Layout and Task Aware Instruction Prompt for Zero-shot Document Image Question Answering

Wenjin Wang, Yunhao Li, Yixin Ou, Yin Zhang*

Zhejiang University, Hangzhou, China
{wangwenjin,12121078,ouyixin,zhangyin98}@zju.edu.cn

Abstract

Layout-aware pre-trained models has achieved significant progress on document image question answering. They introduce extra learnable modules into existing language models to capture layout information within document images from text bounding box coordinates obtained by OCR tools. However, extra modules necessitate pre-training on extensive document images. This prevents these methods from directly utilizing off-the-shelf instruction-tuning language foundation models, which have recently shown promising potential in zero-shot learning. Instead, in this paper, we find that *instruction-tuning language models like Claude and ChatGPT can understand layout by spaces and line breaks*. Based on this observation, we propose the **LA**yout and **T**ask aware **In**struction **Prompt** (**LATIN-Prompt**), which consists of layout-aware document content and task-aware instruction. Specifically, the former uses appropriate spaces and line breaks to recover the layout information among text segments obtained by OCR tools, and the latter ensures that generated answers adhere to formatting requirements. Moreover, we propose the **LA**yout and **T**ask aware **In**struction **Tuning** (**LATIN-Tuning**) to improve the performance of small instruction-tuning models like Alpaca. Experimental results show that LATIN-Prompt enables zero-shot performance of Claude and ChatGPT to be comparable to the fine-tuning performance of SOTAs on document image question answering, and LATIN-Tuning enhances the zero-shot performance of Alpaca significantly. For example, LATIN-Prompt improves the performance of Claude and ChatGPT on DocVQA by 263% and 20% respectively. LATIN-Tuning improves the performance of Alpaca on DocVQA by 87.7%. Quantitative and qualitative analyses demonstrate the effectiveness of LATIN-Prompt and LATIN-Tuning. We release our code to facilitate future research[1].

1 Introduction

Intelligent document image question answering, as an important application of document intelligence, aims to develop AI systems to automatically answer natural language questions based on the understanding of document images. Compared with text documents, document images contain textual, visual, and layout information, which pose unique challenges for machine comprehension.

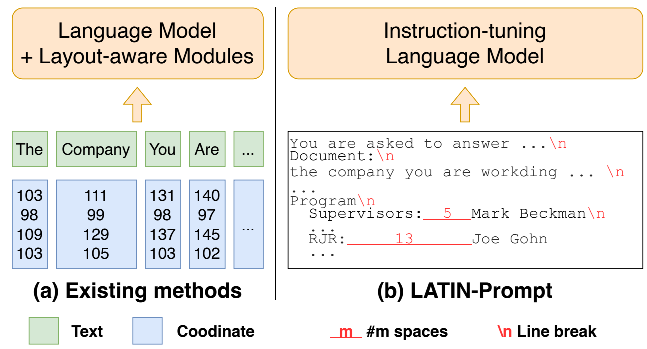


Figure 1: (a) Existing methods introduce layout-aware modules into language models to capture layout information within document images from text bounding box coordinates obtained by OCR tools. They need further pre-training on extensive document images. (b) Our method allows instruction-tuning language models to *capture layout by spaces and line breaks* and can conduct *zero-shot inference* on document image question-answering.

Recently, layout-aware pre-trained models has achieved significant progress on document image question answering. They introduce extra learnable modules on top of language models (Devlin et al. 2019; Liu et al. 2019b; Raffel et al. 2020; Bao et al. 2020) to capture layout information within document images from coordinates of text bounding box obtained by OCR tools (Fig. 1(a)). LayoutLM (Xu et al. 2020) introduces coordinate information into model input by 2D position embeddings. LayoutLMv2 (Xu et al. 2021b), LayoutLMv3 (Huang et al. 2022), and ERNIE-Layout (Peng et al. 2022) capture the layout information from coordinate by layout-aware attention mechanism. These methods conduct pre-training on extensive document images for the newly introduced layout-aware modules.

However, the need of pre-training prevents these methods from directly utilizing off-the-shelf instruction-tuning language foundation models, which have recently shown promising potential in zero-shot learning. On the one hand, commercial large instruction-tuning models like Claude (Anthropic 2023) and ChatGPT (OpenAI 2022) are closed-source, impeding further pre-training. On the other hand, open-source instruction-tuning models are of

*Corresponding author: Yin Zhang.

[1]https://github.com/WenjinW/LATIN-Prompt

much larger scale than traditional models. For example, Alpaca (Rohan Taori et al. 2023) consists of 7 billion parameters, whereas $\text{BERT}_{\text{large}}$ (Devlin et al. 2019) only comprises 300+ million parameters. Existing methods (Xu et al. 2020, 2021b; Huang et al. 2022; Peng et al. 2022) select over 10 million pages from the IIT-CDIP Test Collection dataset (Lewis et al. 2006) for pre-training. But the cost of pre-training instruction-tuning models like Alpaca on 10 million pages is too expensive.

Instead, in this work, we find that *instruction-tuning language models like Claude and ChatGPT can understand layout by spaces and line breaks* (Fig. 1(b)). Based on this observation, we propose the **LA**yout and **T**ask aware **In**struction **Prompt** (**LATIN-Prompt**), which consists of layout-aware document content and task-aware instruction. Specifically, given the OCR results, we use appropriate spaces and line breaks to connect all the text segments together, resulting in the layout-aware document content. The layout information contained within the coordinates is translated into spaces and line breaks. Further, we integrate task instruction into layout-aware document content, ensuring that the model generates answers that adhere to the formatting requirements. Although simple, our method is intuitive and consistent with human behavior. Humans employ whitespace (blank) regions to represent and comprehend layout.

We also find that small instruction-tuning language foundation models like Alpaca are not good at understanding layout by spaces. So we propose the **LA**yout and **T**ask aware **In**struction **Tuning** (**LATIN-Tuning**) to improve the performance of them. We convert CSV-format tables into strings containing spaces and line breaks and construct instruction-tuning data from these strings by Claude.

Our contributions are summarized as follows:

- We find that instruction-tuning models like Claude and ChatGPT can capture layout by spaces and line breaks, and propose LATIN-Prompt to conduct zero-shot inference on document image question-answering tasks.
- We propose the LATIN-Tuning to enhance the ability of Alpaca to comprehend layout by spaces and line breaks.
- Experiment results on three datasets show that LATIN-Prompt enables zero-shot performance of Claude and ChatGPT to be comparable to the fine-tuning performance of SOTAs on document image question answering, and LATIN-Tuning enhances the zero-shot performance of Alpaca significantly. Quantitative and qualitative analyses demonstrate the effectiveness of LATIN-Prompt and LATIN-Tuning.

2 Related Work

2.1 Visually-rich document understanding

Visually-rich Document Understanding (VrDU) focus on recognizing and understanding scanned or digital-born document images with language, vision, and layout information. Traditional works in the VrDU employ CNN (Yang et al. 2017; Katti et al. 2018; Denk and Reisswig 2019; Zhao et al. 2019; Sarkhel and Nandi 2019; Zhang et al. 2020; Wang et al. 2021; Lin et al. 2021), GNN (Liu et al. 2019a; Qian et al. 2019; Yu et al. 2020; Wei, He, and Zhang 2020; Carbonell et al. 2021), and language transformer (Majumder et al. 2020; Wang et al. 2020) to mine information from document images.

