OCR or Not? Rethinking Document Information Extraction in the MLLMs Era with Real-World Large-Scale Datasets

Jiyuan Shen[1], Peiyue Yuan[1], Atin Ghosh[1], Yifan Mai[2], Daniel Dahlmeier[1]
[1]SAP [2]Stanford University
{jiyuan.shen,peiyue.yuan,atin.ghosh,d.dahlmeier}@sap.com yifan@cs.stanford.edu

Abstract

Multimodal Large Language Models (MLLMs) enhance the potential of natural language processing. However, their actual impact on document information extraction remains unclear. In particular, it is unclear whether an MLLM-only pipeline—while simpler—can truly match the performance of traditional OCR+MLLM setups. In this paper, we conduct a large-scale benchmarking study that evaluates various out-of-the-box MLLMs on business-document information extraction. To examine and explore failure modes, we propose an automated hierarchical error analysis framework that leverages large language models (LLMs) to diagnose error patterns systematically. Our findings suggest that OCR may not be necessary for powerful MLLMs, as image-only input can achieve comparable performance to OCR-enhanced approaches. Moreover, we demonstrate that carefully designed schema, exemplars, and instructions can further enhance MLLMs performance. We hope this work can offer practical guidance and valuable insight for advancing document information extraction.

1 Introduction

Within the field of natural language processing (NLP), a key application involves automatically extracting key information from various sources, such as invoices, insurance quotes, and financial statements, and turning it into structured information. This capability is used in various industries, which help businesses automate and streamline document-based and scene-text workflows, improving operational efficiency (Gartner).

However, the vast majority of mature document information extraction systems in the industry still rely on a two-stage framework, where optical character recognition (OCR) first extracts textual content before a secondary specialized model converts the text into structured information following a schema (Wang et al., 2023). This approach, while effective, is inherently complex, difficult to generalize to new domains and susceptible to error propagation from OCR to downstream extraction. These limitations have motivated growing interest in OCR-free and few-shot learning approaches (Kim et al., 2022; Ye et al., 2023; Liu et al., 2024; MistralAI). The rapid advancement of general-purpose MLLMs further strengthens this trend, as many are pretrained on large-scale structured document and should, in principle, possess strong information extraction capabilities (Team et al., 2024; Intelligence, 2024). Yet their true effectiveness in this area remains highly unclear.

Therefore, we evaluate a range of state-of-the-art MLLMs on a large-scale, high-quality benchmark dataset, which reflects our experience in developing enterprise document AI services. Specifically, we experiment with three different input modalities: OCR-extracted text only, raw document images only, and a combination of both.

Furthermore, we leverage large language models (LLMs) capabilities to develop an automated error analysis framework that systematically categorizes prediction errors through a hierarchical reasoning approach. By analyzing failure cases and benchmarking results, we provide deeper insights into critical questions, such as *Is OCR necessary for MLLM-based document information extraction? Can MLLMs serve as a promising path for streamlining the pipeline?* Through this study, our objective is to bridge the gap between academic research and real-world applications, shedding light on the strengths and limitations of advanced approaches in document information extraction.

The main contributions of this work are summarized as follows:

1. We investigate the role of OCR in document information extraction with MLLMs and find that for specific powerful models, OCR may not be necessary and can even have a slightly

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 385–396
March 25-27, 2026

negative impact. Our findings suggest that MLLM-only pipeline is a promising direction for document information extraction.

2. We demonstrate that as MLLMs increase in size, their information extraction performance can still improve accordingly.

3. We propose a hierarchical error analysis framework that can automatically discover the error patterns.

4. We find that general-purpose MLLMs lack task-specific knowledge, highlighting the need for more carefully designed schemas, exemplars, and instructions. We refine our approach and achieve measurable performance improvement by leveraging insights from our error analysis framework.

2 Related Work

Although using domain-specific OCR models together with task-tuned extraction models is widely regarded as good practice in industry (Katti et al., 2018; Huang et al., 2022), the drawbacks are easy to recognize: system complexity, limited generalization, and substantial labor required to adapt pipelines to new domains. These limitations have motivated the research community to explore more streamlined end-to-end approaches, even at the cost of a slight performance trade-off (Ouyang et al., 2025). The rapid advancement of MLLMs has further accelerated this shift (MistralAI; Bai et al., 2023). These powerful models are pretrained on large-scale, diverse image datasets and subsequently refined through instruction tuning, enabling strong visual understanding, layout awareness, and zero-shot reasoning. For example, GPT-4o (Hurst et al., 2024) and Gemini (Team et al., 2023) exhibit impressive capabilities in jointly interpreting visual layouts and textual content, offering a promising balance between accuracy and efficiency. However, a comprehensive benchmark of MLLMs for business-document information extraction is still lacking. To address this gap, we aim to provide a rigorous evaluation and a fair comparison of their effectiveness in real-world scenarios.

