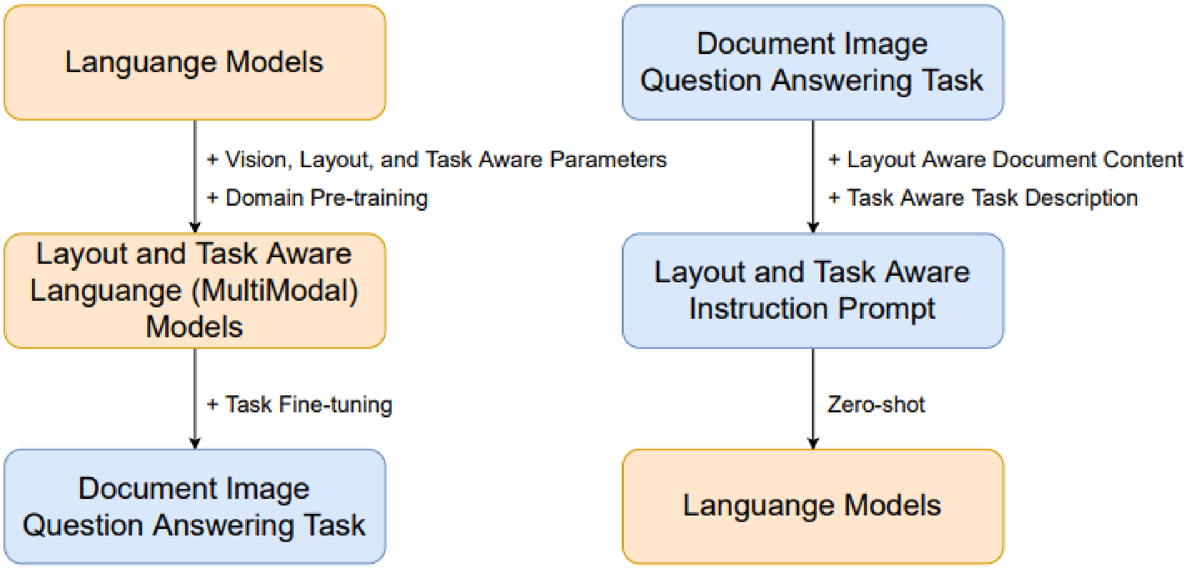
Latin-Prompt

Introduction-

Latin stands for layout and task-aware instruction prompt which consists of layout-aware document content and task-aware descriptions. The first step is getting the information using OCR tools. The second one ensures that the generated answer matches the description of the information asked.

Hangzhou, China


Languange Models
+ Vision, Layout, and Task Aware Parameters
+ Domain Pre-training
Layout and Task Aware Languange (MultiModal) Models
+ Task Fine-tuning
Document Image Question Answering Task
Document Image Question Answering Task
+ Layout Aware Document Content
+ Task Aware Task Description
Layout and Task Aware Instruction Prompt
Zero-shot
Languange Models

(a) Align models to domain

(b) Align domain to models

As we can see there are two approaches that are being discussed.
In the 1st approach, we align the model to the domain by first making a task-aware LLM and then providing fine-tuning for specific tasks.
In the 2nd approach, we align the domain to our model by making a layout and task-aware prompt and then applying zero-shot to our language model by providing the fine-tuning.
This paper approaches this problem by aligning the document image question answering to off-shelf instruction-tuning language models.

Challenges in the above approach-

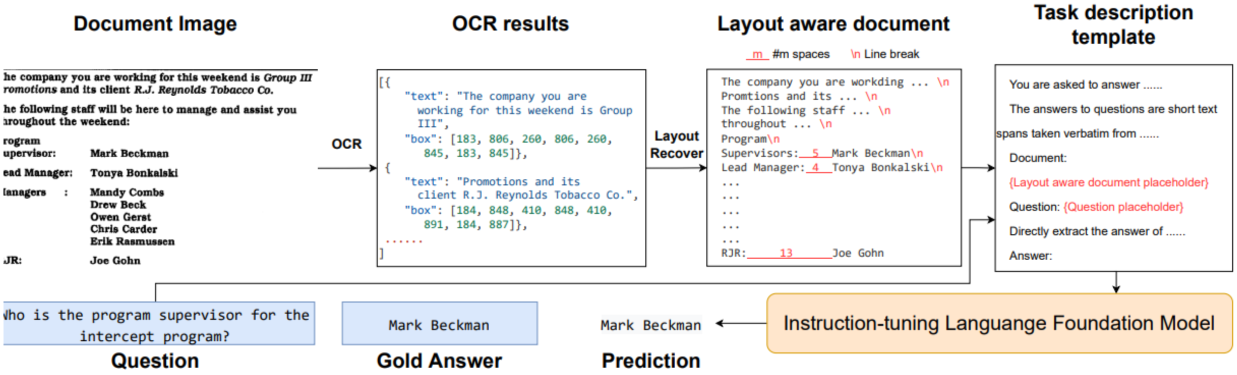
1. Directly concatenating the OCR text segments can result in a loss in the information.
2. The instruction-tuned LLMs can provide some open-ended answers instead of providing any specific answers.