Recently, layout-aware pre-trained Transformers have been proposed (Appalaraju et al. 2021; Garncarek et al. 2021; Hwang et al. 2021; Li et al. 2021a,b,c; Xu et al. 2021a; Hong et al. 2022; Lee et al. 2022a; Peng et al. 2022; Bai et al. 2022a; Lee et al. 2022b; Luo et al. 2023; Dhouib, Bettaieb, and Shabou 2023). LayoutLM (Xu et al. 2020) introduces the 2D position information into the input embedding and LayoutLMv2 (Xu et al. 2021b) proposes the spatial-aware self-attention mechanism. Then, LayoutLMv3 (Huang et al. 2022) removes the CNN by learning visual features extraction with the discrete image tokens reconstruction, and ERNIE-Layout (Peng et al. 2022) introduces the layout knowledge into pre-training. Further, (Zhang et al. 2021; Gu et al. 2022; Wang et al. 2022a) introduce additional designs during the fine-tuning, enabling layout-aware pre-trained models to better adapt to downstream tasks.

However, all of existing methods try to understand layout within document images by coordinates of bounding box obtained by OCR tools. Instead, in this work, we try to directly understand layout by spaces and line breaks. We propose the LATIN-Prompt and LATIN-Tuning to explore zero-shot document image question answering.

A concurrent work ICL-D3IE (He et al. 2023) also leverages large instruction-tuning models in document image understanding, but it differs significantly from our method. It focuses on few-shot document information extraction, but in this paper, we focus on zero-shot document image question-answering.

2.2 Instruction-tuning Language Model

An ideal AI system should be able to learn and accomplish a variety of tasks according to human instructions. To this end, many instruction-tuning datasets and language foundation models have been proposed (Thoppilan et al. 2022; Chung et al. 2022; Wei et al. 2022; Sanh et al. 2022; Mishra et al. 2022; Wang et al. 2022c; Xu, Shen, and Huang 2022; Iyer et al. 2023; Longpre et al. 2023). To fully align models with human intentions and values, reinforcement learning from human feedback and AI feedback (Bai et al. 2022b; Ouyang et al. 2022; Bai et al. 2022c; Open AI 2023) are introduced into instruction-tuning. To reduce the cost of manual annotation of instruction-tuning data, some methods automatically construct instruction-tuning data using off-the-shelf language models (Honovich et al. 2022; Wang et al. 2022b; Rohan Taori et al. 2023; Peng et al. 2023; Zhou et al. 2023; Xu et al. 2023). Recently, some methods (Zhu et al. 2023; Dai et al. 2023) have extended instruction fine-tuning into multimodal domains.

In this paper, we propose the LATIN-Prompt to allow off-the-shelf instruction-tuning language models to conduct zero-shot document image question answering. Moreover, we propose the LATIN-Tuning to enhance the ability of Alpaca to comprehend layout by spaces and line breaks.

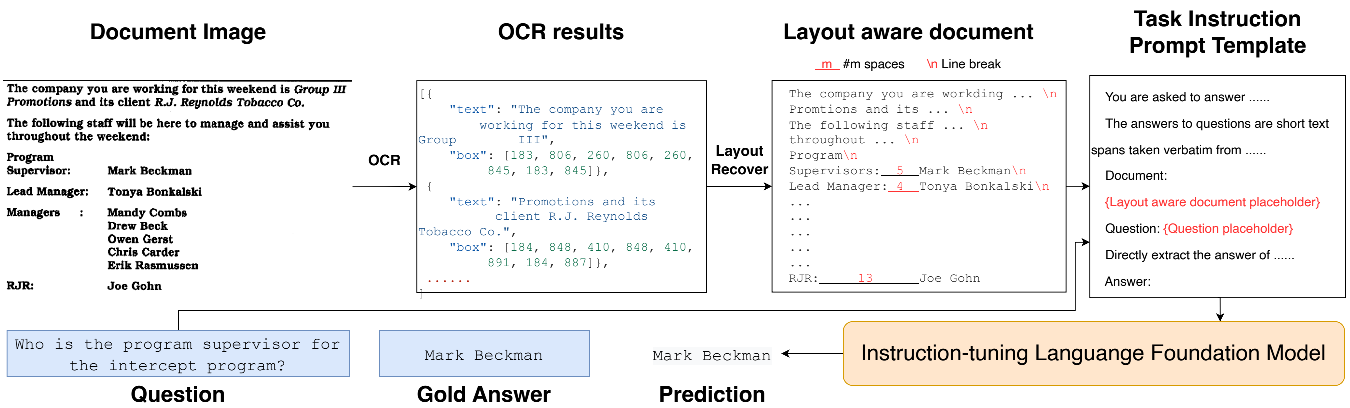


Figure 2: The overview of LATIN-Prompt (Sec. 3.1). Given a document image and the corresponding question, we recover the layout information within the document image from OCR results using appropriate spaces and line breaks, and then insert the layout aware document content and question into the task instruction prompt template together. The instruction-tuning large language foundation model takes the filled template as input and predicts the answer to the question in the required format.

#Line	Prompt
1	You are asked to answer questions asked on a document image.
2	The answers to questions are short text spans taken verbatim from the document. This means that the answers comprise a set of contiguous text tokens present in the document.
3	Document:
4	{Layout Aware Document placeholder}
5	
6	Question: {Question placeholder}
7	
8	Directly extract the answer of the question from the document with as few words as possible.
9	
10	Answer:

Table 1: DocVQA Prompt Template. The {} represents the placeholder.

3 Method

We propose the LATIN-Prompt and LATIN-Tuning for instruction-tuning language models to conduct zero-shot inference on document image question-answering tasks.

3.1 LATIN-Prompt

The key ideas of LATIN-Prompt are as follows: (1) Capture the layout information by spaces and line breaks (2) Generate answers adhere to formatting requirements by task instruction. Fig. 2 illustrates the process of LATIN-Prompt. Given OCR results of a document image, we recover layout information within it by using appropriate spaces and line breaks to connect all the text segments together, resulting in the layout-aware document content. Then we insert the layout-aware document content and question into the task instruction prompt template. The instruction-tuning language model takes the filled template as input and predicts the answer to the question in the required format.

Formally, given a document image D and a question answer pair q and a, we process the document image by an OCR tool. The extracted text segments and corresponding bounding boxes are denoted as $S = \{s_1, s_2, \ldots, s_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, where n represents the number of text segments.

Layout Aware Document We employ appropriate spaces and line breaks to connect all text segments together, resulting in layout-aware document content. The process is as follows:

Step 1. Re-arrange the text segments and bounding boxes in the order from top to bottom and from left to right based on the coordinates.

Step 2. According to the coordinates, place the text segments and bounding boxes in the i-th row into the list S_i and B_i respectively from left to right, and calculate the character count c_i and the width w_i of i-th row. The w_i equals the width of the union of bounding boxes in the list B_i.

Step 3. Calculate the character width of document D, which is defined as follows:

$$\bar{c} := w_i^* / c_i^*, i^* = \text{argmax}_i c_i, \tag{1}$$

where i^*-row has the maximum character count among all rows.

Step 4. Join text segments in the same row from left to right by spaces. Given two adjacent text segments $S_{i,j}$ and $S_{i,k}$, the number of spaces joining them is equal to

$h_{i,jk}/\bar{c}$ where $h_{i,jk}$ is the horizontal distance between the two bounding boxes $B_{i,j}$ and $B_{i,k}$.

Step 5. Join different rows by line breaks to obtain the layout-aware document content (denoted as S').

Recovering layout information by spaces and line breaks is simple but intuitive. In fact, people do represent and understand layout through blank areas between text elements, rather than precise bounding box coordinates.

