DOLMA: Visual Instruction Tuning for Document AI

Anonymous ACL submission

Abstract

The rapid expansion of Vision-Language Models (VLMs) has spurred research into their applicability across various domains. While VLMs excel in understanding environmental contexts, their effectiveness declines with visually-rich scanned documents. Although some VLMs use Optical Character Recognition (OCR) to mitigate this, OCR alone is insufficient for the complex textual and visual insights required. Developing tailored models for Document AI applications also demands substantial labeled data and high training costs. To address these challenges, we conducted experiments with various models, data types, architectures, and training methodologies. Based on our findings, we introduce DOLMA, an OCR-free vision-language model designed for diverse Document AI applications in a zero-shot setting. Despite having a moderate parameter count of 7 billion, DOLMA performs on par with models ten times larger on numerous Document AI benchmarks. The complete model, including weights, training data, and code, is publicly available.

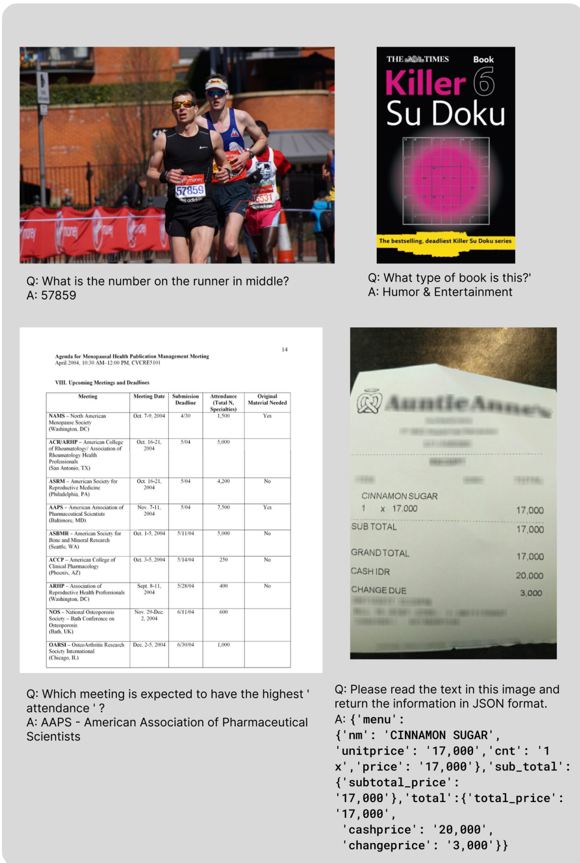

Figure 1: The training pipeline of DOLMA.

1 Introduction

In recent years, there has been a notable surge in interest surrounding the understanding of visually-rich scanned documents (VRD). The latter encompasses PDFs and document images such as business forms, receipts, driving licenses and invoices. The understanding and digitization of those document images entails intricate tasks such as document visual question answering (DVQA), document classification (CLS), and key information extraction (KIE).

Traditional approaches address these challenges by employing Optical Character Recognition (OCR) alongside handcrafted rules or layout analysis. However, these methods often necessitate post-processing steps, potentially limiting the efficacy and use of those models. In recent years, the Document AI community has proposed various transformer-based architectures providing remarkable progress on VRD understanding (VRDU). Notably, Transformer-based models like LayoutLM and its variants have showcased advancements by integrating OCR, image, and layout information. Nevertheless, recent efforts in OCR-free, end-to-end document understanding from images indicate a shift towards more versatile models, minimizing task-specific engineering and reducing reliance on external components during inference.