3 Methodology

3.1 Internal Industrial Document Dataset

Our internal datasets encompass a diverse range of documents, with dataset C1 sourced from the supply chain domain and C2 from finance. For all of these documents, we collected manual annotations with carefully curated structured ground-truth labels, along with OCR-extracted text results. We use our in-house OCR engine that has been developed for business documents and achieves high performance with an average accuracy of more than 90% in multiple languages. In our internal evaluations, it outperforms state-of-the-art OCR methods and OCR services provided by major machine learning platforms. Figure 1 shows a sample document page, and Figure 2 illustrates the textual content extracted by our OCR engine. As demonstrated, we preserve layout information by retaining whitespace as a structural delimiter in the extracted text.

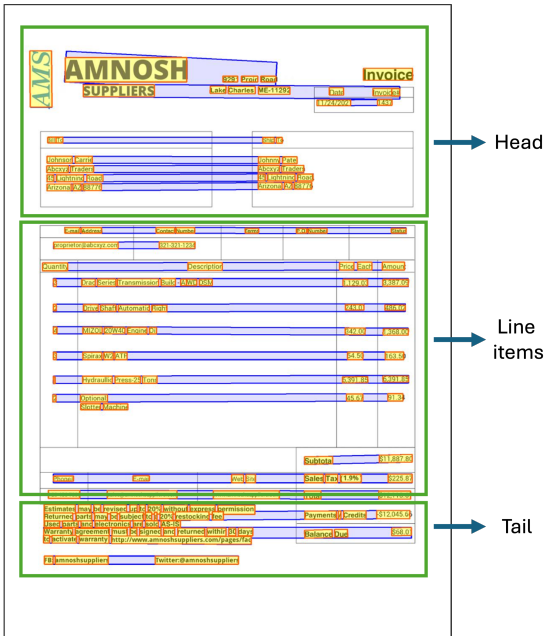

Figure 1: Example of a document page extracted using our OCR engine.

Compared with existing open-source datasets, ours is substantially more challenging. The difficulties stem primarily from two sources: (i) multilingual content and (ii) structural complexity. Regarding multilinguality, we provide comprehensive statistics analysis in Appendix B that reflect the wide language distribution across multiple countries and multi-page documents. Regarding structural complexity, our dataset contains nested information, stacked cells within line items, and heterogeneous header structures—factors that significantly increase the difficulty of document parsing. Refer to Figure 1 for an example.

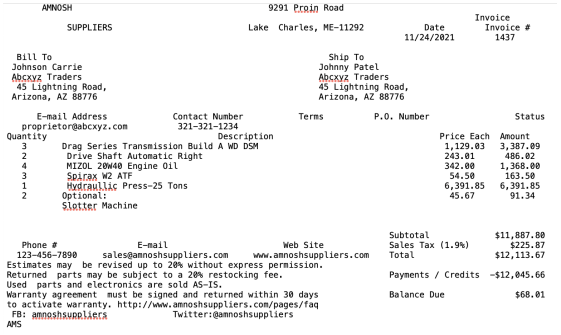
AMNOSH 9291 Proin Road
SUPPLIERS Lake Charles, ME-11292
Invoice
Date 11/24/2021
Invoice # 1437
Bill To
Johnson Carrie
Abcxyz Traders
45 Lightning Road,
Arizona, AZ 88776
Ship To
Johnny Patel
Abcxyz Traders
45 Lightning Road,
Arizona, AZ 88776
E-mail Address: proprietor@abcxyz.com
Contact Number: 321-321-1234
Terms
P.O. Number
Status
Quantity | Description | Price Each | Amount
3 Drag Series Transmission Build A WD DSM 1,129.03 3,387.09
2 Drive Shaft Automatic Right 243.01 486.02
4 MIZOL 20W40 Engine Oil 342.00 1,368.00
3 Spirax W2 ATF 54.50 163.50
1 Hydraulic Press-25 Tons 6,391.85 6,391.85
2 Optional: Slotter Machine 45.67 91.34
Subtotal $11,887.80
Sales Tax (1.9%) $225.87
Total $12,113.67
Payments / Credits -$12,045.66
Balance Due $68.01
Phone # 123-456-7890
E-mail sales@amnoshsuppliers.com
Web Site www.amnoshsuppliers.com
Estimates may be revised up to 20% without express permission.
Returned parts may be subject to a 20% restocking fee.
Used parts and electronics are sold AS-IS.
Warranty agreement must be signed and returned within 30 days to activate warranty. http://www.amnoshsuppliers.com/pages/faq
FB: amnoshsuppliers
Twitter:@amnoshsuppliers
AMS