Task Aware Instruction Different from the open-ended question-answering, document image question-answering typically involves explicit requirements for the answer format. For example, DocVQA (Mathew, Karatzas, and Jawahar 2021) is an extractive QA task that requires answers to be extracted from the document. However, with only the layout-aware document content and question, the model can easily generate answers that are not in the document and generate unnecessary descriptions or explanations.

So we integrate task instruction into layout-aware document content, ensuring that the model generates answers that adhere to the formatting requirements. Specifically, we manually designed different instruction prompt templates for different tasks. Each template $P(S', q)$ contains the requirement of the task as well as placeholders for the layout-aware document content S' and question q.

Table 7 shows the prompt template for DocVQA. In the first and second lines, we explain the meaning of extraction in detail to the model according to the task description in DocVQA. Lines 3 to 6 provide placeholders for the layout-aware document and question. To avoid the model forgetting its task due to the interference of document content, the 8th line summarizes and reiterates the task requirements. Please refer to the supplementary materials for the prompt templates of InfographicVQA (Mathew et al. 2021) and MP-DocVQA (Tito, Karatzas, and Valveny 2023).

Zero-shot Inference At last, the instruction-tuning language model M takes the filled template $P(S', q)$ as input and predicts the answer as follows:

$$a' = f_M(P(S', q)), \tag{2}$$

where a' represents the prediction and the f_M represents the decoding process of model M.

3.2 LATIN-Tuning

Although instruction-tuning models like Claude and ChatGPT can comprehend and utilize LATIN-Prompt well, we found that the performance of smaller models like Alpaca (7B) was not up to par. So we propose LATIN-Tuning to enhance their ability to comprehend layout by spaces and line breaks. As shown in Fig. 3, we employ the Pandas[2] and Claude to construct instruction-tuning dataset from CSV-format tables. The process is as follows:

(1) For each CSV table, we convert it into document string with spaces and line breaks by Pandas. Please refer to the appendix for the code implementation. (2) We insert the document string into the Question Generation Prompt Template and generate a question-answer pair by Claude. (3) We insert

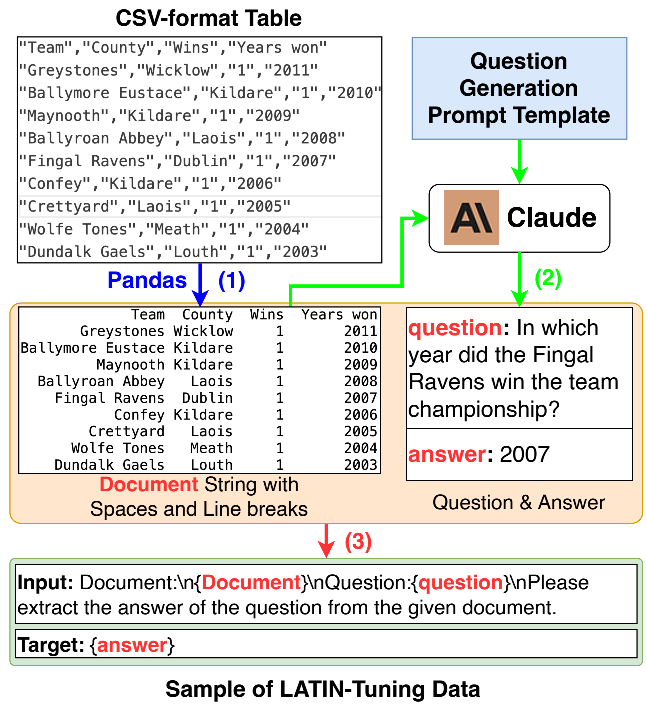


Figure 3: Construction of LATIN-Tuning data (Sec. 3.2). (1) Convert the CSV-format table into document string with spaces and line breaks by Pandas. (2) Insert document string into the Question Generation Prompt Template and generate a question-answer pair by Claude. (3) Insert document string and question into the Instruction Prompt Template to form the input, with the answer serving as the target.

document string and question into the Instruction Prompt Template to form the input, with the answer serving as the target. Refer to the appendix for the details of Question Generation Prompt Template and Instruction Prompt Template.

At last, we fine-tune the Alpaca on the instruction-tuning dataset to enhance its ability to comprehend layout by spaces and line breaks.

4 Experiment

4.1 Experiment Settings

Datasets We evaluate our method on three document image question answering datasets: DocVQA (Mathew, Karatzas, and Jawahar 2021) is an extractive question answering task and consists of 50,000 questions defined on 12,767 document images; InfographicVQA (Mathew et al. 2021) consists of 5,485 infographics, which convey information through text, graphics, and visual elements together; MP-DocVQA (Tito, Karatzas, and Valveny 2023) extends DocVQA to more realistic multi-page scenarios where a document typically consists of multiple pages that should be processed together. Following common practice, we use Azure OCR results for DocVQA provided by DUE (Borchmann et al. 2021) and use offical OCR results[3]

[2]https://pandas.pydata.org

[3]https://rrc.cvc.uab.es/?ch=17&com=introduction

Paradigm	Method	Parameters	Text	Vision	Layout	Fine-tuning Set	ANLS	ΔANLS
Fine-tuning	$\text{BERT}_{\text{LARGE}}$	340M	✓			train	0.6768	
	$\text{RoBERTa}_{\text{LARGE}}$	355M	✓			train	0.6952	
	$\text{UniLMv2}_{\text{LARGE}}$	340M	✓			train	0.7709	
	$\text{LayoutLM}_{\text{LARGE}}$	343M	✓		✓	train	0.7259	
	$\text{LayoutLMv2}_{\text{LARGE}}$	426M	✓	✓	✓	train	0.8348	
	$\text{LayoutLMv3}_{\text{LARGE}}$	368M	✓	✓	✓	train	0.8337	
	$\text{ERNIE-Layout}_{\text{LARGE}}$	507M	✓	✓	✓	train	0.8321	
	$\text{LayoutLMv2}_{\text{LARGE}}$	426M	✓	✓	✓	train + dev	0.8529	
	$\text{ERNIE-Layout}_{\text{LARGE}}$	507M	✓	✓	✓	train + dev	0.8486	
Zero-shot	Alpaca+Plain Prompt	7B	✓			-	0.3567	
	Alpaca+LATIN-Prompt		✓		✓	-	0.4200	+0.0633
	Claude+Plain Prompt	Unknown	✓			-	0.2298	
	Claude+LATIN-Prompt		✓		✓	-	0.8336	+0.6038
	ChatGPT-3.5+Plain Prompt	Unknown	✓		✓	-	0.6866	
	ChatGPT-3.5+LATIN-Prompt		✓		✓	-	0.8255	+0.1389
	GPT-4*	Unknown	not clearly described			-	0.8840	

* represents that we report the result of GPT-4 presented in OpenAI blog (OpenAI 2023). Although lacking a technical detail description, compared with Claude and ChatGPT-3.5, GPT-4 utilizes visual information. The LATIN-Prompt is orthogonal to GPT-4 and can be used to further improve the performance of GPT-4. However, due to API permission limitations, we are unable to evaluate the performance of GPT-4 + LATIN-Prompt and leave it in future.

Table 2: Performance on test dataset of DocVQA. Text, Vision, and Layout represent the modal information used by the model. The ΔANLS represents the gain of LATIN-Prompt compared to Plain Prompt. Unknown indicates missing relevant details.

