LayoutLLM: Large Language Model Instruction Tuning for Visually Rich Document Understanding

Masato Fujitake
FA Research, Fast Accounting Co., Ltd.
fujitake@fastaccounting.co.jp

Abstract

This paper proposes LayoutLLM, a more flexible document analysis method for understanding imaged documents. Visually Rich Document Understanding tasks, such as document image classification and information extraction, have gained significant attention due to their importance. Existing methods have been developed to enhance document comprehension by incorporating pre-training awareness of images, text, and layout structure. However, these methods require fine-tuning for each task and dataset, and the models are expensive to train and operate. To overcome this limitation, we propose a new LayoutLLM that integrates these with large-scale language models (LLMs). By leveraging the strengths of existing research in document image understanding and LLMs' superior language understanding capabilities, the proposed model, fine-tuned with multimodal instruction datasets, performs an understanding of document images in a single model. Our experiments demonstrate improvement over the baseline model in various document analysis tasks.

Keywords: Information Extraction, Language Model, Document Image Understanding

1. Introduction

Visual-rich Document Understanding (VrDU) focuses on analyzing document images, such as invoices, to extract and organize structured information automatically. Different documents have different styles, formats, and contents, so unlike traditional textual information extraction tasks, VrDU relies on both textual and visual information. Therefore, taking advantage of the multimodal nature of visually rich documents is essential. To this end, previous works, such as LayoutLMs (Huang et al., 2022; Xu et al., 2021, 2020), have proposed to acquire feature representations by jointly pre-training textual, visual, and layout information end-to-end in a single model, as shown in Figure 1. The process of fine-tuning is carried out on each task, as illustrated in Figure 1(a). However, this approach requires complex fine-tuning steps for each task and dataset, significantly increasing training and operational costs.

Large language models (LLMs) have gained a lot of attention due to their success in natural language processing tasks (Brown et al., 2020). They acquire linguistic knowledge by predicting the continuation of input sentences through pre-training on large amounts of the corpus (Radford et al., 2019). Then, a model can perform various tasks, such as translation and summarization, by fine-tuning the knowledge with responses to the input text. However, while they can perform various tasks through prompts, which are input instructions, they can only handle one-dimensional sequences of textual information. They must be improved to handle text with significant two-dimensional structure, such as document images.

We propose a new approach, LayoutLLM, which tackles the limitations of conventional models by combining the advantages of VrDU models and LLMs. As an encoder, it employs a model that excels in document layout understanding, while as a decoder, it uses LLMs that excel in language understanding. The proposed single model can flexibly perform multiple tasks by fine-tuning to multiple VrDU tasks, as shown in Figure 1(b). We evaluated the proposed method on various benchmarks, such as document image classification, information extraction, and document visual question-answering. Our experimental results confirm that LayoutLLM outperforms professionally tuned models in the VrDU task on several tasks and also improves performance on NLP tasks.

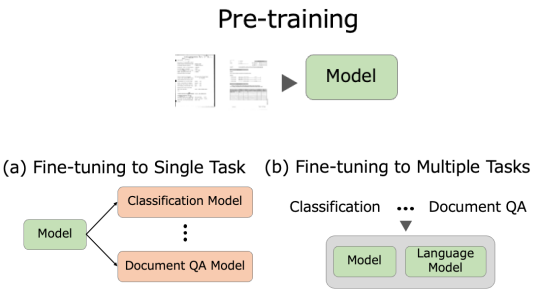

Figure 1: The overview of the existing and proposed method approaches. The current method is fine-tuned for each task after pre-training, as shown in (a). In contrast, our method is fine-tuned via LLMs to handle multiple tasks, as shown in (b).

2. Related Works

VrDU. Previously, language models (Devlin et al., 2018) challenged document image analysis using only optical character recognition (OCR) text (Fuji-

LREC-COLING 2024, pages 10219–10224

take, 2023b, 2024). However, a current approach integrates document images and OCR text to pre-train text, visual, and document layout, providing a more comprehensive understanding of documents. LayoutLM (Xu et al., 2020) combines 2D location information, image embedding, and text for pre-training, like masking language modeling. The improved models and pre-training techniques, such as LayoutLMv3 (Huang et al., 2022), have been proposed for higher accuracy (Xu et al., 2021; Lee et al., 2023). Text and document feature representations have been improved through multimodal encoders (Gu et al., 2021). A method introduced modeling documents as a collection of bounding box tokens (Garncarek et al., 2021). OCR-free models generate text output directly from document images for optimization (Kim et al., 2022; Li et al., 2022; Ye et al., 2023). UDOP (Tang et al., 2023) is a method that combines multiple modes and tasks into a single model with image reconstruction. However, it only works for VrDU tasks and cannot handle NLP tasks. Therefore, this work proposes a flexible framework for multi-domain NLP and VrDU tasks by using pre-trained models as encoders and fine-tuning them with LLMs.

