LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding

Yang Xu[1*], **Yiheng Xu**[2*], **Tengchao Lv**[2*], **Lei Cui**[2], **Furu Wei**[2], **Guoxin Wang**[3], **Yijuan Lu**[3], **Dinei Florencio**[3], **Cha Zhang**[3], **Wanxiang Che**[1], **Min Zhang**[4], **Lidong Zhou**[2]
[1]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology
[2]Microsoft Research Asia [3]Microsoft Azure AI [4]Soochow University
[1]{yxu,car}@ir.hit.edu.cn,
[2]{v-yixu,v-telv,lecu,fuwei,lidongz}@microsoft.com,
[3]{guow,yijlu,dinei,chazhang}@microsoft.com [4]minzhang@suda.edu.cn

Abstract

Pre-training of text and layout has proved effective in a variety of visually-rich document understanding tasks due to its effective model architecture and the advantage of large-scale unlabeled scanned/digital-born documents. We propose **LayoutLMv2** architecture with new pre-training tasks to model the interaction among text, layout, and image in a single multi-modal framework. Specifically, with a two-stream multi-modal Transformer encoder, LayoutLMv2 uses not only the existing masked visual-language modeling task but also the new text-image alignment and text-image matching tasks, which make it better capture the cross-modality interaction in the pre-training stage. Meanwhile, it also integrates a spatial-aware self-attention mechanism into the Transformer architecture so that the model can fully understand the relative positional relationship among different text blocks. Experiment results show that LayoutLMv2 outperforms LayoutLM by a large margin and achieves new state-of-the-art results on a wide variety of downstream visually-rich document understanding tasks, including FUNSD (0.7895 → 0.8420), CORD (0.9493 → 0.9601), SROIE (0.9524 → 0.9781), Kleister-NDA (0.8340 → 0.8520), RVL-CDIP (0.9443 → 0.9564), and DocVQA (0.7295 → 0.8672). We made our model and code publicly available at https://aka.ms/layoutlmv2.

1 Introduction

Visually-rich Document Understanding (VrDU) aims to analyze scanned/digital-born business documents (images of invoices, forms in PDF format, etc.) where structured information can be automatically extracted and organized for many business applications. Distinct from conventional information extraction tasks, the VrDU task relies on not only textual information but also visual and layout information that is vital for visually-rich documents. Different types of documents indicate that the text fields of interest located at different positions within the document, which is often determined by the style and format of each type as well as the document content. Therefore, to accurately recognize the text fields of interest, it is inevitable to take advantage of the cross-modality nature of visually-rich documents, where the textual, visual, and layout information should be jointly modeled and learned end-to-end in a single framework.

The recent progress of VrDU lies primarily in two directions. The first direction is usually built on the shallow fusion between textual and visual/layout/style information (Yang et al., 2017; Liu et al., 2019; Sarkhel and Nandi, 2019; Yu et al., 2020; Majumder et al., 2020; Wei et al., 2020; Zhang et al., 2020). These approaches leverage the pre-trained NLP and CV models individually and combine the information from multiple modalities for supervised learning. Although good performance has been achieved, the domain knowledge of one document type cannot be easily transferred into another, so that these models often need to be re-trained once the document type is changed. Thereby the local invariance in general document layout (key-value pairs in a left-right layout, tables in a grid layout, etc.) cannot be fully exploited. To this end, the second direction relies on the deep fusion among textual, visual, and layout information from a great number of unlabeled documents in different domains, where pre-training techniques play an important role in learning the cross-modality interaction in an end-to-end fashion (Lockard et al., 2020; Xu et al., 2020). In this way, the pre-trained

*Equal contributions during internship at MSRA

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2579–2591
August 1–6, 2021.

models absorb cross-modal knowledge from different document types, where the local invariance among these layouts and styles is preserved. Furthermore, when the model needs to be transferred into another domain with different document formats, only a few labeled samples would be sufficient to fine-tune the generic model in order to achieve state-of-the-art accuracy. Therefore, the proposed model in this paper follows the second direction, and we explore how to further improve the pre-training strategies for the VrDU tasks.

In this paper, we present an improved version of LayoutLM (Xu et al., 2020), aka **LayoutLMv2**. Different from the vanilla LayoutLM model where visual embeddings are combined in the fine-tuning stage, we integrate the visual information in the pre-training stage in LayoutLMv2 by taking advantage of the Transformer architecture to learn the cross-modality interaction between visual and textual information. In addition, inspired by the 1-D relative position representations (Shaw et al., 2018; Raffel et al., 2020; Bao et al., 2020), we propose the spatial-aware self-attention mechanism for LayoutLMv2, which involves a 2-D relative position representation for token pairs. Different from the absolute 2-D position embeddings that LayoutLM uses to model the page layout, the relative position embeddings explicitly provide a broader view for the contextual spatial modeling. For the pre-training strategies, we use two new training objectives for LayoutLMv2 in addition to the masked visual-language modeling. The first is the proposed text-image alignment strategy, which aligns the text lines and the corresponding image regions. The second is the text-image matching strategy popular in previous vision-language pre-training models (Tan and Bansal, 2019; Lu et al., 2019; Su et al., 2020; Chen et al., 2020; Sun et al., 2019), where the model learns whether the document image and textual content are correlated.

We select six publicly available benchmark datasets as the downstream tasks to evaluate the performance of the pre-trained LayoutLMv2 model, which are the FUNSD dataset (Jaume et al., 2019) for form understanding, the CORD dataset (Park et al., 2019) and the SROIE dataset (Huang et al., 2019) for receipt understanding, the Kleister-NDA dataset (Graliński et al., 2020) for long document understanding with a complex layout, the RVL-CDIP dataset (Harley et al., 2015) for document image classification, and the DocVQA dataset (Mathew et al., 2021) for visual question answering on document images. Experiment results show that the LayoutLMv2 model significantly outperforms strong baselines, including the vanilla LayoutLM, and achieves new state-of-the-art results in all of these tasks.