Figure 2: An example of textual content extracted by our in-house OCR engine.

3.2 Evaluation Pipeline and Metrics

We have incorporated some of the principles of VHELM's design and utilize wrapped clients (Lee et al., 2024). Our evaluation pipeline consists of three main stages. The first stage involves using an OCR engine to extract textual content from document images, preserving the positional information. For image-only experiments, the OCR step is skipped.

The second stage focuses on structured information extraction. For MLLM-based approaches, we construct a prompt template (see Appendix A for details) that includes format instructions and the document schema, enabling zero-shot information extraction. The target extraction schema consists of *header fields* and a list of *line items*, which capture structured tabular information. The MLLM output is a JSON object, where keys represent entity types, and values correspond to extracted content from the document. An example of response is shown in Appendix A.

In the final stage, we report the overall performance using the standard F1 score. Specifically, since our outputs are structured as key–value pairs, we compute precision and recall over all key–value predictions, and then derive the F1 score from these metrics.

3.3 Hierarchical Error Analysis Framework

To systematically diagnose errors in document information extraction, we adopt a hierarchical error analysis framework inspired by Chen et al. (2024). Our framework categorizes errors from the middle to the highest level, following a logical progression from direct observations to deeper root causes. This structured approach ensures that errors are first identified based on surface-level discrepancies and then further analyzed to uncover underlying reasons. We show our framework in Figure 3.

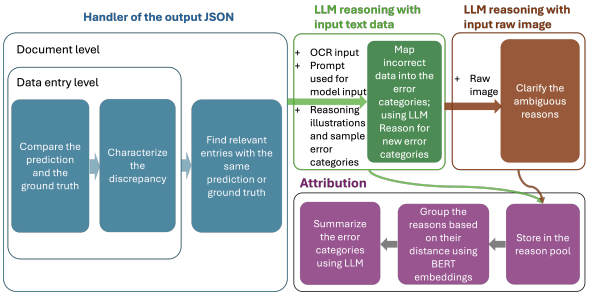

Figure 3: Hierarchical Error Analysis Framework

3.3.1 Handler

The error analysis process begins with an automated error handler that systematically logs and classifies prediction mismatches. Given a set of extracted predictions and ground-truth values, we compare them at both character and semantic levels, ensuring robust error identification. The analysis is performed at both the field level and document level. The process consists of three main steps: (1) comparing the predicted values with the ground truth, (2) characterizing the discrepancies between them, and (3) identifying relevant entries with similar predictions or ground-truth values for further analysis.

3.3.2 LLM Reasoning

To refine the classification of errors and the root cause analysis, we use LLM-based reasoning. Instead of manually analyzing failure cases, we employ LLMs and MLLMs to help generate structured diagnostic reports.

The hierarchical reasoning process consists of two steps: (1) mapping incorrect predictions into predefined error categories using LLMs, which also allows for identifying new error categories when necessary, and (2) clarifying ambiguous errors by incorporating raw document images as additional input for reasoning. The first step utilizes textual input from OCR results, predicted values, and ground-truth labels, along with predefined reasoning templates and few-shot cause-of-failure examples to categorize errors and generate potential causes. In cases where textual reasoning alone is insufficient, such as errors arising from layout complexities or visual ambiguities, we introduce raw document images to refine error attribution. This approach ensures a more comprehensive understanding of extraction failures. By the end of this stage, all errors are categorized into mid-level

Table 1: Performance comparison of different MLLMs across evaluation settings: Image, OCR, and Image + OCR as input formats. C1 and C2 refer to two different datasets, while Mean denotes the arithmetic mean of the F1-scores on C1 and C2.

