ARIAL: An Agentic Framework for Document VQA with Precise Answer Localization

Ahmad Mohammadshirazi
Ohio State University
Flairsoft
Columbus, Ohio, US
mohammadshirazi.2@osu.edu

Pinaki Prasad Guha Neogi
Ohio State University
Columbus, Ohio, US
guhaneogi.2@osu.edu

Dheeraj Kulshrestha
Flairsoft
Columbus, Ohio, US
dheeraj@flairsoft.net

Rajiv Ramnath
Ohio State University
Columbus, Ohio, US
ramnath.6@osu.edu

Abstract

Document Visual Question Answering requires models to understand text layouts and ground answers to specific document regions. However, existing systems prioritize textual accuracy while neglecting spatial grounding, limiting interpretability in high-stakes applications. We present **ARIAL** (Agentic Reasoning for Interpretable Answer Localization), a modular framework using LLMs planning agent to orchestrate specialized components for OCR, retrieval-augmented generation, and spatial grounding. ARIAL decomposes Document VQA into structured tool calls: TrOCR-based text extraction, semantic retrieval over OCR segments, LLMs for answer generation, and precise bounding-box localization. This modular design enables transparent reasoning traces for auditability. We evaluate on four benchmarks (DocVQA, FUNSD, CORD, SROIE) using text similarity (ANLS) and spatial metrics (mAP@IoU). ARIAL achieves new state-of-the-art results, including 88.7 ANLS and 50.1 mAP on DocVQA, surpassing DLaVA by +2.8 ANLS and +3.9 mAP points. ARIAL focuses on spatially-grounded language models, demonstrating how LLMs can be constrained through modular tool orchestration where each answer is locked to specific pixel coordinates and traceable through interpretable reasoning chains. The coding implementation can be found in: https://github.com/ahmad-shirazi/ARIAL

1 Introduction

Document Visual Question Answering (VQA) requires reasoning over both textual content and visual layout in scanned or digitally rendered documents. Models must not only read and understand diverse formats—forms, receipts, reports—but also locate where answers appear within the document structure.

While recent models such as LayoutLMv3 [9], LayoutLLM [18], and DocLayLLM [16] have improved textual accuracy by combining language with layout features, they often treat localization as a secondary task. Consequently, they may generate plausible answers without clearly identifying their source in the document, making verification difficult. Standard metrics like ANLS [27] capture string similarity but fail to reflect spatial correctness, prompting a shift towards combined evaluations that include IoU for grounding precision.

DLaVA [22] introduced answer localization by integrating bounding-box prediction within a large multimodal transformer. However, its monolithic design can be computationally intensive and may struggle with fine-grained details in dense or handwritten layouts.