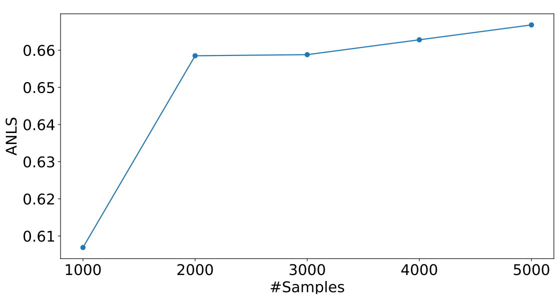


Figure 4: The impact of the size of instruction fine-tuning dataset on LATIN-Tuning. The performance of LATIN-Tuning improves as the number of samples increases.

for InfographicVQA and MP-DocVQA. For all datasets, we adopt the Average Normalized Levenshtein Similarity (ANLS) (Biten et al. 2019) as the evaluation metric.

LATIN-Prompt Baselines To evaluate zero-shot performance of LATIN-Prompt, we compare it with Plain Prompt on three instruction-tuning language models: Claude[4], ChatGPT-3.5[5], and Alpaca. The template of Plain Prompt is as follows: "Document: {document}\nQuestion: {question}\nDirectly extract the answer of the question from the document.\nAnswer:", where {document} and {question} are placeholders of original text segments from OCR tools and question. We also compare LATIN-Prompt's zero-shot performance with fine-tuning performance of pre-training-fine-tuning methods. Moreover, we report the result of multimodal GPT-4 presented in OpenAI blog (OpenAI 2023). Due to API permission restrictions, we leave the exploration of GPT-4+LATIN-Prompt as future work.

We only evaluate Alpaca on DocVQA because the other two tasks are too complex for it. Alpaca cannot follow the task instruction of these two tasks. In fact, Alpaca performs poorly on InfographicVQA (refer to Tab. 5) and cannot generate answers meeting the format requirement of MP-DocVQA. We exclude the ChatGPT-3.5 on MP-DocVQA because it needs to process too many document pages and the experimental cost exceeds the range we can afford.

LATIN-Tuning We randomly sample 5000 CSV-format tables from the WikiTableQuestions (Pasupat and Liang 2015) with replacement to create the instruction-tuning dataset. We fine-tune the Alpaca on the created dataset for 3 epochs using the AdamW (Loshchilov and Hutter 2018) optimizer with a warmup ratio of 0.03 following the Alpaca (Rohan Taori et al. 2023). We use a batch size of 64 and a learning rate of $2e-5$. The resulting model is denoted as Alpaca+LATIN-Tuning and we compare it with Alpaca to evaluate the performance of LATIN-Tuning.

4.2 Performance of LATIN-Prompt

Table 2 presents the experimental results on DocVQA. (1) In the pre-training-fine-tuning paradigm, the layout-aware multimodal pre-trained model performs better than the pure language model. (2) Further, increasing the amount of fine-tuning data can improve the performance of models. (3) The instruction-tuning language models perform poorly with the

[4]We use the claude-v1.3 API.

[5]We use the gpt-3.5-turbo API from Azure OpenAI.

Paradigm	Method	Overall		Answer type			
		ANLS	ΔANLS	Image span	Question span	Multiple spans	Non span
Fine-tuning	BERT	0.2078		0.2625	0.2333	0.0739	0.0259
	LayoutLM	0.2720		0.3278	0.2386	0.0450	0.1371
	LayoutLMv2	0.2829		0.3430	0.2763	0.0641	0.1114
	BROS	0.3219		0.3997	0.2317	0.1064	0.1068
	pix2struct	0.4001		0.4308	0.4839	0.2059	0.3173
	TILT	**0.6120**		**0.6765**	**0.6419**	**0.4391**	**0.3832**
Zero-shot	Claude + Plain Prompt	0.0798		0.0951	0.0913	0.0203	0.0280
	Claude + LATIN-Prompt	0.5451	+0.4653	0.5992	0.5861	0.3985	0.3544
	ChatGPT-3.5 + Plain Prompt	0.3335		0.3749	0.4505	0.0950	0.1822
	ChatGPT-3.5 + LATIN-Prompt	0.4898	+0.1563	0.5457	0.5639	0.3458	0.2798

Method	Evidence					Operation		
	Table/List	Textual	Visual object	Figure	Map	Comparison	Arithmetic	Counting
BERT	0.1852	0.2995	0.0896	0.1942	0.1709	0.1805	0.0160	0.0436
LayoutLM	0.2400	0.3626	0.1705	0.2551	0.2205	0.1836	0.1559	0.1140
LayoutLMv2	0.2449	0.3855	0.1440	0.2601	0.3110	0.1897	0.1130	0.1158
BROS	0.2653	0.4488	0.1878	0.3095	0.3231	0.2020	0.1480	0.0695
pix2struct	0.3833	0.5256	0.2572	0.3726	0.3283	0.2762	0.4198	0.2017
TILT	**0.5917**	**0.7916**	0.4545	**0.5654**	0.4480	**0.4801**	**0.4958**	0.2652
Claude + Plain Prompt	0.0849	0.1099	0.0858	0.0695	0.0496	0.0589	0.0271	0.0368
Claude + LATIN-Prompt	0.5421	0.6725	**0.4897**	0.5027	**0.4982**	0.4598	0.4311	**0.2708**
ChatGPT-3.5 + Plain Prompt	0.3481	0.3893	0.3670	0.3114	0.1843	0.2349	0.1466	0.2320
ChatGPT-3.5 + LATIN-Prompt	0.4917	0.6016	0.4491	0.4585	0.3614	0.4312	0.3157	0.2660

Table 3: Performance on test dataset of InfographicVQA. The questions in InfographicVQA can be grouped according to answer type, evidence source, and operation. We list both the overall performance of the model and its performance on different groups. All performances are evaluated by ANLS. The ΔANLS represents the gain of LATIN-Prompt compared to Plain Prompt. The highest and second-highest scores are bolded and underlined.

plain prompt based on the original text segments obtained by OCR tools. (4) The LATIN-Prompt proposed in this paper significantly improves the zero-shot performance of instruction-tuning language models. It enables the zero-shot performance of Claude and ChatGPT-3.5 to significantly outperform the fine-tuned layout-aware LayoutLM. In addition, despite only using text and layout information, their zero-shot performance is comparable to the performance of fine-tuned layout-aware multimodal pre-trained models. (5) Although unknown, the number of parameters of Claude and GPT-3.5 should be much larger than that of Alpaca. The experimental results show that the final zero-shot performance is positively correlated with the size and ability of the instruction-tuning models. (6) The zero-shot performance of GPT-4 matched the best fine-tuned performance. Although lacking a technical detail description, compared with Claude and GPT-3.5, GPT-4 utilizes visual information, reflecting the importance of visual information for document image understanding. LATIN-Prompt is orthogonal to GPT-4. However, due to API permission restrictions, we can only leave arming GPT-4 with the LATIN-Prompt to the future.

Table 3 presents results on InfographicVQA. Experimental results show that LATIN-Prompt enable the zero-shot performance of Claude and GPT-3.5 to exceed the performance of all fine-tuned baselines except TILT. We find that Claude performs poorly with Plain Prompt, but its performance improves significantly when using LATIN-Prompt.

Table 4 presents results on MP-DocVQA. It shows that, with LATIN-Prompt, Claude's zero-shot performance exceeds fine-tuning performance of Longformer (Beltagy, Peters, and Cohan 2020) and Big Bird (Zaheer et al. 2021) designed for long sequences. Furthermore, its zero-shot performance is comparable to the fine-tuning performance of Hi-VT5 (Tito, Karatzas, and Valveny 2023), a layout-aware multimodal model for multi-page document images.