Company	Model	Image-only			OCR-only			Image + OCR		
		Dataset C1	Dataset C2	Mean	Dataset C1	Dataset C2	Mean	Dataset C1	Dataset C2	Mean
Meta	Llama 4 Scout	67.4	69.3	68.4	68.1	69.7	68.9	67.3	69.8	68.6
	Llama 4 Maverick	62.8	68.2	65.5	63.9	68.1	66.0	62.9	68.2	65.5
MistralAI	Pixtral Large (2411)	68.7	57.4	63.1	75.3	71.2	73.3	72.7	68.0	70.4
Amazon	Nova Pro	77.9	65.1	71.5	68.7	65.1	66.9	77.5	66.6	72.1
OpenAI	GPT-4o mini	68.3	64.9	66.6	66.1	70.5	68.3	71.6	70.5	71.1
	GPT-4o	75.5	68.9	70.1	76.0	69.5	72.8	76.7	69.3	73.0
Anthropic	Claude 3 Opus	43.8	56.4	50.1	72.0	68.2	70.1	74.0	69.1	71.5
	Claude 3.5 Sonnet	65.0	69.3	67.2	73.7	**72.6**	72.8	73.6	69.6	71.6
Google	Gemini 1.5 Pro	**87.3**	66.4	**76.8**	**78.4**	69.8	**74.1**	**86.2**	65.0	**75.6**
	Gemini 2.0 Pro	75.2	**73.3**	74.3	77.6	69.5	73.6	77.1	**73.2**	75.2
	Gemini 2.5 Flash	73.9	71.2	72.6	74.6	69.6	72.1	73.0	71.4	72.2

error reasons, which form a structured foundation for deeper analysis in subsequent attribution steps.

3.3.3 Attribution

The final stage of our framework involves attributing errors to specific highest-level failure sources. Post-processing is performed on the LLM-generated explanations to summarize the error categories. First, the categorized reasons are stored in a structured reason pool. Next, we apply BERT-based embedding clustering to group similar reasons based on cosine similarity, ensuring a coherent categorization of error types. Finally, we extract representative keywords for each error type within the same cluster. We analyze the behavior of the model across multiple documents to determine whether errors originate from OCR misrecognition, layout misinterpretation, prompt misalignment, model capability issues, or schema inconsistencies.

4 Experiments

4.1 Baselines

We evaluate each MLLM using three input formats: document image-only, OCR-extracted text, and a combination of both. Our experiments focus on flagship models from major providers, limited to those released after 2024 to reflect state-of-the-art capabilities. Gemini 2.5 Flash is used in place of the Gemini 2.5 Pro due to Gemini 2.5 Pro's high inference latency. Although current open-source models are generally still underperform in comparison to proprietary models, we add Llama 4 for a comprehensive benchmarking.

4.2 Experiment Results

Table 1 presents a comparative analysis of various MLLMs under three input settings: image-only, OCR-only, and image + OCR. Model performance is evaluated using the F1-score on two business document datasets—C1 (from the supply chain domain) and C2 (from the finance domain)—with the arithmetic mean used as the overall metric.

Models that accept OCR-only input consistently achieve mean F1-scores in the range of 66% to 74%, exhibiting relatively low variance across the board. In contrast, image-only inputs result in a wider performance spread, highlighting larger disparities among models from different providers. Notably, when OCR and image inputs are combined, the variance in mean performance decreases, with scores falling within a narrower range of 70% to 75%. This suggests that incorporating image input can help models produce more stable and robust predictions.

A row-level comparison with the OCR-only setting further reveals that models such as Nova Pro, GPT-4o, and the Gemini series benefit from multimodal input, which shows improvements of 1–3 percentage points in F1-score. However, exceptions do exist. For example, models like Pixtral and Claude 3.5 Sonnet exhibit decreased performance when image input is added. We hypothesize that these models may struggle to effectively integrate visual information with their text processing components, leading to suboptimal fusion of multimodal features.

4.3 Analysis

4.3.1 Is OCR necessary for MLLM-based document information extraction?

From Table 1, we observe an interesting phenomenon when analyzing the flagship Gemini and Nova models. Unlike several other models, the performance of these two model series does not significantly degrade when using image-only input, without OCR-extracted text. In some cases, they even exhibit notable improvements. While this was initially considered a potential anomaly, the trend remained consistent across multiple re-evaluations with varied sampling strategies. This implies that certain advanced multimodal models are capable of directly extracting structured information from document images and comprehending textual content effectively, without the need for OCR as an intermediary. In particular, for the Gemini models, OCR-generated text appears to provide little to no additional benefit. We provide more explanation in Appendix C.