The contributions of this paper are summarized as follows:

- We propose a multi-modal Transformer model to integrate the document text, layout, and visual information in the pre-training stage, which learns the cross-modal interaction end-to-end in a single framework. Meanwhile, a spatial-aware self-attention mechanism is integrated into the Transformer architecture.
- In addition to the masked visual-language model, we add text-image alignment and text-image matching as the new pre-training strategies to enforce the alignment among different modalities.
- LayoutLMv2 significantly outperforms and achieves new SOTA results not only on the conventional VrDU tasks but also on the VQA task for document images, which demonstrates the great potential for the multi-modal pre-training for VrDU.

2 Approach

In this section, we will introduce the model architecture and the multi-modal pre-training tasks of LayoutLMv2, which is illustrated in Figure 1.

2.1 Model Architecture

We build a multi-modal Transformer architecture as the backbone of LayoutLMv2, which takes text, visual, and layout information as input to establish deep cross-modal interactions. We also introduce a spatial-aware self-attention mechanism to the model architecture for better modeling the document layout. Detailed descriptions of the model are as follows.

Text Embedding Following the common practice, we use WordPiece (Wu et al., 2016) to tokenize the OCR text sequence and assign each token to a certain segment $s_i \in \{\texttt{[A]}, \texttt{[B]}\}$. Then, we add `[CLS]` at the beginning of the sequence and `[SEP]` at the end of each text segment. Extra `[PAD]` tokens are appended to the end so that the

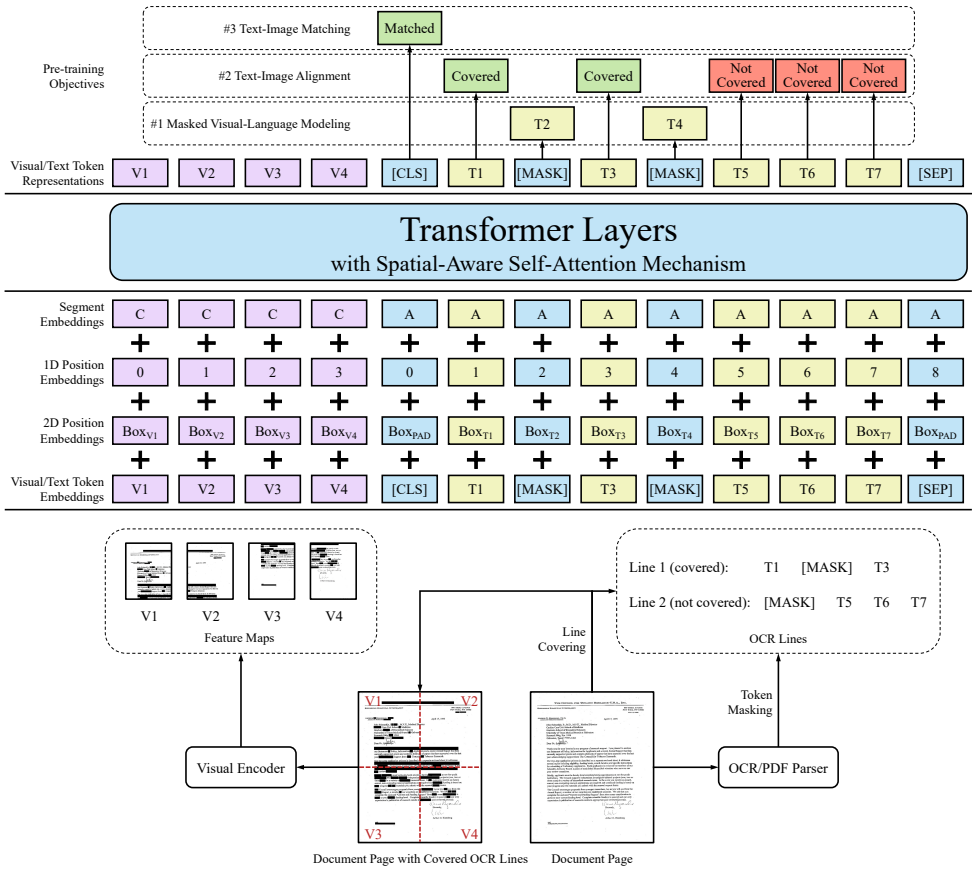

Figure 1: An illustration of the model architecture and pre-training strategies for LayoutLMv2

final sequence's length is exactly the maximum sequence length L. The final text embedding is the sum of three embeddings. Token embedding represents the token itself, 1D positional embedding represents the token index, and segment embedding is used to distinguish different text segments. Formally, we have the i-th ($0 \leq i < L$) text embedding

$$\mathbf{t}_i = \text{TokEmb}(w_i) + \text{PosEmb1D}(i) + \text{SegEmb}(s_i)$$

Visual Embedding Although all information we need is contained in the page image, the model has difficulty capturing detailed features in a single information-rich representation of the entire page. Therefore, we leverage the output feature map of a CNN-based visual encoder, which converts the page image to a fixed-length sequence. We use ResNeXt-FPN (Xie et al., 2017; Lin et al., 2017) architecture as the backbone of the visual encoder, whose parameters can be updated through backpropagation.

