

ProcTag: Process Tagging for Assessing the Efficacy of Document Instruction Data

Yufan Shen^{1*}, Chuwei Luo^{2*†}, Zhaoqing Zhu^{2*}, Yang Chen¹,
Qi Zheng², Zhi Yu^{1†}, Jiajun Bu¹, Cong Yao²

¹Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems,
College of Computer Science, Zhejiang University

²Alibaba Group

syficy,22351044,yuzhirenzhe,bjj@zju.edu.cn

luochuweizhaoqing.z.zhengqisjtu.yaocong2010@gmail.com

Abstract

Recently, large language models (LLMs) and multimodal large language models (MLLMs) have demonstrated promising results on document visual question answering (VQA) task, particularly after training on document instruction datasets. An effective evaluation method for document instruction data is crucial in constructing instruction data with high efficacy, which, in turn, facilitates the training of LLMs and MLLMs for document VQA. However, most existing evaluation methods for instruction data are limited to the textual content of the instructions themselves, thereby hindering the effective assessment of document instruction datasets and constraining their construction. In this paper, we propose ProcTag, a data-oriented method that assesses the efficacy of document instruction data. ProcTag innovatively performs tagging on the execution process of instructions rather than the instruction text itself. By leveraging the diversity and complexity of these tags to assess the efficacy of the given dataset, ProcTag enables selective sampling or filtering of document instructions. Furthermore, DocLayPrompt, a novel semi-structured layout-aware document prompting strategy, is proposed for effectively representing documents. Experiments demonstrate that sampling existing open-sourced and generated document VQA/instruction datasets with ProcTag significantly outperforms current methods for evaluating instruction data. Impressively, with ProcTag-based sampling in the generated document datasets, only 30.5% of the document instructions are required to achieve 100% efficacy compared to the complete dataset.

Introduction

Document visual question answering (VQA), which closely aligns with the general document artificial intelligence (Cui et al. 2021; Han et al. 2023), is currently a significant research and application area.

With the remarkable successes achieved by Large Language Models (LLMs) (OpenAI 2022; Brown et al. 2020; Touvron et al. 2023a,b; Bai et al. 2023a) and Multimodal Large Models (MLLMs) (OpenAI 2023; Bai et al. 2023b; Liu et al. 2023b,a; Ye et al. 2023b) across various real-world tasks, the application of these models (Zhang et al. 2023;

*These authors contributed equally.

†Corresponding Author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

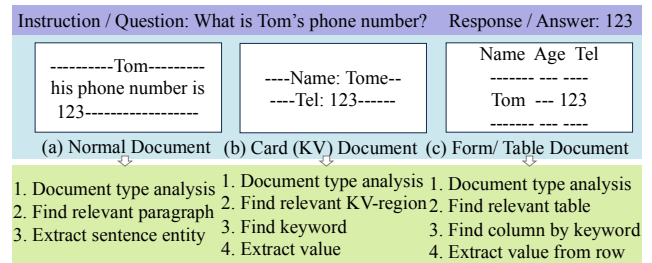


Figure 1: The same instruction text can lead to entirely different execution processes when applied to different types of documents.

Ye et al. 2023a; Yang et al. 2023; Shi et al. 2023; Liu et al. 2023c) for the document VQA has seen substantial advancements.

To achieve good document VQA results by LLMs/MLLMs, recent works (Ye et al. 2023a; Zhang et al. 2023) perform instruction tuning (Wei et al. 2022a; Ouyang et al. 2022) on the document instruction datasets. To facilitate effective instruction tuning, the instruction datasets should better possess high efficacy, which means that the instructions in the dataset are expected to be sufficiently diverse and complex (Xu et al. 2023; Mukherjee et al. 2023; Wang et al. 2024). Instruction datasets are often collected through manual annotation or generated using LLMs/MLLMs (Taori et al. 2023; Ding et al. 2023). Building upon this, recent works (Lu et al. 2023; Li et al. 2023) have proposed several data-oriented methods to evaluate the efficacy of instruction datasets, thereby facilitating the selection of data that is both diverse and complex. These methods have been proven effective in guiding the assessment of instruction data quality.

However, existing evaluation methods for instruction data are biased to the general domain and primarily focus on the textual content of the instructions, resulting in suboptimal performance in the document understanding domain. In the document VQA task, effectively assessing the efficacy of instruction datasets requires a comprehensive understanding and differentiation of the document's content, layout, instructions, and the *instruction execution process*. As illustrated in Figure 3, the data method INSTAG (Lu et al. 2023), which assesses the efficacy of instruction datasets by

the diversity and complexity of instruction text, is applied to DocVQA datasets for training LLMs. The performance of LLMs trained on this filtered dataset exhibits minimal difference compared to models trained on randomly sampled data. The primary issue lies in the fact that, in document understanding, the same instruction text may lead to entirely different data categories depending on the instruction execution processes. Data methods that only focus on the instruction text are incapable of effectively distinguishing these differences. Consider the instruction “What is Tom’s phone number?” as shown in Figure 1. When evaluated using methods like INSTAG, which considers only the text of the instruction, this instruction would be tagged as “information extraction/entity extraction”. In the document understanding scenarios of Figure 1 (a), (b), and (c), it is evident that the same instruction carries three entirely different instruction execution processes. In (a), it is required to locate and understand a paragraph within the document; in (b), the focus shifts to identifying key-value regions before comprehension; and in (c), it involves finding the relevant table region for table understanding. Clearly, the same instruction text can represent entirely different instruction types in different document scenarios. However, there’s a significant lack of research on evaluating the effectiveness of document instruction data, which hinders the development of high-quality datasets for training LLMs/MLLMs in document VQA. Therefore, exploring effective methods for assessing document instruction datasets is crucial.