4.3.2 Does MLLMs performance scale with model size across different input modalities?

It is well established that larger models will perform better (Kaplan et al., 2020). However, does this trend persist within our internal dataset when using different input modalities for MLLMs? Specifically, as shown in Figure 4, the overall performance improves as the size of the model increases[1]. Among the three input types, the most significant performance gain is observed with OCR-only input, where the score increases from 57% to 74%. In contrast, the performance of image-only and multimodal inputs remains relatively comparable. The potential reason is that even the Gemini 1.5 Flash is already capable of a rather high baseline performance score.

A particularly interesting observation is that for the Gemini 2.0 Flash-Lite model, the image-only input outperforms the multimodal input by nearly 3%. This result suggests that OCR-extracted text does not necessarily provide a significant performance boost. Instead, even the powerful, yet small model can extract and understand textual information directly from images without relying on explicit OCR input. Furthermore, the variance in performance across modalities suggests that different

[1]Google does not disclose the exact parameter sizes for each variant, but the size relationship can be partially inferred from the model naming.

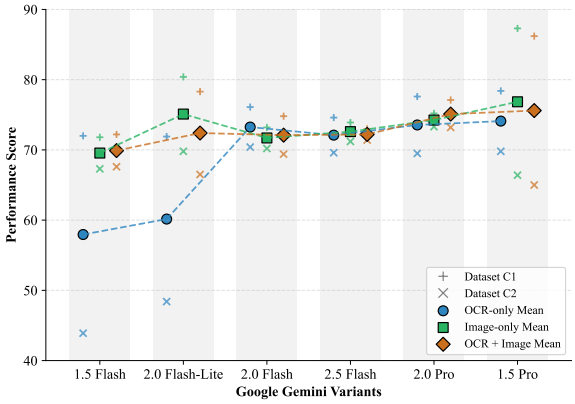

Figure 4: Performance comparison on various size models across different input types. The small shape (●, ■, ◆) denotes the arithmetic mean across two different categories of dataset. + is the F1-score in C1, while × is for C2.

model sizes exhibit varying levels of dependence on OCR-extracted text. Meanwhile, interestingly, for open-source MLLMs such as the Llama 4 series, we observe a negative correlation between model size and multimodal performance gains. This may stem from differences in training corpus scale—for instance, the smaller Scout model is trained on 40T tokens, whereas the larger Maverick model uses only 22T tokens—potentially limiting the larger model's OCR robustness and cross-modal alignment.

Taken together, these findings offer new insights into MLLM scaling behavior and highlight the substantial potential of vision encoders to handle textual information effectively, especially when using genuinely high-capacity MLLMs.

4.3.3 Computational cost and inference latency

Since most of the models we benchmark are closed-source, we report the average cost and inference latency by directly consuming these endpoint. From Table 2, we observe that both speed and cost continue to improve over time. Additionally, MLLMs offer a key advantage — their strong generalization capability. They can adapt more easily to new document types and languages without requiring extensive task-specific fine-tuning. This brings us back to our core motivation: MLLMs hold significant potential to streamline the entire document processing pipeline while maintaining strong performance in information retrieval tasks.

Table 2: Estimated latency and cost per page for different closed-source models.

Model	Latency/Page (Est.)	Cost/Page (Est.)
GPT-4o	∼2.2s	∼$0.006
Claude 3.5 Sonnet	∼4.7s	∼$0.010
Claude 3 Opus	∼7.0s	∼$0.050
Gemini 1.5 Pro	∼3.9s	∼$0.001
Gemini 2.0 Pro	∼2.0s	∼$0.004
Gemini 2.5 Flash	∼1.4s	∼$0.0025
Pixtral Large	∼7.0s	∼$0.0035
Amazon Nova Pro	∼6.6s	∼$0.004

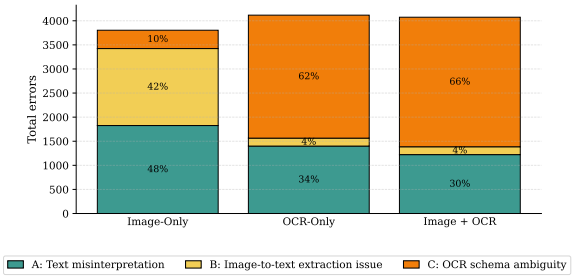

Figure 5: Error analysis results for three different input modalities.