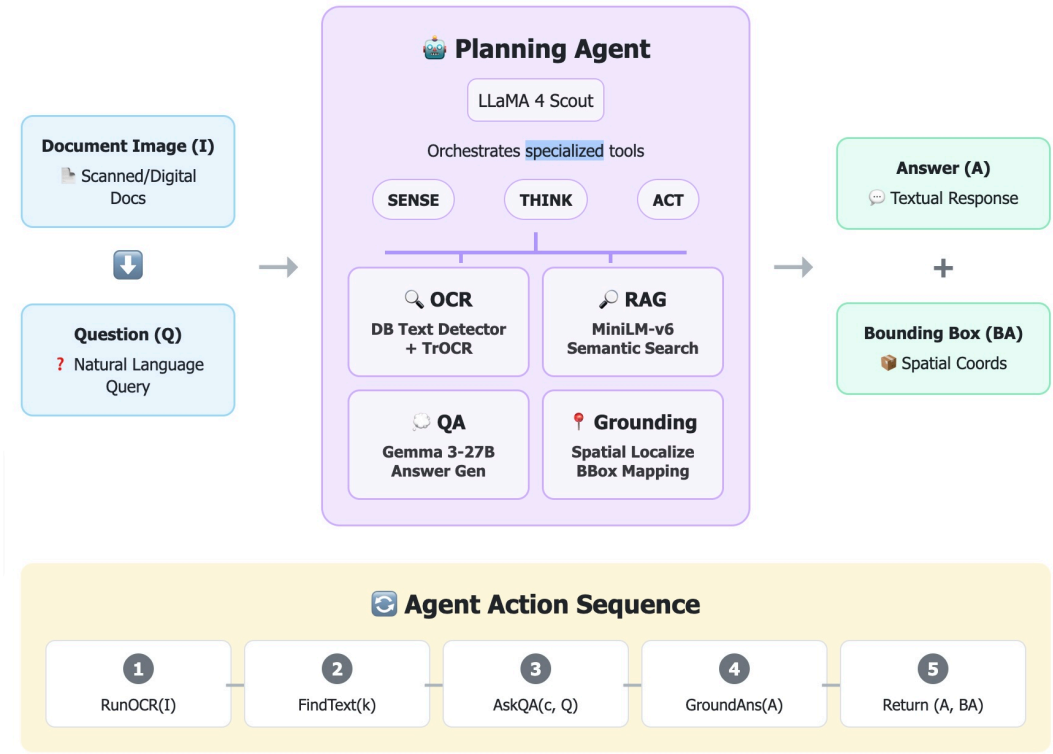

Figure 1: Overview of the ARIAL agentic workflow for Document VQA. The system consists of three modular stages: (1) Input Processing, where an OCR module extracts text segments and bounding boxes from a document image; (2) Agentic Reasoning Pipeline, where the planner agent coordinates task execution—retrieving relevant text, invoking QA or computation, and triggering spatial grounding; and (3) Output Generation, where the final answer and its bounding box are produced. The reasoning loop enables iterative refinement based on confidence, supporting flexible and context-aware decision-making.

We propose **ARIAL** (Agentic Reasoning for Interpretable Answer Localization), a modular document VQA framework built around an agentic planning model. Rather than using a single large model, ARIAL delegates subtasks—OCR, layout analysis, retrieval, reasoning, and grounding—to specialized modules orchestrated by a central agent. This agent, implemented with LLaMA 4 Scout [20], dynamically selects tools and composes multi-step reasoning chains, enabling accurate and interpretable answers with precise spatial grounding. Our key contributions are:

1. **Agentic Document QA**: We introduce an agent-based document VQA system that decomposes queries into tool calls for OCR, retrieval, and grounding. The modular design enables tool reuse, error tracing, and flexible adaptation across document types.
2. **Precise Answer Localization**: ARIAL produces both answer text and corresponding bounding boxes by aligning answers to OCR-detected spans and contextual cues, ensuring visual traceability.
3. **Retrieval-Augmented Reasoning**: ARIAL incorporates retrieval-augmented generation [14] to focus on relevant text segments, enhancing both reasoning accuracy and efficiency for long or noisy documents.
4. **State-of-the-Art Results**: On four benchmarks—DocVQA [19], FUNSD [12], CORD [23], and SROIE [10]—ARIAL achieves new best results in both ANLS and mAP@IoU, reaching 88.7 ANLS and 50.1 mAP on DocVQA.

ARIAL demonstrates how LLMs can be effectively constrained through modular tool orchestration, where each answer is locked to specific pixel coordinates and traceable through interpretable reasoning

chains. This addresses fundamental challenges in developing trustworthy, location-aware AI systems for document understanding.

The remainder of this paper is organized as follows: Section 2 reviews related work in document VQA and agentic AI. Section 3 details ARIAL's architecture and modules. Section 4 outlines datasets and evaluation protocols. Results and analysis appear in Section 5, followed by discussion in Section 6 and conclusions in Section 7.