LLMs. Large language models have been rapidly studied in recent years after the success of language models (Devlin et al., 2018). BERT proposed a pre-training method, masked language modeling, which learns bidirectional text representations and then fine-tunes them to the target task. GPT (Radford et al., 2018) also proposed a method for acquiring representations through next-word prediction pre-training. The succeeding research found that by successfully inputting prompts, the model can be adapted to various tasks without fine-tuning (Radford et al., 2019), and various large-scale language models have been proposed (Brown et al., 2020; Armengol-Estapé et al., 2022). In this study, we used Alpaca (Taori et al., 2023), based on the large language model Llama (Touvron et al., 2023), and fine-tuned with a dataset of 52K instructions and their responses.

3. Method

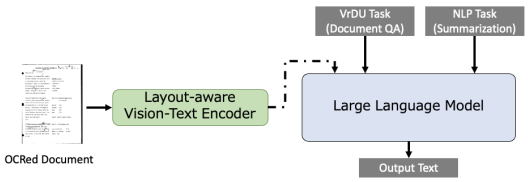

Figure 2: Architectural overview of the proposed LayoutLLM. It consists mainly of an encoder that encodes document images and a decoder that interprets tasks, and outputs.

Table 1: Prompt format for VrDU tasks.

Prompts
The previous information is about document images. Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: {instruction} ### Response:

Figure 2 shows an overview of the proposed method. Our method, LayoutLLM, consists of pre-trained VrDU models and LLMs. VrDU models, particularly LayoutLMv3 (Huang et al., 2022), handle visual and layout comprehension of documents. The LLMs, especially Llama (Touvron et al., 2023), interpret and analyze the document's textual content and the task with the language understanding capabilities, and output the results.

The proposed method is fine-tuned using VrDU and NLP tasks. As a training phase, the proposed method uses LayoutLMv3 as an encoder to process document images. It takes an OCRed document image, which is a document image with OCR text by some methods (AI, 2020; Fujitake, 2023a), as input and generates features. Next, the model receives the document features and subsequent VrDU task instructions as prompts, and inputs them to Llama. The response to the prompt is output in an autoregressive manner, and fine-tuning is performed using CrossEntropy in the same way as in standard LLMs training. We follow Alpaca (Taori et al., 2023), which fine-tunes Llama with Intruct-tuning for the NLP task. Precisely, we fine-tune the model to the NLP task to respond appropriately, such as summarizing, based on the input of text prompts. We do not use the VrDU model at this time and train in the same way as in Alpaca.

In the test phase, the proposed method uses the encoder to generate features for the VrDU task and input them to the LLM with the prompts, as in the training phase. In the case of the NLP task, only text prompts are used to output responses.

Encoder. OCRed textual and visual information from the document image is first encoded using the pre-trained LayoutLMv3 architecture. It captures the layout structure and text of the document, and generates features corresponding to the document. It is made into a 1D sequence with a maximum sequence length of 512 to be input to subsequent Llama. The maximum sequence is determined by the LayoutLMv3 configuration. In addition, one linear layer was used to correspond to the input dimension of Llama. The feature is input to Llama.

Decoder. The decoder uses Llama to receive input data and task instructions, and produces corresponding output. It is an auto-regressive language model using the optimized Transformer architecture, and a 7B parameters model was used in this

study. Document features and VrDU task instruction prompts are used as input for the VrDU tasks. More precisely, after tokenizing and encoding the task instruction prompts, the document features are first input to the LLM, followed by task instruction information. The feature is input at the same stage as the features after embedding the natural language prompts. The NLP task fine-tuning method follows Alpaca completely.

VrDU Prompts. Based on the document data, the task content is given to Llama by prompts to correspond to the target VrDU task. The prompts are made as consistent as possible with the Alpaca format, and the format is shown in Table 1. The "{instruction}" is a variable, the content of which is task-specific. For document classification, we use "Perform document classification. The classification labels are ..." In the case of document information extraction, we use "Perform document information extraction. The classification labels are... The output format is a set of extraction words and their labels, separated by commas. If multiple extraction targets exist, use \n as a separator and split the outputs." For document questions, we use "Perform document question answering. The question is that ..." Ground truth is created from each dataset to match the prompts' output.

4. Experiments

4.1. Dataset and Evaluation

LayoutLLM's performance was evaluated through experiments such as form understanding, receipt recognition, and document classification tasks. Unless stated otherwise, OCR text and bounding boxes are extracted by EacyOCR (AI, 2020).

Document Classification. Document classification predicts the category of each document image. RVL-CDIP (Harley et al., 2015) is used as the target dataset. This dataset comprises 320K/40K/40K training/validation/test images in 16 categories. Classification accuracies for the 16 categories are used to measure model performance.