Given a document page image I, it is resized to 224×224 then fed into the visual backbone. After that, the output feature map is average-pooled to a fixed size with the width being W and height being H. Next, it is flattened into a visual embedding sequence of length $W \times H$. The sequence is named $\text{VisTokEmb}(I)$. A linear projection layer is then applied to each visual token embedding to unify the dimensionality with the text embeddings. Since the CNN-based visual backbone cannot capture the positional information, we also add a 1D positional embedding to these visual token embeddings. The 1D positional embedding is shared with the text embedding layer. For the segment embedding, we attach all visual tokens to the visual segment `[C]`. The i-th ($0 \leq i < WH$) visual embedding can be represented as

$$\begin{aligned}\mathbf{v}_i = &\text{Proj}\big(\text{VisTokEmb}(I)_i\big) \\ &+ \text{PosEmb1D}(i) + \text{SegEmb}(\texttt{[C]})\end{aligned}$$

Layout Embedding The layout embedding layer is for embedding the spatial layout information represented by axis-aligned token bounding boxes from the OCR results, in which box width and height together with corner coordinates are identified. Following the vanilla LayoutLM, we normalize and discretize all coordinates to integers in

the range $[0, 1000]$, and use two embedding layers to embed x-axis features and y-axis features separately. Given the normalized bounding box of the i-th ($0 \leq i < WH + L$) text/visual token $\text{box}_i = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, width, height)$, the layout embedding layer concatenates six bounding box features to construct a token-level 2D positional embedding, aka the layout embedding

$$\mathbf{l}_i = \text{Concat}\big(\text{PosEmb2D}_\text{x}(x_{\min}, x_{\max}, width), \text{PosEmb2D}_\text{y}(y_{\min}, y_{\max}, height)\big)$$

Note that CNNs perform local transformation, thus the visual token embeddings can be mapped back to image regions one by one with neither overlap nor omission. When calculating bounding boxes, the visual tokens can be treated as evenly divided grids. An empty bounding box $\text{box}_\text{PAD} = (0, 0, 0, 0, 0, 0)$ is attached to special tokens `[CLS]`, `[SEP]` and `[PAD]`.

Multi-modal Encoder with Spatial-Aware Self-Attention Mechanism The encoder concatenates visual embeddings $\{\mathbf{v}_0, ..., \mathbf{v}_{WH-1}\}$ and text embeddings $\{\mathbf{t}_0, ..., \mathbf{t}_{L-1}\}$ to a unified sequence and fuses spatial information by adding the layout embeddings to get the i-th ($0 \leq i < WH + L$) first layer input

$$\mathbf{x}_i^{(0)} = X_i + \mathbf{l}_i, \text{ where}$$
$$X = \{\mathbf{v}_0, ..., \mathbf{v}_{WH-1}, \mathbf{t}_0, ..., \mathbf{t}_{L-1}\}$$

Following the architecture of Transformer, we build our multi-modal encoder with a stack of multi-head self-attention layers followed by a feed-forward network. However, the original self-attention mechanism can only implicitly capture the relationship between the input tokens with the absolute position hints. In order to efficiently model local invariance in the document layout, it is necessary to insert relative position information explicitly. Therefore, we introduce the spatial-aware self-attention mechanism into the self-attention layers. For simplicity, the following description is for a single head in a single self-attention layer with hidden size of d_{head} and projection matrics $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$. The original self-attention mechanism captures the correlation between query $\mathbf{x}_i$ and key $\mathbf{x}_j$ by projecting the two vectors and calculating the attention score

$$\alpha_{ij} = \frac{1}{\sqrt{d_{head}}} \left(\mathbf{x}_i \mathbf{W}^Q\right) \left(\mathbf{x}_j \mathbf{W}^K\right)^\intercal$$

Considering the large range of positions, we model the semantic relative position and spatial relative position as bias terms to prevent adding too many parameters. Similar practice has been shown effective on text-only Transformer architectures (Raffel et al., 2020; Bao et al., 2020). Let $\mathbf{b}^{(\text{1D})}$, $\mathbf{b}^{(\text{2D}_\text{x})}$ and $\mathbf{b}^{(\text{2D}_\text{y})}$ denote the learnable 1D and 2D relative position biases respectively. The biases are different among attention heads but shared in all encoder layers. Assuming (x_i, y_i) anchors the top left corner coordinates of the i-th bounding box, we obtain the spatial-aware attention score

$$\alpha'_{ij} = \alpha_{ij} + \mathbf{b}^{(\text{1D})}_{j-i} + \mathbf{b}^{(\text{2D}_\text{x})}_{x_j - x_i} + \mathbf{b}^{(\text{2D}_\text{y})}_{y_j - y_i}$$

Finally, the output vectors are represented as the weighted average of all the projected value vectors with respect to normalized spatial-aware attention scores

$$\mathbf{h}_i = \sum_j \frac{\exp\left(\alpha'_{ij}\right)}{\sum_k \exp\left(\alpha'_{ik}\right)} \mathbf{x}_j \mathbf{W}^V$$

2.2 Pre-training Tasks

Masked Visual-Language Modeling Similar to the vanilla LayoutLM, we use the Masked Visual-Language Modeling (MVLM) to make the model learn better in the language side with the cross-modality clues. We randomly mask some text tokens and ask the model to recover the masked tokens. Meanwhile, the layout information remains unchanged, which means the model knows each masked token's location on the page. The output representations of masked tokens from the encoder are fed into a classifier over the whole vocabulary, driven by a cross-entropy loss. To avoid visual clue leakage, we mask image regions corresponding to masked tokens on the raw page image input before feeding it into the visual encoder.