To this end, we propose ProcTag, a metric for assessing the efficacy of document instruction data by instruction execution process tagging. ProcTag focuses on modeling the process of document instruction execution rather than the instruction text itself. Empowered by the high-performing LLM GPT-3.5, including its chain-of-thought (Wei et al. 2022b; Kojima et al. 2022) reasoning capability, ProcTag introduces a framework that prompts GPT-3.5 to generate the process of document instruction execution. To better represent the process, ProcTag requires GPT-3.5 to express the process in code and then assign tags to the generated process. These tags are used to measure the diversity and complexity, representing the efficacy of the instruction data. Furthermore, to comprehensively represent document content and utilize GPT-3.5 more effectively, we also devise DocLayPrompt, a novel semi-structured prompting strategy based on document layout information for document representation. Compared to existing layout-aware prompts (Lamott et al. 2024), DocLayPrompt outperforms them in capturing the layout information for document representation. Given the relative scarcity of document instruction datasets, we apply the ProcTag method to an existing open-source document VQA dataset and several generated document instruction datasets. After applying ProcTag for data selection, the datasets are used to train both LLMs and MLLMs for experiments. Experimental results demonstrate the effectiveness of the proposed ProcTag method in assessing document instruction data with efficacy when compared to existing data methods and random sampling.

Our contributions are summarized as follows:

1) This paper introduces a framework ProcTag, which mod-

els the instruction execution process instead of the instruction text itself, as a method for assessing the efficacy of document instruction data. To the best of our knowledge, it is the first to explore the instruction execution process as a data quality assessment method for document understanding.

- 2) To effectively model the instruction execution process, ProcTag expresses it for document VQA problems in code and distinguishes the process through tagging. Additionally, a semi-structured layout prompting strategy named DocLayPrompt, which incorporates document layout information for effective document representation, is proposed.
- 3) The proposed ProcTag method, when applied to existing open-source document VQA dataset and generated document understanding instruction datasets, significantly outperforms existing data methods when training LLMs and MLLMs. Additionally, by guiding the generation and filtering of document understanding instruction datasets using ProcTag, only 30.5% of the document instructions are required to achieve 100% efficacy compared to the complete dataset.

Related Works

LLMs/MLLMs for Document VQA. Recently, there have been broad discussions within the community regarding LLMs such as ChatGPT (OpenAI 2022), LLaMA (Touvron et al. 2023a,b) and Qwen (Bai et al. 2023a), and MLLMs like GPT-4V (OpenAI 2023), Gemini (Team et al. 2023) and Qwen-VL (Bai et al. 2023b). These models have achieved considerable success in a wide broad of downstream AI applications. Concurrently, leveraging LLMs/MLLMs for document AI (Yang et al. 2023; Liu et al. 2023c; Perot et al. 2023; Shi et al. 2023), especially document VQA tasks (Zhang et al. 2023; Ye et al. 2023a; Bai et al. 2023b), has demonstrated remarkable performance over previously document pre-trained models (Xu et al. 2020, 2021; Huang et al. 2022; Li, Gu, and Kuen 2021; Luo et al. 2022; Li et al. 2021; Appalaraju, Jasani, and Kota 2021; Gu et al. 2021; Peng et al. 2022; Luo et al. 2023; Xing et al. 2023). LLaVAR (Zhang et al. 2023) enhances the text comprehension capabilities of LLaVA (Liu et al. 2023b) by gathering text-rich images and constructing a corresponding instruction tuning dataset. Moreover, building upon the foundation of mPLUG-Owl (Ye et al. 2023b), mPLUG-DocOwl (Ye et al. 2023a) creates an instruction tuning dataset for various visual-text understanding tasks. Simultaneously, an OCR-free document instruction comprehension evaluation set, LLMDoc, has been developed to better compare models in terms of instruction compliance and document understanding. Likewise, Qwen-VL (Bai et al. 2023b) also considers utilizing high-quality and fine-grained visual-language (VL) instruction datasets to achieve its high-quality multimodal multitask understanding capabilities, which also includes instruction fine-tuning datasets related to documents. Layout-LLM (Luo et al. 2024) proposes a layout instruction tuning method and a document instruction tuning dataset for document understanding. A module called layout chain-of-