Document Information Extraction. To extract information from documents, a model must predict the label for each semantic entity. We use the FUNSD (Jaume et al., 2019) and CORD (Park et al., 2019) datasets. FUNSD uses 149/50 noisy document images during training and testing. Each semantic entity includes a word list, label, and bounding box. The evaluation measure used is an entity-level F1 score for predicting a question, answer, header, or other. We use the OCR text and bounding boxes provided by the dataset.

The CORD dataset is a benchmark for receipt comprehension, with 626/247 receipts for training/testing, respectively. A model must recognize a list of text lines. The receipts are labeled with 30 entities grouped into four categories: company, date, address, and total. The metric is F1, and the task format is the same as FUNSD.

Document Visual Question Answering. We use a document understanding benchmark, DocVQA (Mathew et al., 2021). It consists of 50,000 questions defined on over 12,000 pages of documents. The dataset is organized into training, validation, and test sets, with a ratio of about 8:1:1. It contains an OCRed image page, questions, and answers. The task is evaluated using an ANLS, an edit distance-based metric measuring average normalized Levenshtein similarity.

4.2. Implementation Details

Our model used a pre-trained LayoutLMv3 large encoder and a pre-trained Llama-7B decoder with their official weights. We used VrDU task datasets and NLP task datasets to fine-tune the model. We created the VrDU task dataset using the dataset above and the prompt described in the proposed method section. For the NLP task dataset, we used the Alpaca dataset (Taori et al., 2023). Mini-batches were created during training, separating the VrDU and NLP tasks. It was because the NLP task doesn't require data input to the encoder. Mixing the two tasks prevented back-propagation. Encoder outputs and prompt inputs were consistently supplied in the same order for training and inference. The model is optimized on eight A100 GPUs with a batch size of 16. We follow Alpaca's learning process basically, using AdamW Optimizer (Loshchilov and Hutter, 2018), with a learning rate of 1e-5 and 20 epochs. Cosine learning rate scheduling was used, with a warmup ratio of 0.05 and weight decay of 0.01.

4.3. Main Results

Table 2 shows the performance of each dataset with several state-of-the-art methods.

Document Classification. Our approach results in new state-of-the-art accuracy, surpassing StructuralLM's previous record by 2.6 points without task-specific and special fine-tuning after pre-training. The improvement in accuracy can be attributed to two factors: improved linguistic ability using LLMs and learning multiple tasks simultaneously. The conventional method only considers linguistic context obtained during pre-training and fine-tuning with documents. In contrast, the proposed method takes advantage of a language model specialized for linguistic information, making it easier to classify documents based on their content.

Document Information Extraction. The proposed method has achieved outstanding results on both datasets, with scores of 95.3% and 98.6%

Table 2: Performance comparison with state-of-the-arts on FUNSD, CORD, and RVL-CDIP datasets. Modality V, T, L denote vision, text and layout.

Model	Modality	FUNSD	CORD	RVL-CDIP	Doc VQA
$BERT_{large}$ (Devlin et al., 2018)	T	65.6	90.3	89.9	67.5
DiT_{large} (Li et al., 2022)	V	–	–	92.7	–
Donut (Kim et al., 2022)	V	–	91.6	95.3	72.1
mPLUG-DocOwl (Ye et al., 2023)	V	–	–	–	62.2
$StructuralLM_{large}$ (Li et al., 2021)	T+L	85.1	–	96.2	83.9
$LayoutLM_{large}$ (Xu et al., 2020)	T+L	77.9	–	91.9	–
UniDoc (Gu et al., 2021)	V+T+L	87.9	96.9	95.1	–
LAMBERT (Garncarek et al., 2021)	T+L	–	96.1	–	–
$TILT_{large}$ (Powalski et al., 2021)	V+T+L	–	96.3	95.5	**87.1**
$LayoutLMv2_{large}$ (Xu et al., 2021)	V+T+L	84.2	96.0	95.6	78.8
$LayoutLMv3_{large}$ (Huang et al., 2022)	V+T+L	92.1	97.5	95.9	83.4
UDOP (Tang et al., 2023)	V+T+L	91.6	97.6	96.0	84.7
LayoutLLM (Ours)	V+T+L	**95.3**	**98.6**	**98.8**	86.9

Table 3: Component analysis.

Method	RVL-CDIP	CORD
Encoder-only (LayoutLMv3)	95.9	97.5
+ Decoder (Llama) with Each VrDU Task	97.5	97.9
+ Decoder (Llama) with Multi VrDU Tasks	98.1	98.1
+ Decoder (Llama) with Multi VrDU & NLP Tasks	**98.8**	**98.6**

Table 4: Impact of encoder.