In this study, we aimed to explore various design choices to identify the optimal combination of models, data, and architecture based on our experiments. We also imposed constraints on model size and resource usage to demonstrate the most efficient and cost-effective approach to developing a model that can perform on par with other state-of-the-art models. To assess the quality and utility of the model, we evaluate it based on the following properties:

- **Property 1: Multitasking.** The model is expected to perform the main Document AI tasks such as document classification, document question answering, and key information extraction.
- **Property 2: OCR-independence.** Key information in documents is many times incorporated in non-optical characters such as logos, images, charts and other visuals. OCR-dependent models do not have the capability to extract this information. Nonetheless, we consider the models that do not necessarily rely on OCR yet can improve the results using OCR information. We call them OCR-enhanced models as they can still perform without relying on OCR.
- **Property 3: Instruction following.** The typical usage of information extraction from documents is related to structuring image data into programmatically readable formats such as JSON, XML or CSV. As the use cases of information extraction can be different, the Document AI foundation model should have the ability to follow the user's instruction and generate extracted output in the required format (including notation format such as JSON and its internal structure such as key/value hierarchy).
- **Property 4: Template independence.** The Document AI foundation model should be able to provide competitive performance on the same documents even if the templates are different.

We outline the following roadmap of experiments, which will be discussed in subsequent sections. The modalities we consider include a Vision encoder, a Language decoder, and a bridge connector between them. We establish two stages for training: (1) pretraining and (2) fine-tuning. Stage (1) is designed to enable the model to acquire OCR capabilities, while stage (2) focuses on task-specific supervised instruction tuning.

During stage (1), we experiment with (a) the design of the bridge connector and (b) the choice of language model. For (a), we report findings using design choices from LLAVA (Liu et al., 2023) for the linear projection strategy, QwenVL (Bai et al., 2023) for the cross-attention strategy, and Idefics2 (Laurençon et al., 2024) for the projection + perceiver-resampler strategy. For (b), we evaluate Vicuna (Zheng et al., 2023), LLAMA 3 (Team, 2024), and Phi 3 (Abdin et al., 2024). We select Vicuna as a well-established instruction model, LLAMA 3 as a state-of-the-art large language model, and Phi to assess the impact of using smaller models.

During stage (2), our primary focus is on training strategies. We discovered that training all modalities yields the best results. Consequently, the main variable is the strategic approach to the largest modality, which in our case is the LLM. We report on three strategies: fine-tuning only the attention layers of the LLM, full LLM fine-tuning, and applying LoRA on top of the LLM. In all three scenarios, we fully fine-tune the vision encoder and the bridge connector. All the aforementioned experiments are conducted using 8 H100 GPU spot instances to ensure the fastest possible training time. Building on our observations, we propose DOLMA, "Document Optimized Language Model for Automation," which adheres to the four principles outlined above. DOLMA is a 7-billion-parameter Vision-Language Model (VLM) that achieves results on various Document AI benchmarks on par with state-of-the-art models, even matching the performance of models that are ten times larger.

2 Related Work

The advent of ChatGPT represents a significant advancement in the domain of Large Language

Models (LLMs). LLMs constitute a substantial area of study in natural language processing, specializing in processing and generating textual content for tasks like language translation, summarizing, question answering, and text completion.

2.1 LLMs

Through extensive pre-training on textual datasets, LLMs acquire proficiency in contextual relationships and linguistic patterns. The transformative impact of transformers, as introduced in "Attention is All You Need," has played a pivotal role in the success of LLMs, leading to the development of pre-trained models such as BERT, BART, and others. This success has spurred further exploration into LLMs like OPT (Zhang et al., 2022), BLOOM (Workshop et al., 2023), PaLM (Chowdhery et al., 2022), and LLaMA (Touvron et al., 2023a). Particularly noteworthy is LLama3 (Team, 2024), an open-source LLM demonstrating comparable or superior performance to both open and closed-source models. The open-source nature of Llama has encouraged numerous researchers to build models on top of it, employing diverse training strategies and architectural modifications, including models like Vicuna (Zheng et al., 2023) and Alpaca (Taori et al., 2023).

2.2 Multimodal LLM

In the realm of multimodal AI, Multimodal Language Models (MLMs) have emerged as a significant focus. Unlike text-to-text models, MLMs are designed to comprehend and generate content across multiple modalities, often integrating text and images. These models exhibit proficiency in tasks requiring a fusion of textual and visual understanding, such as generating image captions, image-text matching, visual question answering and contextualizing information in mixed-media environments. Training MLMs involves leveraging datasets encompassing both textual and visual information, facilitating the capture of intricate relationships between words and images. Notable MLMs include GPT-4V, LLaVA (Liu et al., 2023), Gemini Pro Vision, and others.