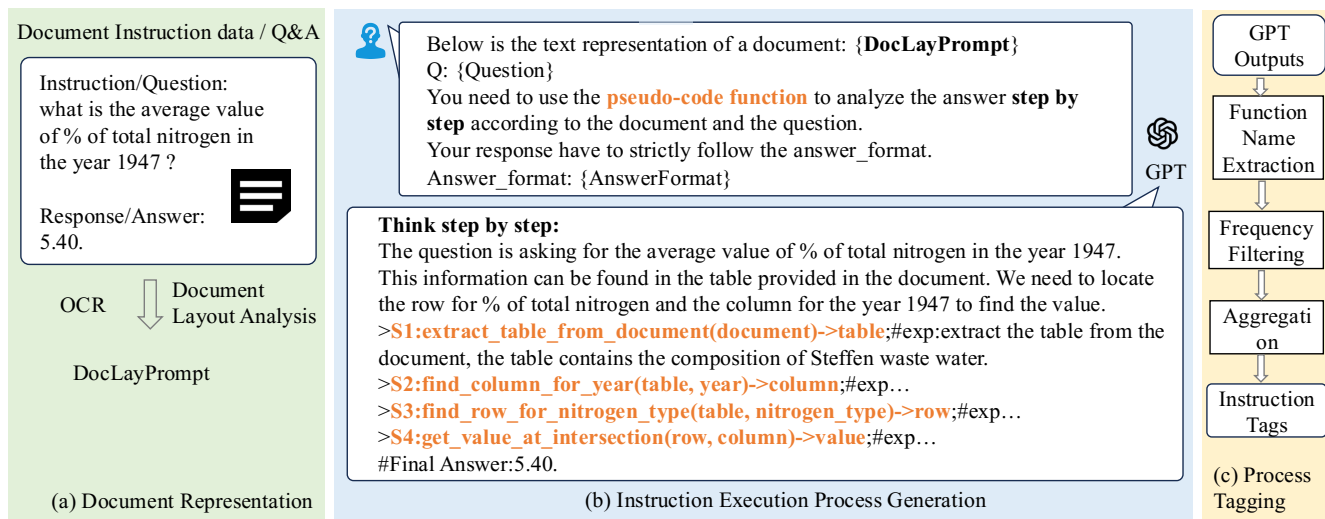


Figure 2: Overview of ProcTag. ProcTag performs tagging on the document instruction execution process for assessing the efficacy of document instruction data, involving three steps: (a) document representation: ProcTag utilizes DocLayPrompt for representing document information; (b) instruction execution process generation: prompting GPT to generate the execution process using pseudo-code; and (c) process tagging: processing the generated pseudo-code to obtain instruction tags.

thought (LayoutCoT) which is effective for document understanding is devised to represent the document-related instruction processes. The success of the aforementioned efforts demonstrates that, for document-related tasks, the document instruction datasets are indispensable.

Instruction Data Methods. Instruction tuning datasets (Wei et al. 2022a; Ouyang et al. 2022) are crucial for calibrating LLMs/MLLMs to align human instructions accurately. The majority of these datasets are sourced by manual annotation or generated by LLMs and MLLMs (Taori et al. 2023; Ding et al. 2023). Recent efforts in the community have recognized the importance of establishing robust instruction dataset evaluation methods to optimize the utility of these datasets (Li et al. 2023; Lu et al. 2023). Li et al. (Li et al. 2023) introduce the Instruction-Following Difficulty (IFD) metric, which autonomously screens for high-quality instruction data by identifying discrepancies between a model’s expected responses and its generative outputs. Despite its innovative approach, this method is hindered by the need for additional training on a pre-experienced model. In contrast, InsTag (Lu et al. 2023) proposes a more cost-efficient alternative that foregoes the need for model re-training. By leveraging existing LLMs to annotate instruction data, InsTag assesses quality along two dimensions: diversity and complexity. This approach also highlights that the quality of instruction datasets is crucial for impacting model performance. For document AI tasks, publicly available document instruction datasets are scarce. Furthermore, the above evaluation metrics are biased towards the general domain and focus primarily on textual instructions. These metrics can not be directly applied to complex multimodal tasks like document VQA, which relies not only on instruction text but also on the understanding of layout and different problem-solving processes. To more effectively employ

document instruction datasets and enhance the document understanding capability of LLMs/MLLMs, it is necessary to develop an evaluation metric suitable for assessing instruction data in document VQA.

ProcTag

Leveraging tags to measure the diversity and complexity for assessing the efficacy of instruction datasets is a very effective way (Lu et al. 2023). As shown in Figure 2, ProcTag tags document instruction datasets by modeling the instruction execution process, including 3 steps.

Document Representation

Effectively representing document content with rich layout information is crucial for tagging document instruction datasets. While directly inputting document images into Multimodal Large Language Models (MLLMs) like GPT-4V is a straightforward approach, it is both costly and inefficient. Research has shown that textual representation alone can achieve satisfactory results in document understanding tasks (Lamott et al. 2024). Methods such as LATIN-Prompt (Wang et al. 2023) and SpatialFormat-prompt (Lamott et al. 2024) reconstruct layout information using Optical Character Recognition (OCR) results, spaces, and line breaks. However, these approaches focus solely on text content, neglecting layout details, which can lead to ambiguities. For instance, it becomes challenging to distinguish between elements like *Page Head* and *Title* when only text is considered, as they often appear at the top of the page with similar fonts.

To address this limitation, we propose a novel semi-structured approach called DocLayPrompt, which incorporates layout information into document representation, enhancing the tagging of document instruction datasets. As

shown in Figure 2 (a), DocLayPrompt supports comprehensive modeling of a document’s content by integrating layout details.

In DocLayPrompt, an input document image \mathcal{D} undergoes OCR and layout detection to extract structural information: OCR results $\mathcal{O} = \langle \mathcal{S}, \mathcal{B}_O \rangle$ and layout results $\mathcal{L} = \langle \mathcal{T}, \mathcal{B}_L \rangle$. Specifically, $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ represents the extracted text sequences, and $\mathcal{B}_O = \{b_{o1}, b_{o2}, \dots, b_{on}\}$ corresponds to their bounding box coordinates. Simultaneously, $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ denotes the identified layout types, with $\mathcal{B}_L = \{b_{l1}, b_{l2}, \dots, b_{ln}\}$ specifying the bounding box coordinates for these layout components. These elements are then utilized to construct DocLayPrompt.