Text-Image Alignment To help the model learn the spatial location correspondence between image and coordinates of bounding boxes, we propose the Text-Image Alignment (TIA) as a fine-grained cross-modality alignment task. In the TIA task, some tokens lines are randomly selected, and their image regions are covered on the document image. We call this operation covering to avoid confusion with the masking operation in MVLM. During pre-training, a classification layer is built above the encoder outputs. This layer predicts a label for each text token depending on whether it is covered,

i.e., [Covered] or [Not Covered], and computes the binary cross-entropy loss. Considering the input image's resolution is limited, and some document elements like signs and bars in a figure may look like covered text regions, the task of finding a word-sized covered image region can be noisy. Thus, the covering operation is performed at the line-level. When MVLM and TIA are performed simultaneously, TIA losses of the tokens masked in MVLM are not taken into account. This prevents the model from learning the useless but straightforward correspondence from [MASK] to [Covered].

Text-Image Matching Furthermore, a coarse-grained cross-modality alignment task, Text-Image Matching (TIM) is applied to help the model learn the correspondence between document image and textual content. We feed the output representation at [CLS] into a classifier to predict whether the image and text are from the same document page. Regular inputs are positive samples. To construct a negative sample, an image is either replaced by a page image from another document or dropped. To prevent the model from cheating by finding task features, we perform the same masking and covering operations to images in negative samples. The TIA target labels are all set to [Covered] in negative samples. We apply the binary cross-entropy loss in the optimization process.

3 Experiments

3.1 Data

In order to pre-train and evaluate LayoutLMv2 models, we select datasets in a wide range from the visually-rich document understanding area. Following LayoutLM, we use IIT-CDIP Test Collection (Lewis et al., 2006) as the pre-training dataset. Six datasets are used as down-stream tasks. The FUNSD (Jaume et al., 2019), CORD (Park et al., 2019), SROIE (Huang et al., 2019) and Kleister-NDA (Graliński et al., 2020) datasets define entity extraction tasks that aim to extract the value of a set of pre-defined keys, which we formalize as a sequential labeling task. RVL-CDIP (Harley et al., 2015) is for document image classification. DocVQA (Mathew et al., 2021), as the name suggests, is a dataset for visual question answering on document images. Statistics of datasets are shown in Table 1. Refer to the Appendix for details.

Dataset	# of keys or categories	# of examples (train/dev/test)
IIT-CDIP	–	11M/0/0
FUNSD	4	149/0/50
CORD	30	800/100/100
SROIE	4	626/0/347
Kleister-NDA	4	254/83/203
RVL-CDIP	16	320K/4K/4K
DocVQA	–	39K/5K/5K

Table 1: Statistics of datasets

3.2 Settings

Following the typical pre-training and fine-tuning strategy, we update all parameters including the visual encoder layers, and train whole models end-to-end for all the settings. Training details can be found in the Appendix.

Pre-training LayoutLMv2 We train LayoutLMv2 models with two different parameter sizes. We use a 12-layer 12-head Transformer encoder and set hidden size $d = 768$ in LayoutLMv2$_{\text{BASE}}$. While in the LayoutLMv2$_{\text{LARGE}}$, the encoder has 24 Transformer layers with 16 heads and $d = 1024$. Visual backbones in the two models are based on the same ResNeXt101-FPN architecture. The numbers of parameters are 200M and 426M approximately for LayoutLMv2$_{\text{BASE}}$ and LayoutLMv2$_{\text{LARGE}}$, respectively.

For the encoder along with the text embedding layer, LayoutLMv2 uses the same architecture as UniLMv2 (Bao et al., 2020), thus it is initialized from UniLMv2. For the ResNeXt-FPN part in the visual embedding layer, the backbone of a Mask-RCNN (He et al., 2017) model trained on PubLayNet (Zhong et al., 2019) is leveraged.[1] The rest of the parameters in the model are randomly initialized.

During pre-training, we sample pages from the IIT-CDIP dataset and select a random sliding window of the text sequence if the sample is too long. We set the maximum sequence length $L = 512$ and assign all text tokens to the segment [A]. The output shape of the average pooling layer is set to $W = H = 7$, so that it transforms the feature map into 49 visual tokens. In MVLM, 15% text tokens are masked among which 80% are replaced by a special token [MASK], 10% are replaced by a random token sampled from the whole vocabulary, and

[1] "MaskRCNN ResNeXt101_32x8d FPN 3X" setting in https://github.com/hpanwar08/detectron2

Model	FUNSD	CORD	SROIE	Kleister-NDA
$\text{BERT}_{\text{BASE}}$	0.6026	0.8968	0.9099	0.7790
$\text{UniLMv2}_{\text{BASE}}$	0.6648	0.9092	0.9459	0.7950
$\text{BERT}_{\text{LARGE}}$	0.6563	0.9025	0.9200	0.7910
$\text{UniLMv2}_{\text{LARGE}}$	0.7072	0.9205	0.9488	0.8180
$\text{LayoutLM}_{\text{BASE}}$	0.7866	0.9472	0.9438	0.8270
$\text{LayoutLM}_{\text{LARGE}}$	0.7895	0.9493	0.9524	0.8340
$\text{LayoutLMv2}_{\text{BASE}}$	0.8276	0.9495	0.9625	0.8330
$\text{LayoutLMv2}_{\text{LARGE}}$	**0.8420**	**0.9601**	**0.9781**	**0.8520**
BROS (Hong et al., 2021)	0.8121	0.9536	0.9548	–
SPADE (Hwang et al., 2020)	–	0.9150	–	–
PICK (Yu et al., 2020)	–	–	0.9612	–
TRIE (Zhang et al., 2020)	–	–	0.9618	–
Top-1 on SROIE Leaderboard (until 2020-12-24)	–	–	0.9767	–
$\text{RoBERTa}_{\text{BASE}}$ in (Graliński et al., 2020)	–	–	–	0.7930

Table 2: Entity-level F1 scores of the four entity extraction tasks: FUNSD, CORD, SROIE and Kleister-NDA. Detailed per-task results are in the Appendix.