2.3 Document AI

Transformer-based architectures have found success in Visual Document Understanding (VRDU) and Document Visual Question Answering (DVQA) tasks (Wang et al., 2023b; Ye et al., 2023a,b; Kim et al., 2022; Hong et al., 2023; Bai et al., 2023). Recent works like LayoutLM (Huang et al., 2022) focus on pre-training a language model, such as BERT (Devlin et al., 2019), alongside an OCR-based engine to comprehend both textual content and layout information in document images. This approach extends traditional language models by incorporating positional embeddings that encode the spatial arrangement of words on a page, enabling the model to capture both structural relationships and contextual meanings. Recent works, such as DocLLM (Wang et al., 2023a), integrate lightweight visual information by utilizing spatial positions and dimensions of text tokens obtained through OCR. It employs separate vectors to represent vision and image modalities, extending the self-attention mechanism of the transformer architecture to compute their interdependencies in a disentangled manner. Alternative methods, exemplified by DONUT (Kim et al., 2022), leverage transformer architectures for document understanding tasks, focusing on extracting information directly from the document's content without relying on OCR. DONUT employs the Swin transformer (Liu et al., 2021) as the vision encoder and BART (Lewis et al., 2019) as the decoder model. A more general model, Qwen-VL (Bai et al., 2023), incorporates an adapter with cross-attention layers to attenuate vision encoder embeddings with language embeddings. Qwen-VL, trained on a large corpus of both regular and document images, demonstrates proficiency in tasks such as image captioning, question answering, visual grounding, and text reading.

As shown in the Table 1, among the models mentioned above, GPT4-V, Gemini Pro Vision, LLaVA and Qwen-VL are the only models that satisfy the 4 properties we seek in a Document AI foundation model.

In summary, our review highlights the significant strides in LLMs, the emergence of multimodal AI with MLMs, and the successful applications of transformer architectures in VrDu

Table 1: Comparison of different models across the 4 properties we seek

Model	Property 1 Multitasking	Property 2 OCR-free	Property 3 Instruction following	Property 4 Template independent
GPT4-V	✓	✓	✓	✓
Gemini-Pro-Vision	✓	✓	✓	✓
Donut	×	✓	×	×
LayoutLMV3	✓	×	×	✓
DocLLM	✓	×	×	✓
Qwen-VL	✓	✓	✓	✓
LLaVA	✓	✓	✓	✓
CogAgent	✓	✓	✓	✓
UReader	✓	✓	✓	✓
DocOwl	✓	✓	✓	✓
DOLMA (ours)	✓	✓	✓	✓

and DVQA tasks. These advancements lay the groundwork for versatile models like Qwen-VL, showcasing the evolving landscape of AI and machine learning.

3 Analysing the design possibilities for vision-language models

In this section, we will examine the various design choices for vision-language models in Document AI as documented in the open-source literature and present our findings. For our experiments, we will utilize the IIT-CDIP dataset and our own PDF-generated dataset for pretraining. Additionally, we will employ various benchmarking datasets, including DocVQA (Mathew et al., 2021b), CORD-V2 (Park et al., 2019), Infographics-VQA (Mathew et al., 2021a), ICDAR-SROIE (ICDAR, 2019), Chart-QA (Masry et al., 2022), OCR-VQA (Mishra et al., 2019), RVL-CDIP (Harley et al., 2015), and TextVQA (Singh et al., 2019), for fine-tuning experiments.

3.1 The design of the bridge connector

Vision-language models comprise two modalities: vision and language. While there are numerous models available for these modalities, it is essential for them to effectively "communicate" with each other. We explore three types of connectors: linear projection, cross-attention, and projection + perceiver-resampler.