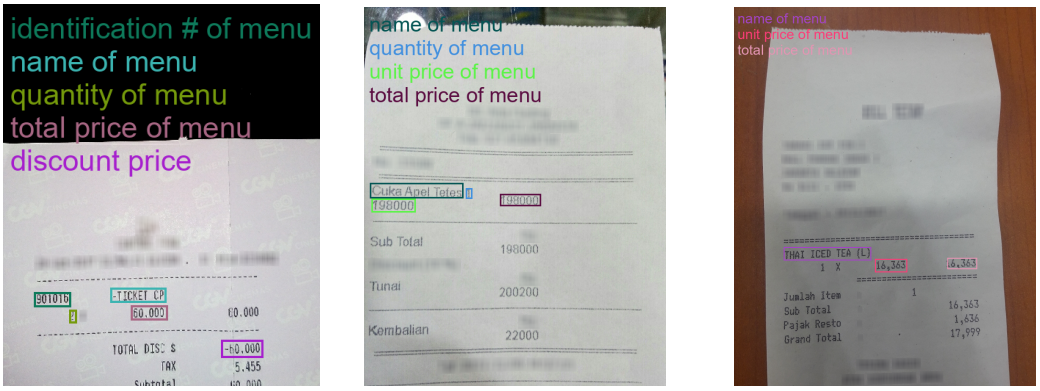

Figure 2: Illustrative examples of visual information extraction on receipt images from the CORD dataset [23]. Each colored annotation corresponds to its extracted answer, highlighted by a matching colored bounding box.

2 Related Work

2.1 Document VQA and Layout-Aware Models

Early document QA systems treated the task as text-only reading comprehension by applying OCR and feeding results into standard NLP models [21]. However, such approaches ignored document structure, prompting the development of layout-aware models. LayoutLM [26], LayoutLMv3 [9], DocFormer [1], and StrucTexT [15] embed both text and spatial coordinates to model document layouts more effectively, achieving strong performance on datasets like DocVQA through unified transformer architectures.

Nevertheless, most models output only answer text and treat localization as auxiliary prediction or post-hoc mapping. Methods like TILT [24] and Donut [13] explore end-to-end generation—Donut bypasses explicit OCR—but lack transparent mechanisms for spatial grounding. As highlighted by DLaVA [22], the inability to visualize answer provenance limits model interpretability and hinders error analysis in high-trust domains.

2.2 Multimodal LLMs for Documents

Multimodal large language models (MLLMs) such as GPT-4o [11], Gemini 2.5 Pro [5], and LLaVA 1.5 [17] extend VQA capabilities by jointly modeling vision and language. These systems answer questions directly from document images using prompt-based interfaces but often function as black boxes, lacking explicit reasoning steps and failing to highlight the visual basis of their answers. Their reliance on global visual understanding can lead to errors in fine-grained text recognition and spatial disambiguation [2].

Recent domain-specific adaptations like LayoutLLM [18] augment prompts with structured spatial cues to guide model focus. DLaVA [22] combines detected text with bounding box metadata or constructed text images, enabling prediction of both answer and spatial location. While DLaVA improves interpretability, it relies on a large, end-to-end multimodal backbone. Our method adopts a modular agentic design enabling more transparent and controllable reasoning while retaining compatibility with any OCR or LLM module.

Table 1: Performance comparison on Document VQA datasets. ANLS measures textual accuracy; mAP@IoU captures spatial localization quality.

Method	DocVQA		FUNSD		CORD		SROIE	
	ANLS	mAP@IoU	ANLS	mAP@IoU	ANLS	mAP@IoU	ANLS	mAP@IoU
DocLayLLM (Llama3-7B)	78.4	-	84.1	-	71.3	-	84.3	-
LayoutLLM (Vicuna-1.5-7B)	74.3	-	80.0	-	63.1	-	72.1	-
DLaVA (OCR-Dep)	74.0	34.9	79.6	32.0	82.1	48.0	91.4	-
DLaVA (OCR-Free)	85.9	46.2	87.6	45.5	84.4	57.9	91.4	-
ARIAL (Ours)	**88.7**	**50.1**	**90.0**	**50.3**	**85.5**	**60.2**	**93.1**	**-**

2.3 Agent-Based and Modular Reasoning

Agentic frameworks have emerged as powerful alternatives to monolithic models for complex tasks [7]. Systems like HuggingGPT [25] use a central language model to coordinate multiple tools for multi-step reasoning. Multi-agent paradigms have been explored for general VQA [28], where specialized agents handle subtasks like reading, counting, or visual interpretation. HAMMR [3] introduces hierarchical architecture improving reasoning granularity and debuggability.