5 Discussion

We employ our hierarchical error analysis framework to categorize the underlying causes of errors. Figure 5 presents the results, and representative failure cases for each category are detailed in Appendix D. At a high level, the image-only input yields the lowest total error count, followed by the combined input, while the OCR-only input exhibits the highest error rate. We categorize errors into three main types: text misinterpretation (Error **A**), which involves challenges in aligning extracted information with the structured information; image-to-text extraction issues (Error **B**), which assess how well MLLMs understand textual content from images; and OCR schema ambiguity issues (Error **C**), which stem from inaccuracies in text recognition and confusion in document schema description.

We observe that image-to-text extraction errors are relatively high for the image-only setting but lower when OCR is included. This is expected, as our OCR system provides high transcription accuracy, whereas raw MLLMs may naturally introduce text-recognition errors. However, schema-ambiguity errors are notably reduced with image-only input. A likely explanation is that the built-in vision encoder integrates more effectively with the text encoder–decoder and captures page layout and document structure more faithfully, resulting in fewer overall mistakes. Nonetheless, there remains substantial room for improvement. Motivated by these, we apply several enhancements to further improve performance:

Google Gemini 1.5 Pro	**Initial**	**Final**
Dataset C1	87.3	**89.1**
Dataset C2	66.4	**68.6**
Mean	76.8	**78.9**

Table 3: Performance results for the optimized prompt template with image-only input.

- Prompt Optimization: Introducing explicit emphasis and reasoning cues to encourage a more thoughtful generation.
- Format Refinement: Strengthening format constraints to reduce output inconsistencies.
- Schema Adjustment: Clarifying schema descriptions to minimize ambiguity.

Using these improvements, we performed a follow-up comparison experiment using a refined prompt template (details in Appendix E) for the input of only images. As shown in Table 3, the results show a further boost in performance, with the mean score increasing from 76.8% to 78.9%, which surpasses both the OCR-only and combined inputs. This promising result further validates the feasibility and effectiveness of the image-only approach in document information extraction.

6 Conclusion

In summary, we conducted a comprehensive benchmarking study on two internal document information extraction datasets, evaluating three distinct input modalities: OCR-only, image-only, and image+OCR. In addition, we perform an automatic error analysis in failure cases. Our findings reveal that powerful MLLMs can achieve competitive performance with image-only input, suggesting that OCR is not necessary in some cases. Furthermore, our automated error analysis helps developers identify common error patterns. Based on these, we demonstrate how well-designed schemas, exemplars, and instructions can further improve MLLM performance. We believe that these findings offer valuable insight to advance research in document information extraction.

Limitations

Despite the promising results, our current approach has several limitations. First, we did not systematically validate the effectiveness of few-shot learning. Second, incorporating chain-of-thought (CoT) or a self-reflection mechanism could potentially further improve model performance, but this was not explored in our current setup due to the resource constraint. Finally, our error analysis framework could further benefit from enhanced reasoning capabilities by integrating a reasoning model, such as O1 (Jaech et al., 2024) or DeepSeek R1 (Guo et al., 2025). Exploring the use of such reasoning-centric models represents a direction for future work.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, et al. 2024. Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pages 674–688.

Gartner. Intelligent document processing solutions reviews and ratings. https://www.gartner.com/reviews/market/intelligent-document-processing-solutions.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. 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.

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

Amazon Artificial General Intelligence. 2024. The amazon nova family of models: Technical report and model card.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. *arXiv preprint arXiv:1809.08799*.

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*, pages 498–517. Springer.

Tony Lee, Haoqin Tu, Chi Heem Wong, Wenhao Zheng, Yiyang Zhou, Yifan Mai, Josselin Roberts, Michihiro Yasunaga, Huaxiu Yao, Cihang Xie, and Percy Liang. 2024. VHELM: A holistic evaluation of vision language models. *Advances in Neural Information Processing Systems*, 37:140632–140666.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024. TextMonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

MistralAI. Mistral ocr technique report. https://mistral.ai/news/mistral-ocr.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, et al. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023. VRDU: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. 2023. mPLUG-DocOwl: Modularized multimodal large language

model for document understanding. *arXiv preprint arXiv:2307.02499*.

A Details in Evaluation Pipeline

We use the following prompt template in our original evaluation pipeline:

Prompt Template:

You are a warehouse manager receiving a delivery. As an expert, you go through the attached delivery note and carefully extract the data that you require to receive the shipped goods and process them in your ERP system. So it is important to focus on the actually received goods and quantities.