For our experiments, we fixed Swin Base (Liu et al., 2021) as the vision model and Vicuna (Zheng et al., 2023) as the language model across all three bridge connector designs. During the pretraining stage, we trained both the vision model and each connector while keeping the LLM model frozen. Utilizing a total of 3 million image-text pairs from the IIT-CDIP (Soboroff, 2022) dataset and our in-house PDF-generated data (details of which will be discussed in subsequent sections), we pretrained the model for the text extraction task, thereby imparting OCR capabilities. We trained each configuration for a total of one epoch. Among the three connectors, only the linear projector connector successfully converged. We hypothesize that extended training might enhance the performance of the other connectors; however, given our experimental setup and resource constraints, the linear projector layer demonstrated the best results.

Insight 1

The linear projector layer is the fastest and most straightforward method to connect the vision and language models, achieving training convergence with just one epoch on 3 million image-text pairs.

3.2 The design of the Vision model

For the vision model, we selected the SWIN Transformer (Liu et al., 2021) (base, large) and the Vision Transformer (Dosovitskiy et al., 2020) (CLIP ViT-L/14). Similar to the previous section, we conducted the pretraining stage by keeping the LLM frozen and training the vision model along with the connector. We fixed the bridge connector to the linear projection design and experimented with different vision encoders.

Our findings revealed that 3 million image-text pairs and one epoch of training were insufficient for the vision models to acquire text extraction capabilities. We utilized pretrained weights for each model, but none of the trainings converged except for the Swin Base model. For Swin Base, we used weights from the DONUT (Kim et al., 2022) model, which had been pretrained for the text extraction task using over 11 million image-text pairs and trained for 200K steps with a batch size of 196. The DONUT model employed Swin Base, and similarly, when we integrated Swin Base into our architecture, the training eventually converged.

Insight 2

Vision models require tens of millions of text extraction pretraining data and extended training sessions to develop OCR capabilities.

3.3 The design of the LLM model

We selected Vicuna 1.5 (Zheng et al., 2023), LLama3 (Team, 2024), and Phi (Abdin et al., 2024) for our experiments. Following the success of LLaVA, we chose Vicuna as our initial model. We included LLama3 to evaluate the impact of a relatively newly released state-of-the-art model. Additionally, we decided to use the Phi-3 model to assess the performance of a model with fewer than seven billion parameters. The vision encoder employed was Swin Base, and the projection design was used as the bridge connector. During the pretraining stage, we kept the LLM frozen and only pretrained the vision model and bridge connector. At the conclusion of the experiments, both Phi and LLama3 failed to converge during training, whereas only Vicuna was able to achieve near-zero loss for the text extraction task.

The objective of this experiment is to compare different fine-tuning strategies within the given setup, based on performance across various benchmarks as well as computational complexity. Our experimental configuration includes Swin Base (pretrained with Donut) as the vision encoder, a projection as the bridge connector, and Vicuna as the LLM decoder. Swin Base and the projection were pretrained as described in the previous sections. During the fine-tuning stage, we continue training the entire model (all three modalities) on benchmark datasets. While we fully train the Vision Encoder and the bridge connector, we apply three different strategies for training the LLM: full LLM fine-tuning, fine-tuning only the attention layers of the LLM, and applying LoRA to the LLM. We focus these strategies on the LLM because it constitutes 98% of the model's weights, making the full training of the Vision Encoder and bridge connector less computationally intensive. For LoRA, we set r=128. Each strategy involves training the model for a total of four epochs.

We report the results of these three strategies on selected benchmark datasets in Table 2. For each benchmark dataset, we use the official train, validation, and test splits. Evaluation results are presented on the test split, except for the TextVQA dataset, as the test set labels are not available. For each dataset, we use the corresponding evaluation metric commonly employed in the literature. The experiments indicate that LoRA training produces the lowest results across all benchmarks compared to full and attention-only fine-tuning. We also experimented with changing the compression dimension of LoRA to r=256 but obtained similar, near-identical scores.