In the document domain, MDocAgent [6] employs multiple agents for long-document QA with roles spanning retrieval, modality-specific analysis, key information extraction, and summarization. This modular approach demonstrated notable performance gains, showing the potential of agentic decomposition.

ARIAL builds upon these foundations by tailoring an agentic framework for document VQA. Unlike generic VQA agents, ARIAL handles document-specific challenges such as dense typography, noisy scans, and form-based structures. Its modularity allows independent component upgrades, facilitating efficient domain adaptation and improving interpretability. ARIAL advances document understanding by combining the reasoning power of MLLMs with the transparency and controllability of agentic pipelines, enabling precise answer localization and robust performance across diverse document types.

3 Methodology

3.1 Overview

ARIAL is a modular framework employing a reasoning agent to orchestrate specialized tools for accurate answer generation and precise spatial grounding. The central component is a Planner Agent instantiated by LLaMA 4 Scout, which interprets queries and dynamically routes them through OCR, retrieval, QA, and grounding modules following a sense-think-act paradigm.

Given a document image I and question Q, the system returns answer A and bounding box B_A. The agent constructs a sequence of actions $\{a_1, a_2, \ldots, a_n\}$, where each a_i is either a tool call (`RunOCR(I)`, `FindText(keywords)`, `AskQA(context, Q)`, `GroundAnswer(answer)`) or an internal reasoning step guiding tool selection. This sequence adapts dynamically to query complexity, terminating when the agent produces a confident answer with visual grounding.

Table 2: Ablation Study (DocVQA and FUNSD)

Model Variant	DocVQA ANLS	DocVQA mAP@IoU	FUNSD ANLS	FUNSD mAP@IoU
Full ARIAL (Agent + RAG + GenQA)	88.7	50.1	90.0	50.3
– No Retrieval (all text to QA)	86.2	48.5	88.1	47.9
– Heuristic Agent (no LLM planning)	83.6	44.2	85.4	42.8
– No Generative QA (lookup only)	87.0	49.0	89.0	49.5

3.2 OCR and Layout Parsing

We employ a two-stage OCR pipeline using DB text detector with ResNet-50 backbone for text region identification, followed by TrOCR for recognition. This yields OCR results $\{(T_i, B_i)\}_{i=1}^{N}$, where T_i is recognized text and B_i is the corresponding bounding box. Standard preprocessing includes resolution scaling, grayscale conversion, noise removal, and de-skewing. The OCR module maintains reading order and optionally groups segments into structured units using layout heuristics.

3.3 Retrieval-Augmented Generation

The agent performs both lexical and semantic search over OCR segments $\{T_i\}$ using `FindText(keywords)`. Text segments are encoded using MiniLM-v6 Sentence Transformer, with question Q similarly encoded. Retrieved segments $\{(T_j, B_j)\}$ with highest cosine similarity and keyword matches are passed to the QA module. The agent invokes `AskQA(Context, Q)` using Gemma 3-27B [4], which generates answers from retrieved context, reducing hallucination compared to processing entire documents.

For computational queries, the agent identifies relevant numeric fields and invokes `Compute(sum, values)` operations. When no relevant segments are found, the system outputs "No answer found" to avoid unsupported responses.

3.4 Spatial Grounding

After QA generates answer A, the agent invokes `GroundAnswer(A)` to localize the answer. For exact matches to OCR segment T_k, we use bounding box B_k. For multi-segment answers, we merge involved boxes into unified region B_A. For computed answers, the module highlights supporting evidence. Ambiguous answers are disambiguated using contextual cues from retrieved segments and question keywords.