10% remains the same. In TIA, 15% of the lines are covered. In TIM, 15% images are replaced, and 5% are dropped.

Fine-tuning LayoutLMv2 We use the `[CLS]` output along with pooled visual token representations as global features in the document-level classification task RVL-CDIP. For the extractive question answering task DocVQA and the other four entity extraction tasks, we follow common practice like (Devlin et al., 2019) and build task specified head layers over the text part of LayoutLMv2 outputs.

In the DocVQA paper, experiment results show that the BERT model fine-tuned on the SQuAD dataset (Rajpurkar et al., 2016) outperforms the original BERT model. Inspired by this fact, we add an extra setting, which is that we first fine-tune LayoutLMv2 on a question generation (QG) dataset followed by the DocVQA dataset. The QG dataset contains almost one million question-answer pairs generated by a generation model trained on the SQuAD dataset.

Baselines We select three baseline models in the experiments to compare LayoutLMv2 with the text-only pre-trained models as well as the vanilla LayoutLM model. Specifically, we compare LayoutLMv2 with BERT (Devlin et al., 2019), UniLMv2 (Bao et al., 2020), and LayoutLM (Xu et al., 2020) for all the experiment settings. We use the publicly available PyTorch models for BERT (Wolf et al., 2020) and LayoutLM, and use our in-house implementation for the UniLMv2 models. For each baseline approach, experiments are conducted using both the BASE and LARGE parameter settings.

3.3 Results

Entity Extraction Tasks Table 2 shows the model accuracy on the four datasets FUNSD, CORD, SROIE, and Kleister-NDA, which we regard as sequential labeling tasks evaluated using entity-level F1 score. We report the evaluation results of Kleister-NDA on the validation set because the ground-truth labels and the submission website for the test set are not available right now. For text-only models, the UniLMv2 models outperform the BERT models by a large margin in terms of the BASE and LARGE settings. For text+layout models, the LayoutLM family, especially the LayoutLMv2 models, brings significant performance improvement over the text-only baselines. Compared to the baselines, the LayoutLMv2 models are superior to the SPADE (Hwang et al., 2020) decoder method, as well as the text+layout pre-training approach BROS (Hong et al., 2021) that is built on the SPADE decoder, which demonstrates the effectiveness of our modeling approach. Moreover, with the same modal information, our LayoutLMv2 models also outperform existing multi-modal approaches PICK (Yu et al., 2020), TRIE (Zhang et al., 2020) and the previous top-1 method on the leaderboard,[2] confirming the effectiveness of our pre-training for text, layout, and visual information. The best performance on all the four datasets is achieved by

[2]Unpublished results, the leaderboard is available at https://rrc.cvc.uab.es/?ch=13&com=evaluation&task=3

Model	Accuracy
$\text{BERT}_{\text{BASE}}$	89.81%
$\text{UniLMv2}_{\text{BASE}}$	90.06%
$\text{BERT}_{\text{LARGE}}$	89.92%
$\text{UniLMv2}_{\text{LARGE}}$	90.20%
$\text{LayoutLM}_{\text{BASE}}$ (w/ image)	94.42%
$\text{LayoutLM}_{\text{LARGE}}$ (w/ image)	94.43%
$\text{LayoutLMv2}_{\text{BASE}}$	95.25%
$\text{LayoutLMv2}_{\text{LARGE}}$	**95.64%**
VGG-16 (Afzal et al., 2017)	90.97%
Single model (Das et al., 2018)	91.11%
Ensemble (Das et al., 2018)	92.21%
InceptionResNetV2 (Szegedy et al., 2017)	92.63%
LadderNet (Sarkhel and Nandi, 2019)	92.77%
Single model (Dauphinee et al., 2019)	93.03%
Ensemble (Dauphinee et al., 2019)	93.07%

Table 3: Classification accuracy on the RVL-CDIP dataset

Model	Fine-tuning set	ANLS
$\text{BERT}_{\text{BASE}}$	train	0.6354
$\text{UniLMv2}_{\text{BASE}}$	train	0.7134
$\text{BERT}_{\text{LARGE}}$	train	0.6768
$\text{UniLMv2}_{\text{LARGE}}$	train	0.7709
$\text{LayoutLM}_{\text{BASE}}$	train	0.6979
$\text{LayoutLM}_{\text{LARGE}}$	train	0.7259
$\text{LayoutLMv2}_{\text{BASE}}$	train	0.7808
$\text{LayoutLMv2}_{\text{LARGE}}$	train	0.8348
$\text{LayoutLMv2}_{\text{LARGE}}$	train + dev	0.8529
$\text{LayoutLMv2}_{\text{LARGE}}$ + QG	train + dev	**0.8672**
Top-1 (30 models ensemble) on DocVQA Leaderboard (until 2020-12-24)	-	0.8506

Table 4: ANLS score on the DocVQA dataset, "QG" denotes the data augmentation with the question generation dataset.

the $\text{LayoutLMv2}_{\text{LARGE}}$, which illustrates that the multi-modal pre-training in LayoutLMv2 learns better from the interactions from different modalities, thereby leading to the new SOTA on various document understanding tasks.

RVL-CDIP Table 3 shows the classification accuracy on the RVL-CDIP dataset, including text-only pre-trained models, the LayoutLM family as well as several image-based baseline models. As shown in the table, both the text and visual information are important to the document image classification task because document images are text-intensive and represented by a variety of layouts and formats. Therefore, we observed that the LayoutLM family outperforms those text-only or image-only models as it leverages the multi-modal information within the documents. Specifically, the $\text{LayoutLMv2}_{\text{LARGE}}$ model significantly improves the classification accuracy by more than 1.2% point over the previous SOTA results, which achieves an accuracy of 95.64%. This also verifies that the pre-trained LayoutLMv2 model benefits not only the information extraction tasks in document understanding but also the document image classification task through effective multi-model training.