1) **Reorder and Clean Inputs:** To represent real documents accurately, it’s crucial to use clean and organized structural information, such as OCR and layout results. This involves reordering these outcomes using their bounding box coordinates, \mathcal{B}_O and \mathcal{B}_L . Redundant layout instances are removed using Non-Maximum Suppression (NMS), leading to cleaned inputs: \mathcal{O}^C and \mathcal{L}^C .

$$\mathcal{O}^C, \mathcal{L}^C \leftarrow \text{getCleanInputs}(\mathcal{O}, \mathcal{L}). \quad (1)$$

2) **Associate Structural Information:** Both \mathcal{O}^C and \mathcal{L}^C represent the structural information within documents. To create a unified representation, these sources need to be combined into associated structural information, \mathcal{A}_{OL} . Since \mathcal{L}^C covers a broader structural granularity than \mathcal{O}^C , DocLayPrompt integrates a set of OCR results \mathcal{O}_i^C $i = m$ into their corresponding layout component \mathcal{L}_j^C for a tight association. If an OCR result \mathcal{O}_i^C is not encompassed by any layout component, it is associated with the nearest layout component \mathcal{L}_j^C based on Euclidean distance $d(\cdot)$.

$$\mathcal{A}_{OL} \leftarrow \{(\{\mathcal{O}_i^C\}_{i=m}^n, \mathcal{L}_j^C)\}, \text{ where} \\ \{\mathcal{B}_{O_i}\}_{i=m}^n \subseteq \mathcal{B}_{L_j} \text{ or } \arg \min_{i,j} d(\{\mathcal{B}_{O_i}\}_{i=m}^n, \mathcal{B}_{L_j}). \quad (2)$$

3) **Represent Document:** Finally, concatenate \mathcal{A}_{OL} sequentially through LATIN-Prompt and layout type tags to form the document representation \mathcal{R} .

$$\mathcal{R} \leftarrow \text{getDocRep}(\mathcal{A}_{OL}). \quad (3)$$

Instruction Execution Process Generation

As shown in Figure 1, effectively distinguishing document instruction data requires focusing on the instruction execution process rather than just the textual content. Inspired by the chain-of-thought (CoT) (Wei et al. 2022b; Kojima et al. 2022) capabilities of LLMs, ProcTag uses GPT to generate the document instruction execution process. To make this process more precise and concise, ProcTag represents it in pseudo-code. For greater accuracy, GPT first outputs a CoT textual description of the instruction execution, which is then used to generate the corresponding pseudo-code.

As shown in Figure 2 (b), the structural representation \mathcal{R} and corresponding question \mathcal{Q} for the document instruction dataset are employed to prompt GPT to generate instruction execution process content \mathcal{P} . To ensure sufficient and clear

discriminability for subsequent tagging, the generated \mathcal{P} includes a chain-of-thought (CoT) for instruction execution and the corresponding tightly coupled and distinct pseudo-code execution process, where the input of each step is the output of the previous step.

Process Tagging

To obtain distinctive and denoised process tags for assessment, ProcTag normalizes process tags through 3 stages: *Function Name Extraction*, *Frequency Filtering*, and *Aggregation*. As shown in Figure 2 (c), *Function Name Extraction* is utilized to obtain the function names of pseudo-code as the raw process tags from \mathcal{P} . Then, following the InsTag, the raw process tags employ *Frequency Filtering* and *Aggregation* to filter tags that appear too rarely and to aggregate similar tags (e.g., “find_table” and “extract_table”), respectively. To filter out tags that appear infrequently, a threshold is established for their selection. To effectively aggregate semantically similar tags with code-like formats, a code language model is utilized to obtain embeddings for these code-formatted tags, which are then clustered. After that, the final instruction tags obtained through the above stages will be used for subsequent data assessment and experiments.

Efficacy Assessment by Tags

ProcTag defines the efficacy of an instruction dataset through two attributes of its process tags: complexity and diversity. Here, complexity refers to the number of different tags present within the dataset, while diversity denotes the average number of tags per instruction data. Higher levels of complexity and diversity signify a dataset with higher efficacy, whereas lower levels indicate lower efficacy. Following the InsTag data method for data sampling via tags, the ProcTag selects subsets by optimizing for the highest complexity under the condition of maximal tag diversity, thereby yielding sub-datasets with superior efficacy.

Experimental Setup

Datasets

Instruction datasets, including document instruction datasets, are often collected from manual annotations or generated by LLMs or MLLMs. In our experiments, all these types of document instruction datasets are considered for tagging and evaluation.

Manually Annotated Dataset. The widely-used public dataset **DocVQA** (Mathew, Karatzas, and Jawahar 2021) is employed, consisting of 50,000 questions defined across over 12,000 documents from various industry sources.

Generated Dataset. For document VQA tasks, the availability of publicly accessible instruction datasets is limited. Therefore, motivated by existing works that utilize LLMs/MLLMs for generating instruction datasets (Taori et al. 2023; Ding et al. 2023), it is necessary to use LLMs/M-LLMs to generate some expanded corpus of document instruction datasets. Motivated by the LayoutLLM (Luo et al. 2024), to cover a broader spectrum of document types and complexity instructions, four document datasets are used

for building the instruction tuning dataset, including **RVL-CDIP** (Harley, Ufkes, and Derpanis 2015), **DocILE** (Štěpán Šimsa et al. 2023), **PublayNet** (Zhong, Tang, and Yepes 2019), and **PubTabNet** (Zhong, ShafieiBavani, and Jimeno Yepes 2020). RVL-CDIP covers a substantial diversity of document types, with a collection of 400,000 images spanning 16 distinct classes, including, but not limited to, letters, forms, and memos. DocILE is specialized in form-based documents that facilitate diverse information extraction tasks. It contains a set of 6,680 labeled business documents, supplemented by a substantial unlabeled set of 932,000 documents, and a synthetically generated corpus of 100,000 documents. PublayNet, with its focus on document layout analysis, comprises over 360,000 PDF documents rich in textual and layout elements. This dataset features annotations that include common layout components such as text blocks, titles, lists, figures, and tables. Lastly, PubTabNet is characterized by its intricate table structures contained within academic literature. Encompassing more than 568,000 tabular instances in both image and HTML formats, this dataset offers detailed cell bounding box information, which is instrumental for advanced table recognition and understanding tasks.