4.3 Performance of LATIN-Tuning

Table 5 demonstrates that LATIN-Tuning improves the performance of Alpaca on DocVQA by 87.7% and on InfographicVQA by 113%. Nevertheless, its performance still lags behind Claude. We will explore more effective instruction-tuning method in the future.

4.4 Quantitative and Qualitative Analyses

Effect of components of LATIN-Prompt LATIN-Prompt consists of layout-aware document content (Layout) and task instruction (Task). Table 6 presents the results of ablation study of LATIN-Prompt with Claude and ChatGPT-3.5 on DocVQA and InfographicVQA. The results show that both the layout-aware document content and the task instruction can significantly improve the zero-shot performance of Claude and ChatGPT-3.5. The improvement brought by task

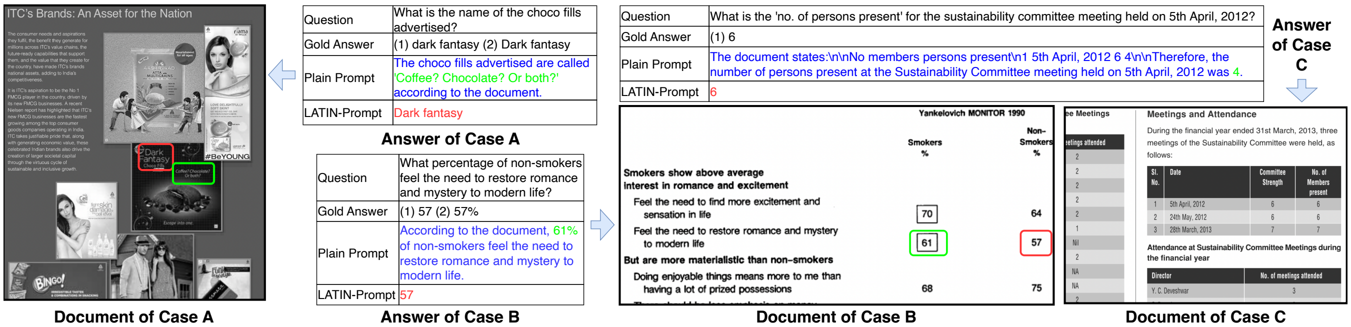


Figure 5: Case study of Claude on DocVQA. Due to the lack of task instruction, Plain Prompt generates unnecessary words (in blue), violating the extraction requirement. Moreover, Plain Prompt cannot capture layout information and generates incorrect answers (in green). In Case (A), it regards "Coffee" and "Chocolate", which are semantically similar to "choco", as the answer. In Case (B), it regards "61" as the answer since it directly follows "romance and mystery". In Case (C), it fails to comprehend the document and engages in erroneous reasoning. In contrast, LATIN-Prompt can understand layout relationships and generate the correct answer (in red). The document images of these cases are complex. Due to limited space, we only display a portion of the original document images here. Please refer to the appendix for the original document images and more cases.

Paradigm	Method	Setup	ANLS
Fine-tuning	BERT	Max Conf	0.5347
		Concat	0.4183
	Longformer	Max Conf	0.5506
		Concat	0.5287
	Big Bird	Max Conf	0.5854
		Concat	0.4929
	LayoutLMv3	Max Conf	0.5513
		Concat	0.4538
	T5	Max Conf	0.4028
		Concat	0.5050
	Hi-VT5	Multipage	**0.6201**
Zero-shot	Claude+LATIN-Prompt	Max Conf	<u>0.6129</u>

Table 4: Performance on test dataset of MP-DocVQA. Concat indicates concatenating multi-page content. MaxConf indicates processing pages separately and choosing an answer based on confidence. We only evaluate Claude with LATIN-Prompt in MaxConf setting because Plain Prompt does not apply to MaxConf, and Concat exceeds the length limit.

instruction is more significant in Claude because it ensures that the format of the answers generated by the model meets the task requirements. On the basis of the correct format, the layout-aware document content further improves the performance of the model because it enables the model to utilize the layout information among text segments.

Effect of instruction-tuning data size for LATIN-Tuning Figure 4 shows that the performance of LATIN-Tuning improves as the number of samples increases. The improvement rate slows down when the sample count exceeds 2000.

Case study of LATIN-Prompt Figure 5 provides cases of Claude on DocVQA. Compared with Plain Prompt, LATIN-Prompt enables the model to comprehend layout more effectively and generate answers meeting the format requirement.

Method	DocVQA		InfographicVQA	
	Valid	Test	Valid	Test
Alpaca	0.3506	0.3567	0.1083	0.1419
Alpaca+LATIN-Tuning	0.6668	0.6697	0.2873	0.3028
Claude	0.8311	0.8336	0.5218	0.5451

Table 5: Effect of LATIN-Tuning. All methods are equipped with LATIN-Prompt.

Prompt	DocVQA		InfographicVQA	
	Claude	ChatGPT	Claude	ChatGPT
LATIN-Prompt	0.8311	0.8135	0.5218	0.4708
w/o Layout	0.7825	0.7491	0.4638	0.4341
w/o Task	0.3637	0.7561	0.1234	0.4296
w/o Task+Layout	0.2144	0.6795	0.0702	0.3103

Table 6: Ablation of LATIN-Prompt on validation data of DocVQA and InfographicVQA.

Case study of LATIN-Tuning Case study on DocVQA shows that LATIN-Tuning enables Alpaca to understand layout by spaces. Please refer to the appendix for details.

5 Conclusion

In this work, we point a new perspective for comprehending layout information within document images. Instead of capturing layout by coordinate of bounding boxes, we find that instruction-tuning language models like Claude and ChatGPT can understand layout by spaces and line breaks. Based on this observation, we propose LATIN-Prompt and it enables zero-shot performance of Claude and ChatGPT to be comparable to the fine-tuning performance of SOTAs on document image question answering. Moreover, we propose LATIN-Tuning, which enhances the ability of Alpaca to comprehend layout by spaces and line breaks. In the future, we will explore to incorporate visual information into LATIN-Prompt and create more effective instruction-tuning dataset for LATIN-Tuning.

References

Anthropic. 2023. Claude. https://www.anthropic.com/product.

Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Manmatha, R. 2021. DocFormer: End-to-End Transformer for Document Understanding. In *ICCV 2021*, 993–1003.

Bai, H.; Liu, Z.; Meng, X.; Li, W.; Liu, S.; Xie, N.; Zheng, R.; Wang, L.; Hou, L.; Wei, J.; Jiang, X.; and Liu, Q. 2022a. Wukong-Reader: Multi-modal Pre-training for Fine-grained Visual Document Understanding. arxiv:2212.09621.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; Joseph, N.; Kadavath, S.; Kernion, J.; Conerly, T.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Hume, T.; Johnston, S.; Kravec, S.; Lovitt, L.; Nanda, N.; Olsson, C.; Amodei, D.; Brown, T.; Clark, J.; McCandlish, S.; Olah, C.; Mann, B.; and Kaplan, J. 2022b. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arxiv:2204.05862.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; Chen, C.; Olsson, C.; Olah, C.; Hernandez, D.; Drain, D.; Ganguli, D.; Li, D.; Tran-Johnson, E.; Perez, E.; Kerr, J.; Mueller, J.; Ladish, J.; Landau, J.; Ndousse, K.; Lukosuite, K.; Lovitt, L.; Sellitto, M.; Elhage, N.; Schiefer, N.; Mercado, N.; DasSarma, N.; Lasenby, R.; Larson, R.; Ringer, S.; Johnston, S.; Kravec, S.; Showk, S. E.; Fort, S.; Lanham, T.; Telleen-Lawton, T.; Conerly, T.; Henighan, T.; Hume, T.; Bowman, S. R.; Hatfield-Dodds, Z.; Mann, B.; Amodei, D.; Joseph, N.; McCandlish, S.; Brown, T.; and Kaplan, J. 2022c. Constitutional AI: Harmlessness from AI Feedback. arxiv:2212.08073.