3.5 Training and Fine-Tuning

ARIAL's modular design enables independent component optimization. OCR uses pretrained DB detector and TrOCR without additional fine-tuning. Retrieval employs off-the-shelf MiniLM-v6 embeddings. The QA module fine-tunes Gemma 3-27B on 70k document QA pairs from DocVQA, CORD, and FUNSD training sets. The Planner Agent uses LLaMA 4 Scout fine-tuned via behavioral cloning on 50 demonstration traces showing appropriate tool usage patterns.

Table 3: End-to-End vs. Agentic Approach Comparison

Metric	LayoutLLM	DocLayLLM	DLaVA OCR-Free	ARIAL (Agentic)
DocVQA ANLS	74.3	78.4	85.9	**88.7**
DocVQA mAP@IoU	–	–	46.2	**50.1**
Average Latency (s/q)	0.7	0.4	1.2	3.2
Interpretability	No	No	Yes	Yes + reasoning trace

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate ARIAL on four benchmarks: DocVQA [19], FUNSD [12], CORD [23], and SROIE [8]. We report text accuracy via ANLS (Average Normalized Levenshtein Similarity [27], 0–100%) and localization precision via mean Average Precision over IoU thresholds 0.50–0.95 (mAP@0.50:0.95).

4.2 Baselines and Comparisons

We compare ARIAL with layout-aware and LLM-based models: **DocLayLLM** [16] and **LayoutLLM** [18]: LLM-based models using LLaMA/Vicuna backbones with layout-aware prompts and spatial tokens. **DLaVA** [22]: Vision-language model with two modes: (1) OCR-Dependent using

detected text with spatial metadata, and (2) OCR-Free synthesizing visual text patches for implicit text handling.

4.3 Implementation Details

Agent: LLaMA 4 Scout fine-tuned on tool usage traces with 5 few-shot examples for chain-of-thought prompting. Retrieval limited to top-5 segments (DocVQA) and top-3 (FUNSD, CORD, SROIE). **OCR:** DB text detector with ResNet-50 backbone and Microsoft TrOCR for recognition, operating at 2 seconds per page. **QA Module:** Gemma 3-27B fine-tuned for 3 epochs on 70k document QA pairs using Adam optimizer (lr=1e-4). **Infrastructure:** 4× NVIDIA H100 80GB GPUs with LLaMA 4 agent and Gemma 3-27B on separate GPUs.

5 Results

5.1 Overall Performance

Table 1 shows ARIAL consistently achieves state-of-the-art results on both text accuracy (ANLS) and spatial grounding (mAP@IoU) across all datasets. On DocVQA, ARIAL attains 88.7 ANLS and 50.1 mAP, outperforming DLaVA by +2.8 ANLS and +3.9 mAP points. These gains highlight the benefit of ARIAL's modular reasoning and fine-grained retrieval over integrated transformer approaches.

5.2 Comparison with Baselines

ARIAL consistently outperforms both encoder-only and LLM-based methods. LayoutLLM and DocLayLLM achieve 74.3% and 78.4% ANLS on DocVQA with no localization capability. DLaVA's OCR-Free mode reaches 85.9% ANLS and 46.2% mAP. ARIAL's agentic pipeline outperforms all baselines by significant margins, demonstrating the effectiveness of explicit tool orchestration.

5.3 Ablation Study

Table 2 quantifies each component's contribution: **No Retrieval**: Feeding entire OCR text to QA causes -2.5 ANLS drop on DocVQA, confirming that targeted context retrieval prevents confusion from irrelevant text. **Heuristic Agent**: Replacing intelligent planning with fixed pipeline drops performance substantially (-5.0 ANLS on DocVQA), highlighting the value of adaptive reasoning. **No Generative QA**: Restricting to string matching degrades ANLS by -1.7 on DocVQA, showing the generator's importance for complex questions requiring text understanding.