Implementation Details

ProcTag Implementation Details. (1) *Document Representation.* The Vision Grid Transformer (VGT) (Da et al. 2023) model is utilized as the layout detection tool described in the section Document Representation for structural document representation. The OCR results for document images are sourced from the official OCR outputs provided by each respective document dataset.

(2) *Instruction Execution Process Generation.* During the instruction execution process generation in the section Instruction Execution Process Generation, if the output format returned from GPT does not meet the generation criteria and cannot be parsed, GPT is invoked again. This process is repeated at most twice, and if it still fails, the data is discarded. The proportion of discarded samples does not exceed 0.1%.

(3) *Process Tagging.* In the section Process Tagging, as following InsTag, the tags appearing fewer than 4 times in DocVQA and fewer than 2 times in other datasets are termed as long-tail tags, removing them in “Frequency Filtering”. During the “Aggregation” stage, after the exclusion of lower-case letters and special characters, embeddings for function names are obtained using the SantaCoder (Allal et al. 2023) and clustered with DBSCAN (Hahsler, Piekenbrock, and Doran 2019). The minimum semantic similarity threshold is set to 0.015. Finally, a minimum support of 40 instances and a minimum confidence of 99% are used for association aggregation. It is important to note that for instruction execution processes, the same types of tags with different orders represent different execution processes. Hence, ProcTag aggregates tags only when they exhibit a high frequency of occurrence and are in close proximity to the execution order. For example, “extract_list” and “find_item” are commonly co-occurring, but if they respectively appear in the first and third steps of the execution process that are non-adjacency,

Metric Agreement	GPT-4	Human	Human-GPT
Tag Precision	96%	92%	0.65
Tag Consistency	80%	88%	0.87

Table 1: The quality evaluation of the tags generated by ProcTag. Tagging precision and consistency are utilized for evaluating ProcTag. The Cohen’s kappa score is used to represent the agreement between GPT-4 and human.

they are not considered to be aggregated. Conversely, they would be combined to “extract_list_item”.

Dataset Details. Instruction Generation. PubLayNet, PubTabNet, RVL-CDIP, and DocILE are utilized for instruction generation. For each dataset, 3,000 document images are randomly sampled to prompt GPT (GPT-3.5) with the same prompt that is used for generating the layout instruction tuning dataset of the LayoutLLM (Luo et al. 2024). 10 instruction items are generated for each document. After parsing and filtering, a random selection of 20,000 instances forms the training set, with another 1,000 instances randomly selected to serve as the test set.

Experimental Details. LLMs and MLLMs. Given the superior performance of the Qwen series models on the DocVQA task within open-source models, we choose the Qwen series models as our experimental benchmark. Both LLMs and MLLMs are employed for experimental validation. For LLMs, all experiments are based on the 7B version of Qwen-Chat. For each experiment, the models are trained for 3 epochs with a batch size of 32. As for MLLM models, the 7B version of QwenVL-Chat is employed, trained for 3 epochs, with a batch size of 16. For all experiments, the learning rate is set to 1e-5.

Evaluation Metric. The widely used metric ANLS (Mathew, Karatzas, and Jawahar 2021) is utilized for evaluating DocVQA and the four generated document instruction datasets.

Quality Evaluation

Following the InsTag method, the quality of the tags generated by ProcTag is evaluated from the perspectives of precision and consistency. Precision is defined to measure whether tags correctly relate to the execution processes of their associated instructions, while consistency assesses whether the tags maintain a uniform definition across all corresponding instructions. GPT-4 and manual annotation are utilized to evaluate the tags generated by ProcTag.

Results. Table 1 shows the quality evaluation of the tags generated by ProcTag in the DocVQA dataset. The precision of our ProcTag on GPT-4 and human annotation reaches 96% and 92%, respectively, with consistency both exceeding 80%. Furthermore, Cohen’s kappa score is used to compute the agreement between human and GPT-4 annotation, yielding precision and consistency agreements exceeding 0.6, which qualifies as “substantial agreement”. Moreover, an additional comparison of final answers derived from step-by-step reasoning via ProcTag depicted in Figure 2(b) with ground truth shows an ANLS increase from 73.84 to 78.68

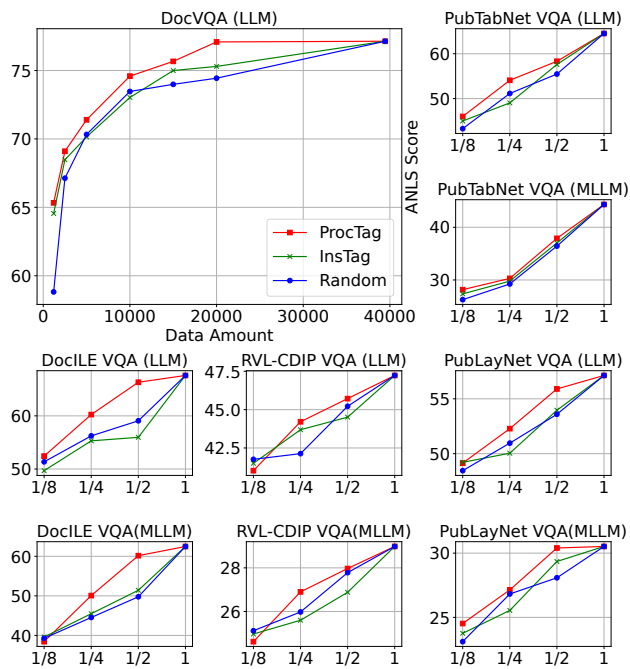


Figure 3: Experimental results of the performance in document VQA after training on LLM (Qwen) and MLLM (Qwen-VL) using datasets sampled with ProcTag, InsTag, and random sampling methods from human-annotated (DocVQA) and generated document instruction datasets.

compared to direct reasoning without ProcTag. In conclusion, the tags marked by ProcTag demonstrate high quality in both precision and consistency.