Attention-only and full fine-tuning yield significantly better results, with each method outperforming the other on different benchmarks. For instance, the attention-only method outperforms full fine-tuning by 2 points on DocVQA, whereas full fine-tuning scores 3 points higher on SROIE compared to attention-only fine-tuning. Overall, the average results are very close, with attention-only fine-tuning being marginally better than full fine-tuning.

Insight 3

Given our experimental setup, fine-tuning only the attention layers of the LLM is equivalent to full LLM fine-tuning in terms of performance. And both are better then LORA in terms of evaluation scores.

In the next section, we will take the best model from our experiments and compare it with other Document Language Models (DocLMs).

Model	DocVQA [ANLS]	CORD V2 [F1]	Info VQA [ANLS]	SROIE [F1]	Chart QA [Rel. EM]	OCR VQA [EM]	RVL-CDIP [Accuracy]	TextVQA [VQA Score]
Attention	**0.75**	**0.76**	**0.3633**	0.76	**0.5952**	0.722	**0.94**	0.4644
LORA	0.47	0.463	0.28	0.53	0.442	0.504	0.91	0.2498
Full	0.73	**0.76**	0.30	**0.79**	0.5948	**0.741**	**0.94**	**0.4698**

Table 2: Performance of fine-tuning strategy on various benchmarks.

4 DOLMA

4.1 Architecture

We constructed DOLMA by incorporating the insights derived from our previous experiments. The architecture consists of Swin Base as the visual encoder and Vicuna as the LLM decoder, connected via a projection layer serving as the bridge connector. Swin Base is a Swin Transformer with a patch size of 4 and a window size of 10, comprising fewer than 100 million parameters. This model is pretrained with Donut and was trained on 11 million image-text pairs.

Following the approach of LLaVA, we employ a 2-layer MLP as the projection layer, utilizing the GELU activation function between layers, resulting in a total of fewer than 30 million parameters. Vicuna 1.5 serves as the LLM, featuring 7 billion parameters. It is trained by fine-tuning Llama 2 (Touvron et al., 2023b) on user-shared conversations collected from ShareGPT.

4.2 Datasets

We utilized two datasets for pretraining and eight datasets for fine-tuning our model. Specifically, IIT-CDIP (Soboroff, 2022) and our own PDF-generated datasets were employed for pretraining the vision encoder and the projection layer. The other datasets—DocVQA, CORD V2, Infographics VQA, SROIE, Chart QA, OCR VQA, RVL-CDIP, and TextVQA—were used to train the full model, with fine-tuning applied only to the attention layers of the LLM. For each dataset, we constructed a unique instruction prompt to ensure that the model retains its instruction-following capabilities. The prompts can be found in the appendix.

IIT-CDIP (Soboroff, 2022). "CDIP" stands for "Complex Document Information Processing" and "IIT" stands for "Illinois Institute of Technology" who originally built the dataset. The dataset consists of documents from the states' lawsuit against the tobacco industry in the 1990s. Labels are the text extracted from the dataset using Tesseract. Ovearall, the datasets consist of around 7 million documents. As the quality of the dataset is crucial for our task we applied some pre processing techniques and removed all the images that had almost no text and ha low quality OCR. The final shortlisted number is around 2 million image-text pairs.

PDF-archive. We downloaded an additional 1 million pages of open-source archive PDF documents and extracted the text from them using PyPDF. To obtain these documents, we utilized the arXiv API to download various scientific papers. To get png images from PDFs we set the image zoom equal to 1.8. Given that the archive data is too clean and perfect, it would not adequately represent the everyday scanned document types that Document AI models typically encounter. Therefore, we employed Augraphy (Project) to augment the PDF data by adding random marks, paper folding effects, various colors, and blur effects. Example of an augmented images can be found in the appendix.

CORD V2 (Park et al., 2019). Public benchmark of 1000 receipts images. We follow the official split of 800 - train, 100 - validation and 100 - test samples. The text is fully in Latin characters. Each image may contain different fields with the total number of unique fields amounting to 30. Our data generation process imposes instruction to extract either all or a subset of those fields in a predefined structured format (e.g. JSON) or in unstructured, question-answering manner.