DocVQA Table 4 lists the Average Normalized Levenshtein Similarity (ANLS) scores on the DocVQA dataset of text-only baselines, LayoutLM family models, and the previous top-1 on the leaderboard. With multi-modal pre-training, LayoutLMv2 models outperform LayoutLM models and text-only baselines by a large margin when fine-tuned on the train set. By using all data (train + dev) as the fine-tuning dataset, the $\text{LayoutLMv2}_{\text{LARGE}}$ single model outperforms the previous top-1 on the leaderboard which ensembles 30 models.[3] Under the setting of fine-tuning $\text{LayoutLMv2}_{\text{LARGE}}$ on a question generation dataset (QG) and the DocVQA dataset successively, the single model performance increases by more than 1.6% ANLS and achieves the new SOTA.

3.4 Ablation Studies

To fully understand the underlying impact of different components, we conduct an ablation study to explore the effect of visual information, the pre-training tasks, spatial-aware self-attention mechanism, as well as different text-side initialization models. Table 5 shows model performance on the DocVQA validation set. Under all the settings, we pre-train the models using all IIT-CDIP data for one epoch. The hyper-parameters are the same as those used to pre-train $\text{LayoutLMv2}_{\text{BASE}}$ in Section 3.2. "LayoutLM" denotes the vanilla LayoutLM architecture in (Xu et al., 2020), which can be regarded as a LayoutLMv2 architecture without visual module and spatial-aware self-attention mechanism. "X101-FPN" denotes the ResNeXt101-FPN visual backbone described in Section 3.2.

We first evaluate the effect of introducing visual information. From #1 to #2a, we add the visual module without changing the pre-training strategy, where results show that LayoutLMv2 pre-

[3]Unpublished results, the leaderboard is available at https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1

#	Model Architecture	Initialization	SASAM	MVLM	TIA	TIM	ANLS
1	$\text{LayoutLM}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$		✓			0.6841
2a	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$ + X101-FPN		✓			0.6915
2b	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$ + X101-FPN		✓	✓		0.7061
2c	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$ + X101-FPN		✓		✓	0.6955
2d	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$ + X101-FPN		✓	✓	✓	0.7124
3	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{BERT}_{\text{BASE}}$ + X101-FPN	✓	✓	✓	✓	0.7217
4	$\text{LayoutLMv2}_{\text{BASE}}$	$\text{UniLMv2}_{\text{BASE}}$ + X101-FPN	✓	✓	✓	✓	0.7421

Table 5: Ablation study on the DocVQA dataset, where ANLS scores on the validation set are reported. "SASAM" means the spatial-aware self-attention mechanism. "MVLM", "TIA" and "TIM" are the three pre-training tasks. All the models are trained using the whole pre-training dataset for one epoch with the BASE model size.

trained with only MVLM can leverage visual information effectively. Then, we compare the two cross-modality alignment pre-training tasks TIA and TIM. According to the four results in #2, both tasks improve the model performance substantially, and the proposed TIA benefits the model more than the commonly used TIM. Using both tasks together is more effective than using either one alone. According to this observation, we keep all the three pre-training tasks and introduce the spatial-aware self-attention mechanism (SASAM) to the model architecture. Compare the results #2d and #3, the proposed SASAM can further improve the model accuracy. Finally, in settings #3 and #4, we change the text-side initialization checkpoint from BERT to UniLMv2, and confirm that LayoutLMv2 benefits from the better initialization.

4 Related Work

In recent years, pre-training techniques have become popular in both NLP and CV areas, and have also been leveraged in the VrDU tasks.

Devlin et al. (2019) introduced a new language representation model called BERT, which is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers. Bao et al. (2020) propose to pre-train a unified language model for both autoencoding and partially autoregressive language modeling tasks using a novel training procedure, referred to as a pseudo-masked language model. Our multi-modal Transformer architecture and the MVLM pre-training strategy extend Transformer and MLM used in these work to leverage visual information.

Lu et al. (2019) proposed ViLBERT for learning task-agnostic joint representations of image content and natural language by extending the popular BERT architecture to a multi-modal two-stream model. Su et al. (2020) proposed VL-BERT that adopts the Transformer model as the backbone, and extends it to take both visual and linguistic embedded features as input. Different from these vision-language pre-training approaches, the visual part of LayoutLMv2 directly uses the feature map instead of pooled ROI features, and benefits from the new TIA pre-training task.

Xu et al. (2020) proposed LayoutLM to jointly model interactions between text and layout information across scanned document images, benefiting a great number of real-world document image understanding tasks such as information extraction from scanned documents. This work is a natural extension of the vanilla LayoutLM, which takes advantage of textual, layout, and visual information in a single multi-modal pre-training framework.

5 Conclusion

In this paper, we present a multi-modal pre-training approach for visually-rich document understanding tasks, aka LayoutLMv2. Distinct from existing methods for VrDU, the LayoutLMv2 model not only considers the text and layout information but also integrates the image information in the pre-training stage with a single multi-modal framework. Meanwhile, the spatial-aware self-attention mechanism is integrated into the Transformer architecture to capture the relative relationship among different bounding boxes. Furthermore, new pre-training objectives are also leveraged to enforce the learning of cross-modal interaction among different modalities. Experiment results on 6 different VrDU tasks have illustrated that the pre-trained LayoutLMv2 model has substantially outperformed the SOTA baselines in the document intelligence area, which greatly benefits a number of real-world document

understanding tasks.