Bao, H.; Dong, L.; Wei, F.; Wang, W.; Yang, N.; Liu, X.; Wang, Y.; Gao, J.; Piao, S.; Zhou, M.; and Hon, H.-W. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, 642–652. PMLR.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. arxiv:2004.05150.

Biten, A. F.; Tito, R.; Mafla, A.; Gomez, L.; Rusiñol, M.; Valveny, E.; Jawahar, C. V.; and Karatzas, D. 2019. Scene Text Visual Question Answering. In *ICCV 2019*. arXiv.

Borchmann, Ł.; Pietruszka, M.; Stanislawek, T.; Jurkiewicz, D.; Turski, M.; Szyndler, K.; and Graliński, F. 2021. DUE: End-to-End Document Understanding Benchmark. In *NeurIPS 2021*.

Carbonell, M.; Riba, P.; Villegas, M.; Fornes, A.; and Llados, J. 2021. Named Entity Recognition and Relation Extraction with Graph Neural Networks in Semi Structured Documents. In *ICPR 2020*, 9622–9627. Milan, Italy: IEEE. ISBN 978-1-72818-808-9.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; Webson, A.; Gu, S. S.; Dai, Z.; Suzgun, M.; Chen, X.; Chowdhery, A.; Castro-Ros, A.; Pellat, M.; Robinson, K.; Valter, D.; Narang, S.; Mishra, G.; Yu, A.; Zhao, V.; Huang, Y.; Dai, A.; Yu, H.; Petrov, S.; Chi, E. H.; Dean, J.; Devlin, J.; Roberts, A.; Zhou, D.; Le, Q. V.; and Wei, J. 2022. Scaling Instruction-Finetuned Language Models. arxiv:2210.11416.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arxiv:2305.06500.

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

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, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dhouib, M.; Bettaieb, G.; and Shabou, A. 2023. DocParser: End-to-end OCR-free Information Extraction from Visually Rich Documents. In *ICDAR 2023*.

Garncarek, Ł.; Powalski, R.; Stanisławek, T.; Topolski, B.; Halama, P.; Turski, M.; and Graliński, F. 2021. LAMBERT: Layout-Aware Language Modeling for Information Extraction. In Lladós, J.; Lopresti, D.; and Uchida, S., eds., *ICDAR 2021*, Lecture Notes in Computer Science, 532–547. Cham: Springer International Publishing. ISBN 978-3-030-86549-8.

Gu, Z.; Meng, C.; Wang, K.; Lan, J.; Wang, W.; Gu, M.; and Zhang, L. 2022. XYLayoutLM: Towards Layout-Aware Multimodal Networks For Visually-Rich Document Understanding. In *CVPR 2022*, 10.

He, J.; Wang, L.; Hu, Y.; Liu, N.; Liu, H.; Xu, X.; and Shen, H. T. 2023. ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction. arxiv:2303.05063.

Hong, T.; Kim, D.; Ji, M.; Hwang, W.; Nam, D.; and Park, S. 2022. BROS: A Pre-Trained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents. In *AAAI 2022*.

Honovich, O.; Scialom, T.; Levy, O.; and Schick, T. 2022. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. arxiv:2212.09689.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *ACM MM 2022*.

Hwang, W.; Yim, J.; Park, S.; Yang, S.; and Seo, M. 2021. Spatial Dependency Parsing for Semi-Structured Document Information Extraction. In *ACL-Findings 2021*.

Iyer, S.; Lin, X. V.; Pasunuru, R.; Mihaylov, T.; Simig, D.; Yu, P.; Shuster, K.; Wang, T.; Liu, Q.; Koura, P. S.; Li, X.;

O'Horo, B.; Pereyra, G.; Wang, J.; Dewan, C.; Celikyilmaz, A.; Zettlemoyer, L.; and Stoyanov, V. 2023. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. arxiv:2212.12017.

Katti, A. R.; Reisswig, C.; Guder, C.; Brarda, S.; Bickel, S.; Höhne, J.; and Faddoul, J. B. 2018. Chargrid: Towards Understanding 2D Documents. In *EMNLP 2018*.

Lee, C.-Y.; Li, C.-L.; Dozat, T.; Perot, V.; Su, G.; Hua, N.; Ainslie, J.; Wang, R.; Fujii, Y.; and Pfister, T. 2022a. FormNet: Structural Encoding beyond Sequential Modeling in Form Document Information Extraction. In *ACL 2022*.

Lee, K.; Joshi, M.; Turc, I.; Hu, H.; Liu, F.; Eisenschlos, J.; Khandelwal, U.; Shaw, P.; Chang, M.-W.; and Toutanova, K. 2022b. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. arxiv:2210.03347.

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*, 665–666. Seattle Washington USA: ACM. ISBN 978-1-59593-369-0.

Li, C.; Bi, B.; Yan, M.; Wang, W.; Huang, S.; Huang, F.; and Si, L. 2021a. StructuralLM: Structural Pre-training for Form Understanding. In *ACL 2021*.

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 *CVPR 2021*, 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. In *ACM MM 2021*.

Lin, W.; Gao, Q.; Sun, L.; Zhong, Z.; Hu, K.; Ren, Q.; and Huo, Q. 2021. ViBERTgrid: A Jointly Trained Multi-modal 2D Document Representation for Key Information Extraction from Documents. In Lladós, J.; Lopresti, D.; and Uchida, S., eds., *ICDAR 2021*, Lecture Notes in Computer Science, 548–563. Cham: Springer International Publishing. ISBN 978-3-030-86549-8.

Liu, X.; Gao, F.; Zhang, Q.; and Zhao, H. 2019a. Graph Convolution for Multimodal Information Extraction from Visually Rich Documents. In *NAACL-HLT 2019*, 32–39. Minneapolis, Minnesota: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arxiv:1907.11692.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; and Roberts, A. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arxiv:2301.13688.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Luo, C.; Cheng, C.; Zheng, Q.; and Yao, C. 2023. GeoLayoutLM: Geometric Pre-training for Visual Information Extraction. In *CVPR 2023*.

Majumder, B. P.; Potti, N.; Tata, S.; Wendt, J. B.; Zhao, Q.; and Najork, M. 2020. Representation Learning for Information Extraction from Form-like Documents. In *ACL 2020*, 6495–6504. Online: Association for Computational Linguistics.

Mathew, M.; Bagal, V.; Tito, R. P.; Karatzas, D.; Valveny, E.; and Jawahar, C. V. 2021. InfographicVQA. In *WACV 2022*.

Mathew, M.; Karatzas, D.; and Jawahar, C. V. 2021. DocVQA: A Dataset for VQA on Document Images. In *WACV 2021*, 2199–2208. Waikoloa, HI, USA: IEEE. ISBN 978-1-66540-477-8.

Mishra, S.; Khashabi, D.; Baral, C.; and Hajishirzi, H. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3470–3487. Dublin, Ireland: Association for Computational Linguistics.

Open AI. 2023. GPT-4 Technical Report.

OpenAI. 2022. ChatGPT. https://openai.com/blog/chatgpt.

