joshi, M.; Chen, D.; Liu, Y.; Weld, D. S.; Zettlemoyer, L.; and Levy, O. 2020. SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics (TACL), 8: 64–77.

Lewis, D.; Agam, G.; Argamon, S.; Frieder, O.; Grossman, D.; and Heard, J. 2006. Building a test collection for complex document information processing. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR), 665–666.

Li, C.; Bi, B.; Yan, M.; Wang, W.; Huang, S.; Huang, F.; and Si, L. 2021a. StructuralLM: Structural Pre-training for Form Understanding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), 6309–6318.

Li, L.; Gao, F.; Bu, J.; Wang, Y.; Yu, Z.; and Zheng, Q. 2020. An End-to-End OCR Text Re-organization Sequence Learning for Rich-text Detail Image Comprehension. In Proceedings of the 16th European Conference on Computer Vision (ECCV).

Li, P.; Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Jain, R.; Manjunatha, V.; and Liu, H. 2021b. SelfDoc: Self-Supervised Document Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5652–5660.

Li, Y.; Qian, Y.; Yu, Y.; Qin, X.; Zhang, C.; Liu, Y.; Yao, K.; Han, J.; Liu, J.; and Ding, E. 2021c. StrucTexT: Structured Text Understanding with Multi-Modal Transformers. arXiv preprint arXiv:2108.02923.

Liu, X.; Gao, F.; Zhang, Q.; and Zhao, H. 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 (NAACL-HLT), Volume 2 (Industry Papers), 32–39.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In Proceedings of the 7th International Conference on Learning Representations (ICLR).

Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; and Lee, H. 2019. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In Workshop on Document Intelligence at NeurIPS 2019.

Powalski, R.; Borchmann, L.; Jurkiewicz, D.; Dwojak, T.; Pietruszka, M.; and Palka, G. 2021. Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer. arXiv preprint arXiv:2102.09550.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30 (NeurIPS), 5998–6008.

Wang, Z.; Xu, Y.; Cui, L.; Shang, J.; and Wei, F. 2021. LayoutReader: Pre-training of Text and Layout for Reading Order Detection. arXiv:2108.11591.
Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 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), 1192–1200.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), 2579–2591.