So this paper provided two methods layout-aware document content and task-aware task descriptions. By equipping with LATINPrompt, instruction-tuning language foundation models can better
Understand the layout and task information in the document image
It is a zero-shot approach that doesn’t need any specific finetuning.

Related works-

1. **Visually-rich document understanding-**
   It focuses on recognizing and understanding scanned document images. Many such approaches include using CNN, and GNNs, some of the recent approaches involve pre-trained transformers like layoutLMV2 and others. DONUT provides an OCR-free approach for doing text recognition in images. The thing is these models require some task-specific fine-tuning to provide the LATIN-prompt for our large language model.
2. **Instruction Tuning of Large Language Models-**
   Instruction tuning of LLMs started with GPT-2m then GPT-3 and the launch of ChatGPT revolutionized this domain releasing many open-source LLMs like LLAMA, Falcon, stable-LLMs, and others. LORA finetuning helps in doing fine-tuning on memory-constrained systems and langchains can be used for inference these LLMs.

Methodology-



he company you are working for this weekend is *Group III romotions* and its client *R.J. Reynolds Tobacco Co.*

he following staff will be here to manage and assist you hroughout the weekend:

rogram upervisor:	Mark Beckman
ead Manager:	Tonya Bonkalski
lanagers :	Mandy Combs Drew Beck Owen Gerst Chris Carder Erik Rasmussen
JR:	Joe Gohn

```
[{
   "text": "The company you are
     working for this weekend is Group
     III",
   "box": [183, 806, 260, 806, 260,
      845, 183, 845]},
 {
   "text": "Promotions and its
     client R.J. Reynolds Tobacco Co.",
   "box": [184, 848, 410, 848, 410,
      891, 184, 887]},
 ......
]
```

```
The company you are workding ... \n
Promtions and its ... \n
The following staff ... \n
throughout ... \n
Program\n
Supervisors:__5__Mark Beckman\n
Lead Manager:_4__Tonya Bonkalski\n
...
...
...
...
...
RJR:_____13______Joe Gohn
```

You are asked to answer ......

The answers to questions are short text spans taken verbatim from ......

Document:

{Layout aware document placeholder}

Question: {Question placeholder}

Directly extract the answer of ......

Answer:

Who is the program supervisor for the intercept program?

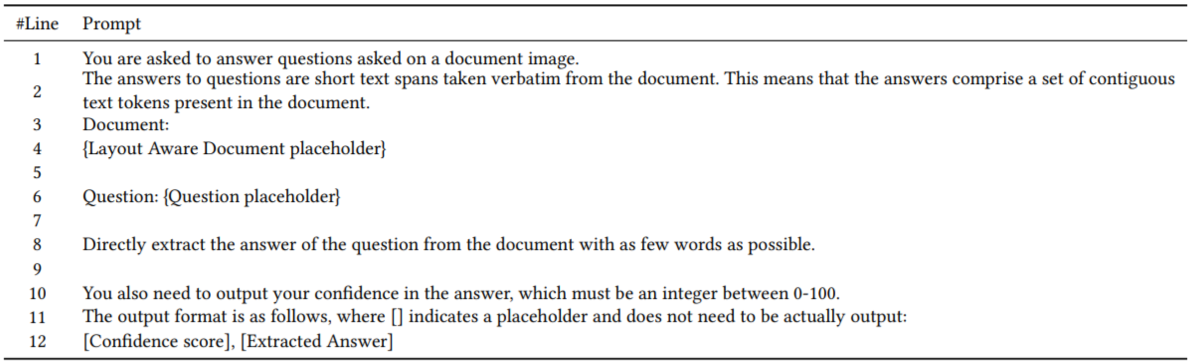
Mark Beckman

Mark Beckman

1. **Overview of Latin-Prompt-**
   As we can see the pipeline goes like the following, first we get the OCR results of our document image and then make a layout-aware document for the OCR result make a task-aware template feed it into instruction-tuning LLM, and get our answer.
2. **Layout Aware Document-**
   The core layout depends upon the usage of spaces and line breaks. So now for making a layout-aware document we first need to get OCR results and then
   1. Re-arrange those in the top to bottom and right to left
   2. Now according to the co-ordinates arrange the results in a proper document format.
   3. Calculate the character width of our document.
   4. Join different text segments that are in a single row by providing spaces between them.
   5. Now the different rows can be joined by providing line breaks between different rows.