For future research, we will further explore the network architecture as well as the pre-training strategies for the LayoutLM family. Meanwhile, we will also investigate the language expansion to make the multi-lingual LayoutLMv2 model available for different languages, especially the non-English areas around the world.

Acknowledgments

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 61772153.

References

Muhammad Zeshan Afzal, Andreas Kölsch, Sheraz Ahmed, and Marcus Liwicki. 2017. Cutting the error by half: Investigation of very deep cnn and advanced training strategies for document image classification. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:883–888.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR.

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 *ECCV*.

Arindam Das, Saikat Roy, and Ujjwal Bhattacharya. 2018. Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3180–3185.

Tyler Dauphinee, Nikunj Patel, and Mohammad Mehdi Rashidi. 2019. Modular multimodal architecture for document classification. *ArXiv*, abs/1912.04376.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

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

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society.

Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. {BROS}: A pre-trained language model for understanding texts in document.

Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. V. Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2020. Spatial dependency parsing for semi-structured document information extraction.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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), pages 32–39, Minneapolis, Minnesota. Association for Computational Linguistics.

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*, pages 8105–8117, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

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*, pages 6495–6504, Online. Association for Computational Linguistics.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ritesh Sarkhel and Arnab Nandi. 2019. Deterministic routing between layout abstractions for multi-scale classification of visually rich documents. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3360–3366. ijcai.org.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: pre-training of generic visual-linguistic representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7463–7472. IEEE.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4278–4284. AAAI Press.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Mengxi Wei, Yifan He, and Qiong Zhang. 2020. Robust layout-aware IE for visually rich documents with pre-trained language models. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2367–2376. ACM.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5987–5995. IEEE Computer Society.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4342–4351. IEEE Computer Society.

Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. 2020. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks.

Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. 2020. Trie: End-to-end text reading and information extraction for document understanding.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE.

Appendix

A Details of Datasets

Introduction to the dataset and task definitions along with the description of required data processing are presented as follows.

Pre-training Dataset Following LayoutLM, we pre-train LayoutLMv2 on the IIT-CDIP Test Collection (Lewis et al., 2006), which contains over 11 million scanned document pages. We extract text and corresponding word-level bounding boxes from document page images with the Microsoft Read API.[4]

FUNSD FUNSD (Jaume et al., 2019) is a dataset for form understanding in noisy scanned documents. It contains 199 real, fully annotated, scanned forms where 9,707 semantic entities are annotated above 31,485 words. The 199 samples are split into 149 for training and 50 for testing. The official OCR annotation is directly used with the layout information. The FUNSD dataset is suitable for a variety of tasks, where we focus on semantic entity labeling in this paper. Specifically, the task is assigning to each word a semantic entity label from a set of four predefined categories: question, answer, header, or other. The entity-level F1 score is used as the evaluation metric.

CORD We also evaluate our model on the receipt key information extraction dataset, i.e. the public available subset of CORD (Park et al., 2019). The dataset includes 800 receipts for the training set, 100 for the validation set, and 100 for the test set. A photo and a list of OCR annotations are equipped for each receipt. An ROI that encompasses the area of receipt region is provided along with each photo because there can be irrelevant things in the background. We only use the ROI as input instead of the raw photo. The dataset defines 30 fields under 4 categories and the task aims to label each word to the right field. The evaluation metric is entity-level F1. We use the official OCR annotations.

SROIE The SROIE dataset (Task 3) (Huang et al., 2019) aims to extract information from scanned receipts. There are 626 samples for training and 347 samples for testing in the dataset. The task is to extract values from each receipt of up to four predefined keys: company, date, address, or total. The evaluation metric is entity-level F1. We use the official OCR annotations and results on the test set are provided by the official evaluation site.

Kleister-NDA Kleister-NDA (Graliński et al., 2020) contains non-disclosure agreements collected from the EDGAR database, including 254 documents for training, 83 documents for validation, and 203 documents for testing. This task is defined to extract the values of four fixed keys. We get the entity-level F1 score from the official

[4] https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-recognizing-text

evaluation tools.[5] Words and bounding boxes are extracted from the raw PDF file. We use heuristics to locate entity spans because the normalized standard answers may not appear in the utterance. As the labeled answers are normalized into a canonical form, we apply post-processing heuristics to convert the extracted date information into the "YYYY-MM-DD" format, and company names into the abbreviations such as "LLC" and "Inc.".

RVL-CDIP RVL-CDIP (Harley et al., 2015) consists of 400,000 grayscale images, with 8:1:1 for the training set, validation set, and test set. A multi-class single-label classification task is defined on RVL-CDIP. The images are categorized into 16 classes, with 25,000 images per class. The evaluation metric is the overall classification accuracy. Text and layout information is extracted by Microsoft OCR.

DocVQA As a VQA dataset on the document understanding field, DocVQA (Mathew et al., 2021) consists of 50,000 questions defined on over 12,000 pages from a variety of documents. Pages are split into the training set, validation set, and test set with a ratio of about 8:1:1. The dataset is organized as a set of triples ⟨page image, questions, answers⟩. Thus, we use Microsoft Read API to extract text and bounding boxes from images. Heuristics are used to find given answers in the extracted text. The task is evaluated using an edit distance based metric ANLS (aka average normalized Levenshtein similarity). Given that human performance is about 98% ANLS on the test set, it is reasonable to assume that the found ground truth which reaches over 97% ANLS on training and validation sets is good enough to train a model. Results on the test set are provided by the official evaluation site.

B Model Training Details

Pre-training We pre-train LayoutLMv2 models using Adam optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019), with the learning rate of 2×10^{-5}, weight decay of 1×10^{-2},and $(\beta_1, \beta_2) = (0.9, 0.999)$. The learning rate is linearly warmed up over the first 10% steps then linearly decayed. LayoutLMv2$_{\text{BASE}}$ is trained with a batch size of 64 for 5 epochs, and LayoutLMv2$_{\text{LARGE}}$ is trained with a batch size of 2048 for 20 epochs on the IIT-CDIP dataset.