The document may be in English, German or any other language. Some of the fields that you need may be indicated by abbreviations in the language of the document. It is important that you carefully extract the information and that you only retrieve information actually on the document. If you have any doubts on a field, skip the field.

Instructions: {format instructions}.
{document schema}.

Return date fields in YYYY-MM-DD format.
For country and currency use ISO format.
Do not include the schema in the answer.
Return missing values as empty string.
Always return valid json and don't wrap you response in backticks!
Do not include a comma before the closing curly bracket.

Here is the document: {OCR extracted content}

Here is the image:

The response format is like below:

Response Example:

```
{
"deliveryDate": [""],
"deliveryNoteNumber": ["ID"],
"documentDate": ["YYYY-MM-DD"],
"purchaseOrderNumber": [""],
"supplierId": [""],
"lineItems": [
{
"lineItem.customerMaterialNumber": "",
"lineItem.itemNumber": "1",
"lineItem.purchaseOrderItemNumber": "",
"lineItem.purchaseOrderNumber": "",
"lineItem.quantity": "QUANTITY",
"lineItem.supplierMaterialNumber": "MATERIAL CODE",
"lineItem.unitOfMeasure": ""
},
...
]
}
```

B Dataset Statistics

A summary of our dataset statistics is provided in Table 4:

Dataset	Approx. Doc Count	Avg. Word Density	Page Language Distribution	Document Currencies	Document Countries
C1 + C2	Around 1,000	High density (financial tabular + semi-structured text, ∼150–400 words per page)	English (∼200), Spanish (∼150), French (∼100), Italian (∼80), German (∼90), Romanian (∼20), Slovak (∼10), Hungarian (∼10), Portuguese (∼10), Mixed/Unknown (∼150), Other (<10 each)	Euro (∼70), Indian Rupee (∼70), US Dollar (∼50), British Pound (∼30), Generic/Masked (∼500), Chinese Yuan (∼10), UAE Dirham (∼10), Indonesian Rupiah (∼10), Swiss Franc (∼10), Vietnamese Dong (∼5), Malaysian Ringgit (∼5), Saudi Riyal (∼5), Venezuelan Bolívar (∼5), Australian Dollar (∼5), Philippine Peso (∼5), South Korean Won (<5), South African Rand (<5), Singapore Dollar (<5), Moroccan Dirham (<5), New Zealand Dollar (<5), Bolivian Boliviano (<5), Canadian Dollar (<5), Azerbaijani Manat (<5), Turkish Lira (<5), Hungarian Forint (<5), Danish Krone (<5), Null/Unspecified (∼20)	Spain (∼120), Romania (∼80), France (∼80), Italy (∼80), Germany (∼90), India (∼70), Netherlands (∼70), US (∼40), UK (∼30), UAE (∼10), China (∼10), Indonesia (∼10), Venezuela (∼10), Saudi Arabia (∼10), Ireland (∼10), Austria (∼5), Switzerland (∼5), Denmark (∼5), Slovakia (∼5), Vietnam (∼5), Malaysia (∼5), Portugal (∼5), Hungary (∼5), Australia (∼5), Philippines (∼5), South Korea (<5), South Africa (<5), Singapore (<5), Morocco (<5), Peru (<5), New Zealand (<5), Bolivia (<5), Canada (<5), Azerbaijan (<5), Turkey (<5), Null/Unspecified (∼10)

Table 4: Dataset characteristics: document counts, density, language distribution, currencies, and country coverage.

C Why some MLLMs perform even better with only image as input?

Empirically, high-capacity models (e.g., Gemini variants, Nova Pro) often match or even surpass multimodal inputs when given image-only inputs. We identify two main drivers behind this pattern.

First, at the mechanistic level, web-scale pretraining equips these MLLMs with strong implicit OCR: visual tokenizers and 2D attention layers can recover glyphs, reading order, and layout hierarchies directly from images. This preserves typographic and spatial cues that external OCR systems may distort or lose. Consistent with this, our error analysis shows that OCR-only inputs are dominated by schema-ambiguity errors, whereas image-only inputs yield fewer total errors.

Second, scaling amplifies these advantages. As model capacity increases and instruction tuning improves, MLLMs internalize increasingly robust text recognition and layout-aware reasoning. This narrows—and occasionally reverses—the expected multimodal advantage. For example, as shown in Figure 4, Gemini 2.0 Flash-Lite's image-only configuration slightly surpasses its image+OCR setting.

Figure 6: The corresponding image(cropped and censored) for example 1,2 and 3.