3. Task Aware Description

Table 3: MP-DocVQA Prompt Template

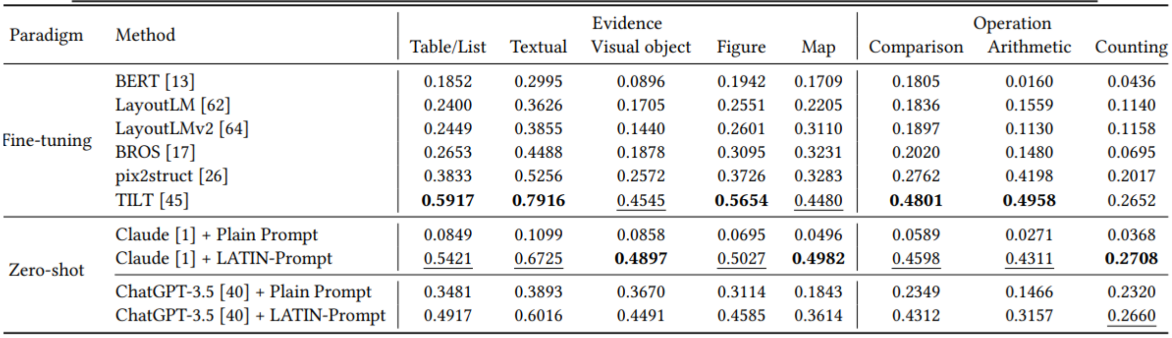


#Line	Prompt
1	You are asked to answer questions asked on a document image.
2	The answers to questions are short text spans taken verbatim from the document. This means that the answers comprise a set of contiguous text tokens present in the document.
3	Document:
4	{Layout Aware Document placeholder}
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]

This is a template for giving the prompt for the question this prompt can be made using langchains and then can be fed into the LLMs a bit of prompt engineering can also be done to get better results. This template needs to be fed because the answers for any LLM are more open-ended so instead of providing any direct answers it is going to provide any general answers.

Experimentations-

They evaluated this pipeline on DocVQA



Paradigm	Method	Evidence					Operation		
		Table/List	Textual	Visual object	Figure	Map	Comparison	Arithmetic	Counting
Fine-tuning	BERT [13]	0.1852	0.2995	0.0896	0.1942	0.1709	0.1805	0.0160	0.0436
	LayoutLM [62]	0.2400	0.3626	0.1705	0.2551	0.2205	0.1836	0.1559	0.1140
	LayoutLMv2 [64]	0.2449	0.3855	0.1440	0.2601	0.3110	0.1897	0.1130	0.1158
	BROS [17]	0.2653	0.4488	0.1878	0.3095	0.3231	0.2020	0.1480	0.0695
	pix2struct [26]	0.3833	0.5256	0.2572	0.3726	0.3283	0.2762	0.4198	0.2017
	TILT [45]	**0.5917**	**0.7916**	0.4545	**0.5654**	0.4480	**0.4801**	**0.4958**	0.2652
Zero-shot	Claude [1] + Plain Prompt	0.0849	0.1099	0.0858	0.0695	0.0496	0.0589	0.0271	0.0368
	Claude [1] + LATIN-Prompt	0.5421	0.6725	**0.4897**	0.5027	**0.4982**	0.4598	0.4311	**0.2708**
	ChatGPT-3.5 [40] + Plain Prompt	0.3481	0.3893	0.3670	0.3114	0.1843	0.2349	0.1466	0.2320
	ChatGPT-3.5 [40] + LATIN-Prompt	0.4917	0.6016	0.4491	0.4585	0.3614	0.4312	0.3157	0.2660

A summary of the results shows that the zero-shot prompting on these LLMs provided much better results.

Limitations-

1. As we can see the the question totally depends upon the textual information that is being extracted from OCR thus limiting its usage to visual figures in our document. By using GPT-4 it showed much better results.
2. Its performance depends upon the OCR tool's performance.
3. An automated template is required for generalized extraction of our information.
4. It only provides the document question answering it is not applicable to image understanding.