ICDAR SROIE (ICDAR, 2019). A dataset of 1000 whole scanned receipt images. The text is in English characters and each image contains around 4 main fields. The dataset comes with JSON structured annotation intended for KIE task. We separate 347 images for the testing set and utilize the rest in training.

DocVQA (Mathew et al., 2021b). Document question answering dataset consisting of 50k records sourced from the Industry Documents Library, maintained by the UCSF. The dataset includes mixture of printed, typewritten and handwritten documents that are letters, memos, notes, reports and other types of documents. We follow the official split with 40k - train, 5k - validation and 5k - test sets.

RVL-CDIP (Harley et al., 2015). Relatively larger dataset of 400k images used for document classification task. The dataset includes documents such as letter, memo, email and others. Overall, there are 16 unique classes with 25k images per class. We follow the official split of 320k - training, 40k - validation and 40k - testing splits.

Infographic VQA (Mathew et al., 2021a). Similar to typical VQA task, task is to answer questions asked on a given infographic image. Similar to extractive QA framework popular in NLP, and the DocVQA dataset, here question-answers are primarily extractive type. But there are a small percentage of questions where answers arr not extractive. There are 30 K questions and 5K Images in the dataset. Images are collected from the Internet. Questions and answers are manually annotated.

ChartQA (Masry et al., 2022). A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. The datasets is split into 30K train, 2K validation and 2.5K test image-question-answer pairs.

OCR VQA (Mishra et al., 2019). OCR-VQA dataset contains 207572 images and associated question-answer pairs. They provide questions inquiring about title, author, edition, year and genre of the book and corresponding ground-truth answer. This dataset contains approximately 1 million QA pairs.

Text VQA (Singh et al., 2019). TextVQA requires models to read and reason about text in images to answer questions about them. Specifically, models need to incorporate a new modality of text present in the images and reason over it to answer TextVQA questions.

4.3 Training Details

The training process consists of multiple steps, as outlined in Figure 2.

First, we pre-trained the Swin Base and the MLP projector using the IIT-CDIP and PDF-

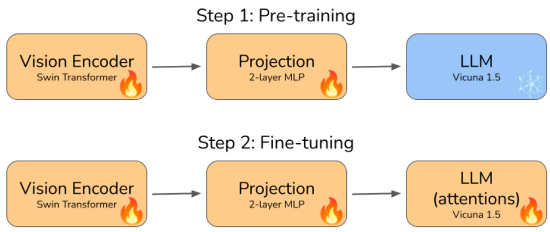

Figure 2: The training pipeline of DOLMA.

archive data. The objective of this pretraining stage is to enable the model to acquire OCR capabilities and learn to project the visual embeddings into the LLM embedding space. During this stage, the language model is kept frozen. We pre-trained the model for 1 epoch, with a batch size of 16 per device, a learning rate of $2e{-}4$, and a cosine learning rate scheduler with 3% warmup steps. We used the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e{-}8$. Given the importance of image resolution in Document AI, we increased the resolution of images to 1280x960 pixels and applied padding when necessary.

Second, we unfroze the entire model and continued with the fine-tuning process. As suggested by Insight 3, we fine-tuned only the attention layers of the LLM. The model was trained for 10 epochs, with a batch size of 10, a learning rate of $2e{-}5$, and a cosine learning rate scheduler with 3% warmup steps. Similarly, we used the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e{-}8$. The image resolution was maintained at 1280x960 pixels.

The training was conducted using 8x H100 80GB GPUs[1].