Experimental Results

Main Results

Our experiments are conducted on the widely-used open-source DocVQA dataset which is manually annotated, as well as on four generated datasets, for supervised fine-tuning of LLMs/MLLMs. For each dataset, models are fine-tuned with different data proportions sampled by three methods: ProcTag, the existing instruction data method InsTag, and random sampling. Then, the document understanding performance of models in these states is evaluated on the test set, where Qwen and Qwen-VL are used as models. As shown in Figure 3, overall, our ProcTag consistently outperforms both InsTag and random variants. Notably, the performance trends of InsTag and random methods are similar. This is because InsTag, which only considers instruction text, cannot effectively distinguish different document instruction data, this finding is consistent with our observations (see Figure 1), which proves the necessity of modeling the process of document instruction execution rather than the instruction text itself.

Evaluation on Manual Annotation Dataset. Due to the scarcity of existing document instruction datasets, Qwen-VL has been trained on most datasets of this category, including

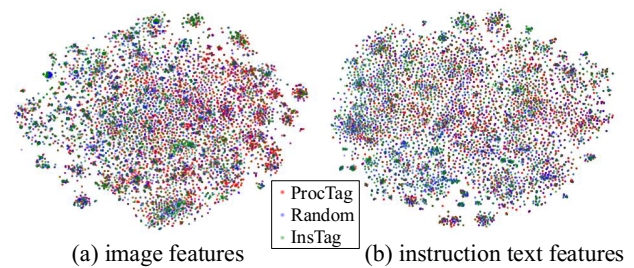


Figure 4: t-SNE visualization of the document image and instruction text features obtained by sampling DocVQA train-set with three methods (ProcTag, Random, and InsTag) using the CLIP feature extractor.

DocVQA. For fair comparison, our experiments avoid validating on Qwen-VL, considering only Qwen.

Evaluation on Generated Dataset. The experimental results corresponding to the four generated datasets show consistent trends on both Qwen and Qwen-VL. In particular, unstable performance has been observed in the setting with 1/8 data proportion on certain datasets (like DocILE and RVL-CDIP), which we hypothesize is caused by limited data amount. Moreover, it can be seen that the performance achieves the highest on DocILE, attributable to its highly diverse document content. Conversely, ProcTag performs the least effectively on PubTabNet and PubLayNet, which are confined to tabular and academic paper layouts, respectively. The performance on RVL-CDIP is among them. In summary, our ProcTag exhibits superior performance on datasets with more diverse document data.

Visualization of the original document data features sampled by different methods. In addition, to evaluate the feasibility of direct feature extraction from raw instructional document data to sample effective instances, the widely acknowledged feature extractor CLIP (Radford et al. 2021) is utilized to extract features from images and corresponding instructional texts within document datasets gathered by various sampling methods. Subsequently, these features are visualized using t-SNE (Arora, Hu, and Kothari 2018). As depicted in Figure 4, the raw document instruction features originating from diverse sample sources could not be effectively differentiated, despite observable performance variances amongst them. This indicates that for document instruction data, utilizing raw data features does not afford effective discrimination. Hence, it is imperative to employ specialized document representations like modeling the process of document instruction execution for effective modeling rather than relying on raw data.

ProcTag for Data Efficacy

Existing instruction datasets employed for LLMs/MLLMs are often sourced from generated content, characterized by significant size and redundancy, which can lead to extended training durations and resource inefficiency. To address this challenge, data efficacy is introduced, denoted as E , to serve as a criterion for selecting high-quality data. This efficacy is defined by the ratio of current performance P_{cur} to the best

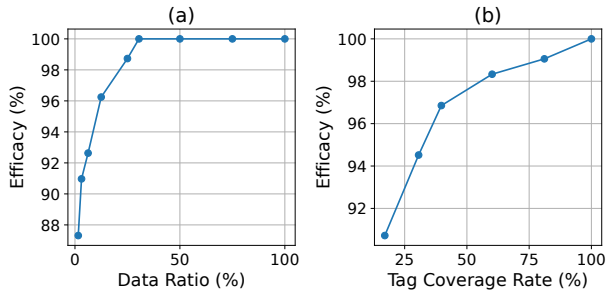


Figure 5: Experimental analysis results of data efficacy in terms of data amount ratio and tag coverage rate.

Verbalizer	GPT-3.5 (direct prompting)	Qwen (fine-tuning)
PlainText	63.66	77.14
SpatialFormat	71.03	80.93
DocLayPrompt	73.84	81.80

Table 2: Ablation study on the impact of different prompts for representing documents in the DocVQA task.

performance P_{best} , given by $E = P_{cur}/P_{best}$. To assess data efficacy, we conduct experiments based on data ratio and tag coverage rate. Within these two settings, P_{best} refers to the performance achieved at 100% data ratio and tag coverage rate, respectively.

As shown in Figure 5, when the data ratio is varied, the peak performance is observed at 30%, indicating that our ProcTag can maintain high effectiveness with only a small subset of data. Furthermore, in terms of coverage rate, which represents the percentage of all tags covered in the dataset. Experimental results show that as coverage increases, performance exhibits a positive correlation with improvement. The results confirm the effectiveness of all tags generated by ProcTag, indicating that each tag contributes to the enhancement of performance. These results validate that our ProcTag for tagging is rational and effective, while also ensuring the diversity and complexity of these tags.