6 Discussion

ARIAL's modular design demonstrates clear advantages over monolithic models through consistent ANLS and mAP gains. The explicit tool orchestration enables both higher textual accuracy (+2.8–4.4 pp) and improved spatial precision (+3.9–4.8 pp) compared to DLaVA. Unlike encoder-only approaches lacking spatial outputs, ARIAL produces per-answer bounding boxes with transparent audit traces through its tool sequence. The modular architecture allows independent component upgrades, facilitating domain adaptation and improving interpretability. While ARIAL incurs higher latency (≈ 3.2 s/query) compared to monolithic models (0.4–1.2 s), this cost provides crucial trustworthiness through explicit grounding and reasoning transparency essential for high-stakes applications.

7 Conclusion

We introduced ARIAL, an agentic framework for Document VQA emphasizing accurate answer extraction and explicit spatial grounding. By orchestrating OCR, retrieval, and QA through a modular planning agent, ARIAL achieves SoTA performance on DocVQA, FUNSD, CORD, and SROIE, surpassing prior methods in both textual accuracy and localization precision. ARIAL's modular pipeline enables transparent reasoning steps, tool-level auditability, and adaptability to diverse document types—capabilities lacking in monolithic models. This makes ARIAL particularly suited for high-stakes settings requiring answer traceability. Our work demonstrates the potential of

agent-driven AI for document understanding by merging LLM reasoning capabilities with specialized vision and OCR tools, producing a system that delivers SoTA performance while meeting real-world demands for trustworthy, explainable AI.

References

[1] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1003, 2021.

[2] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

[3] Lluis Castrejon, Thomas Mensink, Howard Zhou, Vittorio Ferrari, Andre Araujo, and Jasper Uijlings. Hammr: Hierarchical multimodal react agents for generic vqa. *arXiv preprint arXiv:2404.05465*, 2024.

[4] Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL `https://arxiv.org/abs/2503.19786`.

[5] Google Cloud. Gemini 2.5 pro, June 2025. URL `https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro`. Last updated 2025-06-27 UTC.

[6] Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*, 2025.

[7] Kadhim Hayawi and Sakib Shahriar. Ai agents from copilots to coworkers: Historical context, challenges, limitations, implications, and practical guidelines. *Preprints*, 10, 2024.

[8] Wen Huang, Minghui Qiao, Cong Bai, Yulin Yong, Sheng Zhang, and Qun Guo. Sroie: Scanned receipt ocr and information extraction. In *Proceedings of the ICDAR 2019 Competition on Scanned Receipts OCR and Information Extraction*, 2019.

[9] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

[10] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019.

[11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[12] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019.

[13] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022.

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33. 2020.

[15] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers. In *Proceedings of the 29th ACM international conference on multimedia*, pages 1912–1920, 2021.

[16] Wenhui Liao, Jiapeng Wang, Hongliang Li, Chengyu Wang, Jun Huang, and Lianwen Jin. Doclayllm: An efficient multi-modal extension of large language models for text-rich document understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4038–4049, 2025.

[17] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[18] Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. Layoutllm: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15630–15640, 2024.

[19] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[20] Meta. Llama 4 Scout, 2025. URL `https://www.llama.com/docs/get-started/`. Large language model, version released April 5, 2025.

[21] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

[22] Ahmad Mohammadshirazi, Pinaki Prasad Guha Neogi, Ser-Nam Lim, and Rajiv Ramnath. Dlava: Document language and vision assistant for answer localization with enhanced interpretability and trustworthiness. *arXiv preprint arXiv:2412.00151*, 2024.

[23] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[24] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 732–747. Springer, 2021.

[25] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

[26] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1192–1200, 2020. doi: 10.1145/3394486.3403172.

[27] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

[28] Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, et al. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *ACM Computing Surveys*, 57(8): 1–39, 2025.