4.4 Qualitative analysis and benchmark results

We compare DOLMA with models that satisfy the four properties outlined in Table 1. The evaluation scores are reported in Table 3. All scores are sourced from their respective papers. For scores that were not directly available, we referenced other papers: specifically, the OCR VQA score of Qwen VL was taken from the CogAgent paper, and the InfographicsQA, ChartQA, and TextVQA scores of Donut

[1]Cloud resources were generously provided by AWS

Model	DocVQA [ANLS]	CORD V2 [F1]	Info VQA [ANLS]	Chart QA [Rel. EM]	OCR VQA [EM]	RVL-CDIP [Accuracy]	TextVQA [VQA Score]
DOLMA (ours)	0.75	0.76	0.363	0.595	0.722	0.94	0.464
Donut	0.675	0.841	0.116	0.418	-	0.95	0.435
Qwen-VL	0.651	-	0.354	0.657	0.757	-	0.638
UReader	0.654	-	0.422	0.593	0.411	-	0.576
DocOwl	0.622	-	0.382	0.574	-	-	0.526
CogAgent	0.816	-	0.445	0.684	0.75	-	0.761

Table 3: Comparison of document AI models on various Document AI tasks.

were sourced from the UReader paper. For each benchmark dataset, we used the official train, validation, and test splits. Evaluation results are reported on the test split, except for the TextVQA dataset, where test set labels are unavailable. We employed the evaluation metrics commonly used in the literature for each dataset.

DOLMA outperforms Donut in all tasks except for CORD V2 and RVL-CDIP. The reason for this discrepancy is the relative simplicity of these tasks and the fact that the evaluation used task-specific fine-tuned models, meaning that the models were fine-tuned on a single dataset for many epochs, as described in the Donut paper. Nevertheless, DOLMA managed to outperform Donut in the DocVQA tasks under the same training conditions.

Overall, DOLMA demonstrated performance on par with models such as Qwen VL and DocOWL, even though the vision encoders in these models are 20x and 5x larger in parameter size, respectively. For the DocVQA task, DOLMA outperforms all models except CogAgent. It is important to note that while the other models listed have fewer than 10 billion parameters, CogAgent has 17 billion parameters. As the scores illustrate, model size has a significant impact on performance in our case.

5 Conclusion and Future Work

In this paper, we conducted experiments to understand the requirements for building a Vision-Language model for Document AI tasks. Our findings highlight the effectiveness of different model architectures, model sizes, pretraining, and fine-tuning strategies. Based on these insights, we introduced DOLMA, an OCR-free, instruction-following vision-language model that can be utilized for various Document AI tasks. We demonstrated that DOLMA can perform on par with larger VLM models, despite being trained on fewer data samples and with fewer resources.

In future research, we plan to investigate the possibility of scaling DOLMA to handle multilingual and multi-page documents.

6 Limitations

While DOLMA demonstrates promising results in various Document AI tasks, several limitations must be acknowledged:

1. *Data Diversity*: Although we utilized a substantial amount of data for pretraining and fine-tuning, the datasets may not fully capture the diversity of real-world documents. This could limit the model's generalizability to unseen document types and formats.

2. *Model Size*: Despite DOLMA's competitive performance with a moderate parameter count of 7 billion, it remains computationally intensive. This may pose challenges for deployment in resource-constrained environments.

3. *OCR Capabilities*: While DOLMA is designed to be OCR-free, its performance in extracting text from highly complex or degraded documents may still lag behind specialized OCR systems. Further improvements are needed to enhance its robustness in such scenarios.

4. *Multilingual and Multi-page Documents*: Our current experiments focus primarily on single-page, monolingual documents. The model's effectiveness in handling multilingual and multi-page documents remains unexplored and warrants further investigation.

5. *Training Costs*: Although we aimed to minimize training costs, the process still requires significant computational resources, particularly for fine-tuning. This could be a barrier for smaller research groups or organizations

with limited access to high-performance computing resources.

6. *Evaluation Metrics*: The evaluation metrics used in our experiments are standard in the literature, but they may not fully capture the nuanced performance of the model in practical applications. Future work should consider more comprehensive evaluation frameworks.

7. *Ethical Considerations*: As with any AI model, there are ethical considerations related to data privacy and potential biases in the training data. These issues need to be addressed to ensure the responsible deployment of DOLMA.