OpenAI. 2023. GPT-4. https://openai.com/research/gpt-4.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Gray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. In *NeurIPS 2022*.

Pasupat, P.; and Liang, P. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1470–1480. Beijing, China: Association for Computational Linguistics.

Peng, B.; Li, C.; He, P.; Galley, M.; and Gao, J. 2023. Instruction Tuning with GPT-4. arxiv:2304.03277.

Peng, Q.; Pan, Y.; Wang, W.; Luo, B.; Zhang, Z.; Huang, Z.; Cao, Y.; Yin, W.; Chen, Y.; Zhang, Y.; Feng, S.; Sun, Y.; Tian, H.; Wu, H.; and Wang, H. 2022. ERNIE-Layout: Layout Knowledge Enhanced Pre-training for Visually-rich Document Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3744–3756. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Qian, Y.; Santus, E.; Jin, Z.; Guo, J.; and Barzilay, R. 2019. GraphIE: A Graph-Based Framework for Information Extraction. In *NAACL-HLT 2019*, 751–761. Minneapolis, Minnesota: Association for Computational Linguistics.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Rohan Taori; Ishaan Gulrajani; Tianyi Zhang; Yann Dubois; Xuechen Li; Carlos Guestrin; and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA Model. *GitHub*.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; Dey, M.; Bari, M. S.; Xu, C.; Thakker, U.; Sharma, S. S.; Szczechla, E.; Kim, T.; Chhablani, G.; Nayak, N.; Datta, D.; Chang, J.; Jiang, M. T.-J.; Wang, H.; Manica, M.; Shen, S.; Yong, Z. X.; Pandey, H.; Bawden, R.; Wang, T.; Neeraj, T.; Rozen, J.; Sharma, A.; Santilli, A.; Fevry, T.; Fries, J. A.; Teehan, R.; Bers, T.; Biderman, S.; Gao, L.; Wolf, T.; and Rush, A. M. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR 2022*.

Sarkhel, R.; and Nandi, A. 2019. Deterministic Routing between Layout Abstractions for Multi-Scale Classification of Visually Rich Documents. In *IJCAI 2019*, 3360–3366. Macao, China: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1.

Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; Li, Y.; Lee, H.; Zheng, H. S.; Ghafouri, A.; Menegali, M.; Huang, Y.; Krikun, M.; Lepikhin, D.; Qin, J.; Chen, D.; Xu, Y.; Chen, Z.; Roberts, A.; Bosma, M.; Zhao, V.; Zhou, Y.; Chang, C.-C.; Krivokon, I.; Rusch, W.; Pickett, M.; Srinivasan, P.; Man, L.; Meier-Hellstern, K.; Morris, M. R.; Doshi, T.; Santos, R. D.; Duke, T.; Soraker, J.; Zevenbergen, B.; Prabhakaran, V.; Diaz, M.; Hutchinson, B.; Olson, K.; Molina, A.; Hoffman-John, E.; Lee, J.; Aroyo, L.; Rajakumar, R.; Butryna, A.; Lamm, M.; Kuzmina, V.; Fenton, J.; Cohen, A.; Bernstein, R.; Kurzweil, R.; Aguera-Arcas, B.; Cui, C.; Croak, M.; Chi, E.; and Le, Q. 2022. LaMDA: Language Models for Dialog Applications. arxiv:2201.08239.

Tito, R.; Karatzas, D.; and Valveny, E. 2023. Hierarchical Multimodal Transformers for Multi-Page DocVQA. arxiv:2212.05935.

Wang, J.; Liu, C.; Jin, L.; Tang, G.; Zhang, J.; Zhang, S.; Wang, Q.; Wu, Y.; and Cai, M. 2021. Towards Robust Visual Information Extraction in Real World: New Dataset and Novel Solution. In *AAAI 2021*, volume 35, 2738–2745.

Wang, W.; Huang, Z.; Luo, B.; Chen, Q.; Peng, Q.; Pan, Y.; Yin, W.; Feng, S.; Sun, Y.; Yu, D.; and Zhang, Y. 2022a. mmLayout: Multi-grained MultiModal Transformer for Document Understanding. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, 4877–4886. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-9203-7.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022b. Self-Instruct: Aligning Language Model with Self Generated Instructions. arxiv:2212.10560.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Arunkumar, A.; Ashok, A.; Dhanasekaran, A. S.; Naik, A.; Stap, D.; Pathak, E.; Karamanolakis, G.; Lai, H. G.; Purohit, I.; Mondal, I.; Anderson, J.; Kuznia, K.; Doshi, K.; Patel, M.; Pal, K. K.; Moradshahi, M.; Parmar, M.; Purohit, M.; Varshney, N.; Kaza, P. R.; Verma, P.; Puri, R. S.; Karia, R.; Sampat, S. K.; Doshi, S.; Mishra, S.; Reddy, S.; Patro, S.; Dixit, T.; Shen, X.; Baral, C.; Choi, Y.; Smith, N. A.; Hajishirzi, H.; and Khashabi, D. 2022c. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP 2022*.

Wang, Z.; Zhan, M.; Liu, X.; and Liang, D. 2020. DocStruct: A Multimodal Method to Extract Hierarchy Structure in Document for General Form Understanding. In *EMNLP 2020 Findings*, 898–908. Online: Association for Computational Linguistics.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models Are Zero-Shot Learners. In *ICLR 2022*. arXiv.

Wei, M.; He, Yi.; and Zhang, Q. 2020. Robust Layout-aware IE for Visually Rich Documents with Pre-trained Language Models. In *ACM SIGIR 2020*, SIGIR '20, 2367–2376. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8016-4.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; and Jiang, D. 2023. WizardLM: Empowering Large Language Models to Follow Complex Instructions. arxiv:2304.12244.

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 *KDD 2020*, 1192–1200.

Xu, Y.; Lv, T.; Cui, L.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; and Wei, F. 2021a. LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding. *arXiv:2104.08836 [cs]*.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; Zhang, M.; and Zhou, L. 2021b. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *ACL 2021*, 2579–2591. Online: Association for Computational Linguistics.

Xu, Z.; Shen, Y.; and Huang, L. 2022. MultiInstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning. arxiv:2212.10773.

Yang, X.; Yumer, E.; Asente, P.; Kraley, M.; Kifer, D.; and Giles, C. L. 2017. Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. In *CVPR 2017*, 4342–4351.

Yu, W.; Lu, N.; Qi, X.; Gong, P.; and Xiao, R. 2020. PICK: Processing Key Information Extraction from Documents Using Improved Graph Learning-Convolutional Networks. In *ICPR 2020*.

Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; and Ahmed, A. 2021. Big Bird: Transformers for Longer Sequences. arxiv:2007.14062.

Zhang, P.; Xu, Y.; Cheng, Z.; Pu, S.; Lu, J.; Qiao, L.; Niu, Y.; and Wu, F. 2020. TRIE: End-to-End Text Reading and Information Extraction for Document Understanding. In *ACM MM 2020*, 1413–1422. Seattle WA USA: ACM. ISBN 978-1-4503-7988-5.

Zhang, Y.; Bo, Z.; Wang, R.; Cao, J.; Li, C.; and Bao, Z. 2021. Entity Relation Extraction as Dependency Parsing in Visually Rich Documents. In *EMNLP 2021*, 2759–2768. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Zhao, X.; Niu, E.; Wu, Z.; and Wang, X. 2019. CUTIE: Learning to Understand Documents with Convolutional Universal Text Information Extractor. *arXiv:1903.12363 [cs]*.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2023. Large Language Models Are Human-Level Prompt Engineers. In *ICLR 2023*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. arxiv:2304.10592.