[5] https://gitlab.com/filipg/geval

Fine-tuning for Visual Question Answering We treat the DocVQA as an extractive QA task and build a token-level classifier on top of the text part of LayoutLMv2 output representations. Question tokens, context tokens and visual tokens are assigned to segment `[A]`, `[B]` and `[C]`, respectively. The maximum sequence length is set to $L = 384$.

Fine-tuning for Document Image Classification This task depends on high-level visual information, thereby we leverage the image features explicitly in the fine-tuning stage. We pool the visual embeddings into a global pre-encoder feature, and pool the visual part of LayoutLMv2 output representations into a global post-encoder feature. The pre and post-encoder features along with the `[CLS]` output feature are concatenated and fed into the final classification layer.

Fine-tuning for Sequential Labeling We formalize FUNSD, SROIE, CORD, and Kleister-NDA as the sequential labeling tasks. To fine-tune LayoutLMv2 models on these tasks, we build a token-level classification layer above the text part of the output representations to predict the BIO tags for each entity field.

C Detailed Experiment Results

Tables list per-task detailed results for the four entity extraction tasks, with Table 6 for FUNSD, Table 7 for CORD, Table 8 for SROIE, and Table 9 for Kleister-NDA.

Model	Precision	Recall	F1
$\text{BERT}_{\text{BASE}}$	0.5469	0.6710	0.6026
$\text{UniLMv2}_{\text{BASE}}$	0.6349	0.6975	0.6648
$\text{BERT}_{\text{LARGE}}$	0.6113	0.7085	0.6563
$\text{UniLMv2}_{\text{LARGE}}$	0.6780	0.7391	0.7072
$\text{LayoutLM}_{\text{BASE}}$	0.7597	0.8155	0.7866
$\text{LayoutLM}_{\text{LARGE}}$	0.7596	0.8219	0.7895
$\text{LayoutLMv2}_{\text{BASE}}$	0.8029	0.8539	0.8276
$\text{LayoutLMv2}_{\text{LARGE}}$	**0.8324**	**0.8519**	**0.8420**
BROS (Hong et al., 2021)	0.8056	0.8188	0.8121

Table 6: Model accuracy (entity-level Precision, Recall, F1) on the FUNSD dataset

Model	Precision	Recall	F1
$\text{BERT}_{\text{BASE}}$	0.8833	0.9107	0.8968
$\text{UniLMv2}_{\text{BASE}}$	0.8987	0.9198	0.9092
$\text{BERT}_{\text{LARGE}}$	0.8886	0.9168	0.9025
$\text{UniLMv2}_{\text{LARGE}}$	0.9123	0.9289	0.9205
$\text{LayoutLM}_{\text{BASE}}$	0.9437	0.9508	0.9472
$\text{LayoutLM}_{\text{LARGE}}$	0.9432	0.9554	0.9493
$\text{LayoutLMv2}_{\text{BASE}}$	0.9453	0.9539	0.9495
$\text{LayoutLMv2}_{\text{LARGE}}$	**0.9565**	**0.9637**	**0.9601**
SPADE (Hwang et al., 2020)	-	-	0.9150
BROS (Hong et al., 2021)	0.9558	0.9514	0.9536

Table 7: Model accuracy (entity-level Precision, Recall, F1) on the CORD dataset

Model	Precision	Recall	F1
$\text{BERT}_{\text{BASE}}$	0.9099	0.9099	0.9099
$\text{UniLMv2}_{\text{BASE}}$	0.9459	0.9459	0.9459
$\text{BERT}_{\text{LARGE}}$	0.9200	0.9200	0.9200
$\text{UniLMv2}_{\text{LARGE}}$	0.9488	0.9488	0.9488
$\text{LayoutLM}_{\text{BASE}}$	0.9438	0.9438	0.9438
$\text{LayoutLM}_{\text{LARGE}}$	0.9524	0.9524	0.9524
$\text{LayoutLMv2}_{\text{BASE}}$	0.9625	0.9625	0.9625
$\text{LayoutLMv2}_{\text{LARGE}}$	0.9661	0.9661	0.9661
$\text{LayoutLMv2}_{\text{LARGE}}$ (Excluding OCR mismatch)	**0.9904**	**0.9661**	**0.9781**
BROS (Hong et al., 2021)	0.9493	0.9603	0.9548
PICK (Yu et al., 2020)	0.9679	0.9546	0.9612
TRIE (Zhang et al., 2020)	-	-	0.9618
Top-1 on SROIE Leaderboard (Excluding OCR mismatch)	0.9889	0.9647	0.9767

Table 8: Model accuracy (entity-level Precision, Recall, F1) on the SROIE dataset (until 2020-12-24)

Model	F1
$\text{BERT}_{\text{BASE}}$	0.779
$\text{UniLMv2}_{\text{BASE}}$	0.795
$\text{BERT}_{\text{LARGE}}$	0.791
$\text{UniLMv2}_{\text{LARGE}}$	0.818
$\text{LayoutLM}_{\text{BASE}}$	0.827
$\text{LayoutLM}_{\text{LARGE}}$	0.834
$\text{LayoutLMv2}_{\text{BASE}}$	0.833
$\text{LayoutLMv2}_{\text{LARGE}}$	**0.852**
$\text{RoBERTa}_{\text{BASE}}$ in (Graliński et al., 2020)	0.793

Table 9: Model accuracy (entity-level F1) on the validation set of the Kleister-NDA dataset using the official evaluation toolkit