By acknowledging these limitations, we aim to provide a balanced view of our work and highlight areas for future research and improvement.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

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)*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogagent: A visual language model for gui agents.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *Preprint*, arXiv:2204.08387.

ICDAR. 2019. Sroie. Available at: `https://rrc.cvc.uab.es/?ch=13&com=introduction`.

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 *European Conference on Computer Vision (ECCV)*.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models?

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *Preprint*, arXiv:2103.14030.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. 2021a. Infographicvqa.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021b. Docvqa: A dataset for vqa on document images. *Preprint*, arXiv:2007.00398.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

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.

The Augraphy Project. Augraphy: an augmentation pipeline for rendering synthetic paper printing, faxing, scanning and copy machine processes.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read.

Ian Soboroff. 2022. Complex document information processing (cdip) dataset, national institute of standards and technology.

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. https://github.com/tatsu-lab/stanford_alpaca.

Meta LLaMA Team. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023a. Docllm: A layout-aware generative language model for multimodal document understanding. *Preprint*, arXiv:2401.00908.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023b. Cogvlm: Visual expert for pretrained language models.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella

Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Alshaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023a. mplug-docowl: Modularized multimodal large language model for document understanding.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Alex Lin, and Fei Huang. 2023b. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Prompts for the pertaining stage

System

You are a helpful language and vision assistant. You are able to understand the visual content that the user provides. "

User

Extract all the text from the document.

B Prompts for the fine-tuning stage

Prompt template for CORD [task: KIE]

```
Please read the text in this image and return the information in JSON format.
```

```
    The nested JSON should have the following keys: menu, void_menu,
   subtotal, total. Each key has subkeys as listed below (with descriptions
   in brackets):

menu:
 - nm (name of menu)
 - num (identification # of menu)
 - unitprice (unit price of menu)
 - menu.cnt (quantity of menu)
 - discountprice (discounted price of menu)
 - price (total price of menu)
 - itemsubtotal (price of each menu after discount applied)
 - vatyn (whether the price includes tax or not)
 - etc (others)
 - sub_nm (name of submenu)
 - sub_unitprice (unit price of submenu)
 - sub_cnt (quantity of submenu)
 - sub_price (total price of submenu)
 - sub_etc (others)

void_menu:
 - nm (name of menu)
 - price (total price of menu)

subtotal:
 - subtotal_price (subtotal price)
 - discount_price (discounted price in total)
 - service_price (service charge)
 - othersvc_price (added charge other than service charge)
 - tax_price (tax amount)
 - etc (others)

total:
 - total_price (total price)
 - total_etc (others)
 - cashprice (amount of price paid in cash)
 - changeprice (amount of change in cash)
 - creditcardprice (amount of price paid in credit/debit card)
 - emoneyprice (amount of price paid in emoney, point)
 - menutype_cnt (total count of type of menu)
 - menuqty_cnt (total count of quantity)
```

Prompt template for SROIE [task: KIE]

```
Please read the text in this image and return the information in JSON format.
    The JSON should have the following keys: company, date, address, total.
```

Prompt template for DocVQA [task: VQA]

```
"Please read the text in this image and answer to the question: {question}\n
```

```
    <image>"
```

Prompt template for InfographicVQA [task: VQA]

```
"Given this infographic image, {question}\n<image>"
```

Prompt template for TextVQA [task: VQA]

```
"Given the image, {question}\n<image>"
```

Prompt template for OCRVQA [task: VQA]

```
"Here is an image of a book cover, {question}\n<image>"
```

Prompt template for RVL-CDIP [task: classification]

```
'Please classify the given image to one of the following classes: ["letter",
     "memo", "email", "filefolder", "form", "handwritten", "invoice", "
    advertisement", "budget", "news article", "presentation", "scientific
    publication", "questionnaire", "resume", "scientific report", "
    specification"].'
```

Prompt template for CartQA [task: VQA]

```
Given this image of a chart, {question}\n<image>"
```

C Samples from our PDF-arxiv dataset