Ablation study

Impact of DocLayPrompt To evaluate the effectiveness of DocLayPrompt in document text representation, we conducted comparative experiments with two commonly used prompts: PlainText and SpecialFormat (Lamott et al. 2024). In direct prompting tests on the DocVQA test set using GPT-3.5, as shown in Table 2, DocLayPrompt improved ANLS by 2.81 compared to SpecialFormat (Lamott et al. 2024). Additionally, fine-tuning experiments with Qwen using various document prompts showed that DocLayPrompt outperformed both PlainText and SpecialFormat, demonstrating its superiority in document representation, whether for direct prompting or LLM training.

Impact of different representations of the process. Though a straightforward process representation using natural language is recognized as a practical alternative, pseudo-

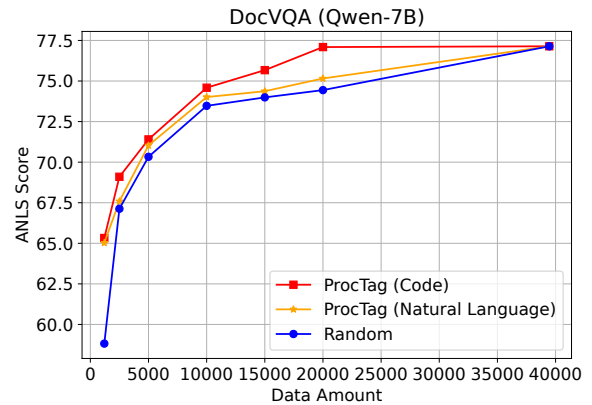


Figure 6: The impact of representing instruction execution processes with pseudo-code and natural language.

code is utilized to represent the instruction execution process in ProcTag. The preference for pseudo-code stems from its ability to provide a more granular and structured representation, which is anticipated to yield a more detailed and succinct explanation of the instruction execution process. Thus, it is likely to enhance the effectiveness of the data sampling process. To evaluate the effectiveness of pseudo-code representation, an analysis is conducted with process tags generated in natural language and assessed using the DocVQA dataset. As shown in Figure 6, the experimental results show that while natural language tags offer some advantages over random sampling, they do not provide the significant improvements observed with pseudo-code tags, confirming the superior efficacy of the pseudo-code format.

Limitation

Due to cost constraints, the current approach employs GPT-3.5 and a text-based document prompt, rather than incorporating multimodal large language models like GPT-4V. Clearly, relying on textual document representation results in the loss of certain visual information, thereby hindering the applicability of our approach on visually-rich datasets such as InfographicVQA (Mathew et al. 2022).

Conclusion

In this paper, we propose ProcTag, a data method that assesses the efficacy of document instruction data. ProcTag performs tagging on the execution process of document instructions and utilizes the diversity and complexity of these tags to assess the efficacy of the dataset. Additionally, DocLayPrompt, a novel semi-structured layout-aware document prompt, is proposed for effectively representing documents. Experimental results demonstrate the effectiveness of the ProcTag method in assessing document instruction data with efficacy when compared to existing data methods and random sampling. As modeling the process of instruction execution is a generic approach, in the future, we will extend this approach to the general artificial intelligence domain, exploring more effective data evaluation strategies.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62372408)

References

- Allal, L. B.; Li, R.; Kocetkov, D.; Mou, C.; Akiki, C.; Ferlandis, C. M.; Muennighoff, N.; Mishra, M.; Gu, A.; Dey, M.; et al. 2023. SantaCoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*.
- Appalaraju, S.; Jasani, B.; and Kota, B. U. 2021. DocFormer: End-to-End Transformer for Document Understanding. In *ICCV*, 4171–4186.
- Arora, S.; Hu, W.; and Kothari, P. K. 2018. An analysis of the t-sne algorithm for data visualization. In *Conference on learning theory*, 1455–1462. PMLR.
- Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023b. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *NeurIPS*, 33: 1877–1901.
- Cui, L.; Xu, Y.; Lv, T.; and Wei, F. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.
- Da, C.; Luo, C.; Zheng, Q.; and Yao, C. 2023. Vision grid transformer for document layout analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19462–19472.
- Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Zheng, Z.; Hu, S.; Liu, Z.; Sun, M.; and Zhou, B. 2023. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. *arXiv preprint arXiv:2305.14233*.
- Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Barmpalios, N.; Jain, R.; Nenkova, A.; and Sun, T. 2021. Unified Pretraining Framework for Document Understanding. In *NeurIPS*.
- Hahsler, M.; Piekenbrock, M.; and Doran, D. 2019. dbSCAN: Fast density-based clustering with R. *Journal of Statistical Software*, 91: 1–30.
- Han, S. C.; Ding, Y.; Luo, S.; Poon, J.; Yoon, H.; Huang, Z.; Duuring, P.; and Holden, E. J. 2023. Workshop on Document Intelligence Understanding. *arXiv:2307.16369*.
- Harley, A. W.; Ufkes, A.; and Derpanis, K. G. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *ICDAR*, 991–995. IEEE.
- 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 Multimedia*.
- Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *NeurIPS*, 35: 22199–22213.
- Lamott, M.; Weweler, Y.-N.; Ulges, A.; Shafait, F.; Krechel, D.; and Obradovic, D. 2024. LAPDoc: Layout-Aware Prompting for Documents. *arXiv preprint arXiv:2402.09841*.
- Li, M.; Zhang, Y.; Li, Z.; Chen, J.; Chen, L.; Cheng, N.; Wang, J.; Zhou, T.; and Xiao, J. 2023. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.
- Li, P.; Gu, J.; and Kuen, J. 2021. SelfDoc: Self-Supervised Document Representation Learning. In *CVPR*, 5652–5660.
- Li, Y.; Qian, Y.; Yu, Y.; Qin, X.; Zhang, C.; Liu, Y.; Yao, K.; Han, J.; Liu, J.; and Ding, E. 2021. StrucTexT: Structured text understanding with multi-modal Transformers. In *ACM Multimedia*, 1912–1920.
- Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2023a. Improved Baselines with Visual Instruction Tuning. *arXiv preprint arXiv:2310.03744*.
- Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023b. Visual Instruction Tuning.
- Liu, Y.; Li, Z.; Li, H.; Yu, W.; Liu, Y.; Yang, B.; Huang, M.; Peng, D.; Liu, M.; Chen, M.; Li, C.; Yin, X.; Lin, L.; and Bai, X. 2023c. On the Hidden Mystery of OCR in Large Multimodal Models. *arXiv:2305.07895*.
- Lu, K.; Yuan, H.; Yuan, Z.; Lin, R.; Lin, J.; Tan, C.; Zhou, C.; and Zhou, J. 2023. # InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models. In *The Twelfth International Conference on Learning Representations*.
- Luo, C.; Cheng, C.; Zheng, Q.; and Yao, C. 2023. Geo-LayoutLM: Geometric Pre-Training for Visual Information Extraction. In *CVPR*, 7092–7101.
- Luo, C.; Shen, Y.; Zhu, Z.; Zheng, Q.; Yu, Z.; and Yao, C. 2024. LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding. In *CVPR*.
- Luo, C.; Tang, G.; Zheng, Q.; Yao, C.; Jin, L.; Li, C.; Xue, Y.; and Si, L. 2022. Bi-VLDoc: Bidirectional Vision-Language Modeling for Visually-Rich Document Understanding. *arXiv preprint arXiv:2206.13155*.
- Mathew, M.; Bagal, V.; Tito, R.; Karatzas, D.; Valveny, E.; and Jawahar, C. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1697–1706.
- Mathew, M.; Karatzas, D.; and Jawahar, C. 2021. Docvqa: A dataset for vqa on document images. In *WACV*, 2200–2209.
- Mukherjee, S.; Mitra, A.; Jawahar, G.; Agarwal, S.; Palangi, H.; and Awadallah, A. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.
- OpenAI. 2022. Introducing chatgpt. <https://openai.com/blog/chatgpt>.
- OpenAI, R. 2023. GPT-4 technical report. *arXiv*, 2303–08774.
- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.;