A Prompt Templates

Table 7 shows the prompt template for DocVQA(Mathew, Karatzas, and Jawahar 2021). In the first and second lines, we explain the meaning of extraction in detail to the model according to the task description in DocVQA (Mathew, Karatzas, and Jawahar 2021). Lines 3 to 6 provide placeholders for the layout-aware document and question. To avoid the model forgetting its task due to the interference of document content, the 8th line summarizes and reiterates the task requirements.

Table 8 shows the prompt template for InfographicVQA(Mathew et al. 2021). Compared with DocVQA, InfographicVQA has more complex answer sources. We describe the answer requirements in detail in lines 1 to 6, and the rest is similar to the DocVQA prompt template.

Table 9 shows the prompt template for MP-DocVQA (Tito, Karatzas, and Valveny 2023). MP-DocVQA is a multi-page question answering task, in which each question has multiple candidate page images, and we adopt the Max Confidence (Max Conf) (Mathew, Karatzas, and Jawahar 2021) setup to solve it. Specifically, the model extracts one answer on each page and gives confidence to this answer. We choose the answer with the highest confidence as the final prediction. Lines 10 to 12 of Tab. 9 instruct the model to output the answer and confidence simultaneously, and the rest is similar to Tab. 7.

Table 10 and Table 11 provide the details of Question Generation Prompt Template and Instruction Prompt Template in LATIN-Tuning.

B Datasets

To evaluate LATIN-Prompt, we conduct experiments on three document image question answering datasets, including DocVQA (Mathew, Karatzas, and Jawahar 2021), InfographicVQA (Mathew et al. 2021), and MP-DocVQA (Tito, Karatzas, and Valveny 2023).

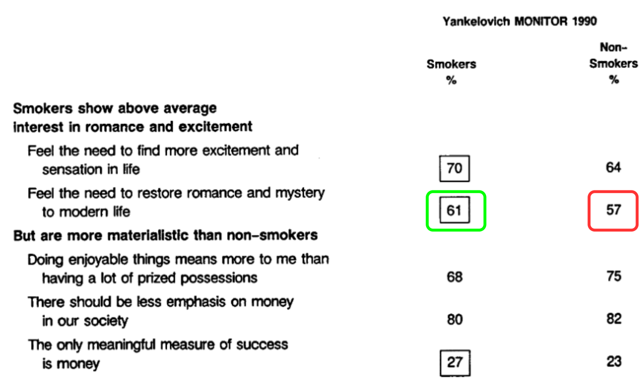
The DocVQA is an extractive question answering task and consists of 50,000 questions defined on 12,767 document images. The train split has 39,463 questions, the validation split has 5,349 questions, and the test split has 5,188 questions. DocVQA contains a large number of questions related to forms, layouts, tables and lists in the image, posing high requirements on the model's ability to understand document image layouts.

The InfographicVQA consists of 5,485 infographics, which convey information through text, graphics, and visual elements together. Compared with DocVQA, InfographicVQA emphasizes questions that require basic reasoning and arithmetic skills, and has more complex answer sources, including Document(Image)-Span, Question-Span, Multi-Span and Non-extractive.

MP-DocVQA extends DocVQA to more realistic multi-page scenarios where a document typically consists of multiple pages that should be processed together. It comprises 46,000 questions posed over 48,000 scanned pages belonging to 6,000 industry documents, and the page images contain diverse layouts. The variability between documents in MP-DocVQA is very high. The number of pages in each document varies from 1 to 20, and the number of recognized OCR words varies from 1 to 42,313.

Following common practice, we use Azure OCR results for the DocVQA provided by DUE (Borchmann et al. 2021). For the InfographicVQA and MP-DocVQA, we use offical OCR results[6].

C Case Study


Yankelovich MONITOR 1990

	Smokers %	Non-Smokers %
Smokers show above average interest in romance and excitement		
Feel the need to find more excitement and sensation in life	70	64
Feel the need to restore romance and mystery to modern life	61	57
But are more materialistic than non-smokers		
Doing enjoyable things means more to me than having a lot of prized possessions	68	75
There should be less emphasis on money in our society	80	82
The only meaningful measure of success is money	27	23

Question	What percentage of non-smokers feel the need to restore romance and mystery to modern life?
Gold Answer	(1) 57 (2) 57%
Alpaca + LATIN-Prompt	61%
Alpaca + LATIN-Tuning + LATIN-Prompt	57

Figure 6: Case study of LATIN-Tuning.

Figure 6 provides a case to compare the performance of Alpaca and Alpaca + LATIN-Tuning. After LATIN-Tuning, the Alpaca can comprehend layout more effectively.

[6]https://rrc.cvc.uab.es/?ch=17&com=introduction

#Line	Prompt
1	You are asked to answer questions asked on a document image.
2	The answers to questions are short text spans taken verbatim from the document. This means that the answers comprise a set of contiguous text tokens present in the document.
3	Document:
4	{Layout Aware Document placeholder}
5	
6	Question: {Question placeholder}
7	
8	Directly extract the answer of the question from the document with as few words as possible.
9	
10	Answer:

Table 7: DocVQA Prompt Template. The {} represents the placeholder.

#Line	Prompt
1	You are asked to answer questions asked on a document image.
2	The answer for a question in this can be any of the following types:
3	1. Answer is a piece contiguous text from the document.
4	2. Answer is a list of ”items” , where each item is a piece of text from the document (multiple spans). In such cases your model/method is expected to output an answer where each item is separated by a comma and a space. For example if the question is ”What are the three common symptoms of COVID-19?” Answer must be in the format ”fever, dry cough, tiredness”. In such cases ”and” should not be used to connect last item and the penultimate item and a space after the comma is required so that your answer match exactly with the ground truth.
5	3. Answer is a contiguous piece of text from the question itself (a span from the question)
6	4. Answer is a number (for example ”2”, ”2.5”, ”2%”, ” 2/3” etc..). For example there are questions asking for count of something or cases where answer is sum of two values given in the image.
7	Document:
8	{Layout Aware Document placeholder}
9	
10	Question: {Question placeholder}
11	
12	Directly answer the answer of the question from the document with as few words as possible.
13	
14	Answer:

Table 8: InfographicVQA Prompt Template. The {} represents the placeholder.

#Line	Prompt
1	You are asked to answer questions asked on a document image.
2	The answers to questions are short text spans taken verbatim from the document. This means that the answers comprise a set of contiguous text tokens present in the document.
3	Document:
4	{Layout Aware Document placeholder}
5	
6	Question: {Question placeholder}
7	
8	Directly extract the answer of the question from the document with as few words as possible.
9	
10	You also need to output your confidence in the answer, which must be an integer between 0-100.
11	The output format is as follows, where [] indicates a placeholder and does not need to be actually output:
12	[Confidence score], [Extracted Answer]

Table 9: MP-DocVQA Prompt Template. The {} represents the placeholder.

#Line	Prompt
1	Document:
2	{Document string with spaces and line from CSV table}
3	Randomly generate a question and corresponding answer for the above document. The answer to the question must be unique, and must be extracted from the document. To answer this question, the layout of the document must be understood. The output should be in the following format without any other text:
4	Question: [Question content]
5	Answer: [Answer content]

Table 10: Instruction Prompt Template. The {} represents the placeholder.

#Line	Prompt
1	Document:
2	{Document string with spaces and line from CSV table}
3	Question: {Question}
4	Please extract the answer of the question from the given document.

Table 11: Question Generation Prompt Template. The {} represents the placeholder.