Encoder	Modal	RVL-CDIP	CORD
LayoutLMv3	V+T+L	**98.8**	**98.6**
UniDoc	V+T+L	98.1	97.9
DiT	V	94.3	89.6

on FUNSD and CORD, respectively. These scores represent a significant improvement of 3.2 points over LayoutLMv3's best accuracy for the FUNSD dataset and 1.0 points over UDOP's best accuracy for the CORD dataset. UDOP and our method are both a single model for VrDU tasks. However, the proposed method uses a language model, extracting more accurate information from documents.

Document Visual Question Answering. Our method achieved an accuracy of 86.9%, which is comparable to the highest accuracy achieved in previous works. The 3.5 points improved over the baseline show its effectiveness, as it uses a language model as a decoder, which enhances language comprehension in Q&A sessions.

4.4. Detailed Analysis

We evaluated the method in detail using CORD and RVL-CDIP datasets.

Component Analysis. We performed a stepwise validation to see whether the proposed method works. Table 3 shows the results of fine-tuning to each task using only the encoder, fine-tuning to each VrDU task only using the decoder, fine-tuning to multiple VrDU tasks simultaneously, and finally fine-tuning with the NLP task. We confirm that incorporating the language model, VrDU multi-task training, and NLP multi-task training are all crucial.

Effects of Encoder Component. We evaluated the impact of the encoder component. Our approach is a flexible framework, and the encoder can be replaced. Table 4 shows the results of replacing the encoder with other methods. The encoder's feature output length is set to 512. It shows the successful performance of various methods and modalities. This suggests that more robust methods in the future can be incorporated flexibly.

Table 5: Affect on the NLP task performance.

Model	Average	ARC	HellaSwag	MMLU	TruthfulQA
Alpaca	52.02	52.05	77.00	41.45	**37.60**
Proposed model	**53.06**	**52.12**	**79.32**	**44.31**	36.49

Impact on the NLP Tasks. We used a generic LLM as a decoder and fine-tuned it with a regular NLP task and the VrDU task. We investigated how learning the NLP and VrDU tasks together affects the NLP task. We used various benchmarks that have been used in recent years to evaluate LLMs. We used the following benchmarks: ARC (Chollet, 2019) for multiple-choice, HellaSwag (Zellers et al., 2019) for sentence completion, MMLU (Hendrycks et al., 2020) for multidomain knowledge understanding, and Truthful TruthfulQA (Lin et al., 2021), which measures the accuracy of answers to questions. The results of the Alpaca model and the proposed method are presented in Table 5. Although the VrDU task appeared to negatively impact the NLP task because it is in a different domain, the average score increased. In particular, it improved by 2.86 points in language comprehension with MMLU. Further research is needed to explore which NLP tasks, such as summarization, are associated with higher scores with VrDU.

5. Conclusion

This study has presented a document analysis framework capable of performing multiple tasks. The proposed approach, LayoutLLM, combines a VrDU encoder to capture document images and a decoder, LLM, to receive task instructions and process them accordingly. It allows us to efficiently understand document images by capturing visual and textual context. Experimental results show that our method significantly improves the performance of various VrDU tasks. Furthermore, unlike previous studies, it can exploit LLMs' pure NLP task processing capability, not only for VrDU tasks.

6. Bibliographical References

Jaided AI. 2020. Easyocr. https://github.com/JaidedAI/EasyOCR.

Jordi Armengol-Estapé, Ona de Gibert Bonet, and Maite Melero. 2022. On the multilingual capabilities of very large-scale English language models. In *LREC*, pages 3056–3068.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Masato Fujitake. 2023a. A3s: Adversarial learning of semantic representations for scene-text spotting. In *ICASSP*, pages 1–5.

Masato Fujitake. 2023b. Diffusionstr: Diffusion model for scene text recognition. In *ICIP*, pages 1585–1589.

Masato Fujitake. 2024. Dtrocr: Decoder-only transformer for optical character recognition. In *WACV*, pages 8025–8035.

Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. 2021. Lambert: layout-aware language modeling for information extraction. In *ICDAR*, pages 532–547.

Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. 2021. Unidoc: Unified pretraining framework for document understanding. *NeurIPS*, 34:39–50.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *ICLR*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *ACM MM*, pages 4083–4091.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *ECCV*, pages 498–517.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, pages 18893–18912.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. In *ACL*, pages 6309–6318.

Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022. Dit: Self-supervised pre-training for document image transformer. In *ACM MM*, page 3530–3539.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *ICLR*, pages 1–10.

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *ICDAR*, pages 732–747.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *CVPR*, pages 19254–19264.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*, pages 2579–2591.

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 *ACM KDD*, pages 1192–1200.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. 2023. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

7. Language Resource References

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *ICDAR*, pages 991–995.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *ICDARW*, volume 2, pages 1–6.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *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. In *NeurIPSW*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.