- et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS*, 35: 27730–27744.
- Peng, Q.; Pan, Y.; Wang, W.; Luo, B.; Zhang, Z.; Huang, Z.; Hu, T.; 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 *EMNLP findings*.
- Perot, V.; Kang, K.; Luisier, F.; Su, G.; Sun, X.; Boppana, R. S.; Wang, Z.; Mu, J.; Zhang, H.; and Hua, N. 2023. LMDX: Language Model-based Document Information Extraction and Localization. *arXiv preprint arXiv:2309.10952*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8748–8763.
- Shi, Y.; Peng, D.; Liao, W.; Lin, Z.; Chen, X.; Liu, C.; Zhang, Y.; and Jin, L. 2023. Exploring ocr capabilities of gpt-4v (ision): A quantitative and in-depth evaluation. *arXiv preprint arXiv:2310.16809*.
- Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.
- Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023a. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardaş, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv:2307.09288*.
- Wang, W.; Li, Y.; Ou, Y.; and Zhang, Y. 2023. Layout and Task Aware Instruction Prompt for Zero-shot Document Image Question Answering. *arXiv preprint arXiv:2306.00526*.
- Wang, Y.; Ivison, H.; Dasigi, P.; Hessel, J.; Khot, T.; Chandu, K.; Wadden, D.; MacMillan, K.; Smith, N. A.; Beltagy, I.; et al. 2024. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36.
- Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned language models are zero-shot learners. *ICLR*.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35: 24824–24837.
- Xing, H.; Gao, F.; Long, R.; Bu, J.; Zheng, Q.; Li, L.; Yao, C.; and Yu, Z. 2023. LORE: logical location regression network for table structure recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2992–3000.
- 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 preprint arXiv:2304.12244*.
- Xu, Y.; Li, M.; Cui, L.; and Huang, S. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *KDD*, 1192–1200.
- 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. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In *ACL*.
- Yang, Z.; Li, L.; Lin, K.; Wang, J.; Lin, C.-C.; Liu, Z.; and Wang, L. 2023. The dawn of Imms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9.
- Ye, J.; Hu, A.; Xu, H.; Ye, Q.; Yan, M.; Dan, Y.; Zhao, C.; Xu, G.; Li, C.; Tian, J.; et al. 2023a. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*.
- Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; Li, C.; Xu, Y.; Chen, H.; Tian, J.; Qi, Q.; Zhang, J.; and Huang, F. 2023b. mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality. *arXiv:2304.14178*.
- Zhang, Y.; Zhang, R.; Gu, J.; Zhou, Y.; Lipka, N.; Yang, D.; and Sun, T. 2023. Llavav: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*.
- Zhong, X.; ShafieiBavani, E.; and Jimeno Yepes, A. 2020. Image-based table recognition: data, model, and evaluation. In *ECCV*, 564–580. Springer.
- Zhong, X.; Tang, J.; and Yepes, A. J. 2019. Publaynet: largest dataset ever for document layout analysis. In *ICDAR*, 1015–1022. IEEE.
- Štěpán Šimsa; Šulc, M.; Uříčář, M.; Patel, Y.; Hamdi, A.; Kocián, M.; Skalický, M.; Matas, J.; Doucet, A.; Coustaty, M.; and Karatzas, D. 2023. DocILE Benchmark for Document Information Localization and Extraction. *arXiv:2302.05658*.