D Failure Case Study

D.1 Text misinterpretation

Example 1

For the data entry "lineItem.itemNumber", the ground truth specifies the item number as "2" while the prediction erroneously records it as "002". The cause analysis indicates that this mistake is likely from a misreading or misunderstanding of the given text format. The item number as shown in Figure 6 is "002" confirms the correct OCR extraction. This suggests that the error is due to omission in the interpretation of the format guideline.

Example 1:

Data entry: `"lineItem.itemNumber"`
Ground truth: ["2"]
Prediction: `"002"`
Cause: *"Error due to misreading or misunderstanding the text format"*

Example 2:

Data entry: `"lineItem.quantity"`
Ground truth: ["8.00]
Prediction: `"1"`
Cause: *"Error due to incorrect quantity extraction"*

Example 2

For the data entry "lineItem.quantity", the ground truth specifies that the quantity should be "8.00", but the prediction inaccurately records it as "1". It is reasoned that this discrepancy arises from an error in the extraction process, where the quantity is incorrectly interpreted or extracted. The model does not capture "8.00Pcs" from the table in Figure 6 and correctly identifies it as the quantity attribute, suggesting a text misinterpretation problem.

Example 3

Following Example 2, the model fails to identify "Pcs" in "8.00 Pcs" as the unit of measure. Instead, the prediction is "Im". This error implies a misinterpretation of abbreviations during the data extraction process.

Example 3:

Data entry: `"lineItem.unitOfMeasure"`
Ground truth: ["Pcs"]
Prediction: `"Im"`
Cause: *"Error due to misinterpretation of abbreviations"*

D.2 Image-to-text extraction issue

Example 4

Regarding with the data entry "lineItem.supplierMaterialNumber", the ground truth specifies "KL-840I" whereas the prediction is "KL-8401". The cause analysis suggests that the error arises from visual similarity between the character "I" and the digit "1" in the document image, as shown in Figure 7. As the model performs direct image-to-text extraction without explicit OCR segmentation, it misinterpreted the final character due to font style, resolution, or noise, replacing the uppercase "I" with the numeral "1".

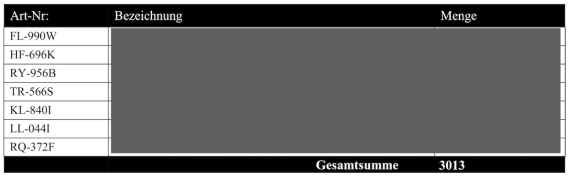

Figure 7: The corresponding image(cropped and censored) for example 4.

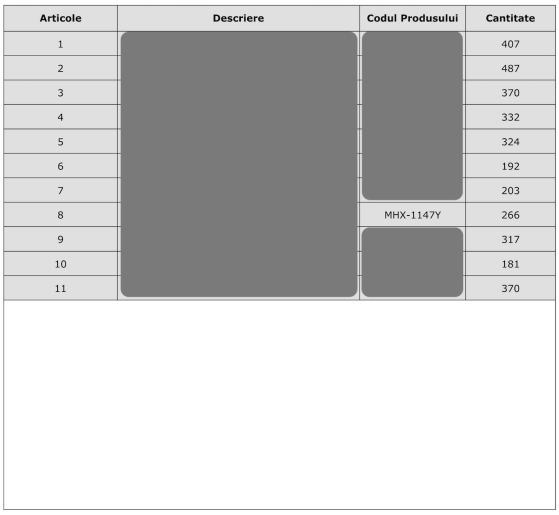

Figure 8: The corresponding image(cropped and censored) for example 5.

Example 4:

Data entry: `"lineItem.supplierMaterialNumber"`
Ground truth: ["KL-840I"]
Prediction: `"KL-8401"`
Cause: *"The model misinterpreted the quantity field as the item number due to their close proximity within the document."*

Example 5

As shown in Figure 8, for the data entry "lineItem.supplierMaterialNumber", the ground truth specifies "MHX-1147Y", whereas the prediction incorrectly records it as "*MHX*-1147Y". This error stems from the misinterpretation of the character "X" as the Greek letter "*X*" (Chi), due to their visual similarity.

Example 5:

Data entry: `"lineItem.supplierMaterialNumber"`
Ground truth: ["MHX-1147Y"]
Prediction: `"\u039c\u0397\u03a7-1147Y"`
Cause: *"The character 'X' was misinterpreted as the Greek letter 'Χ'."*

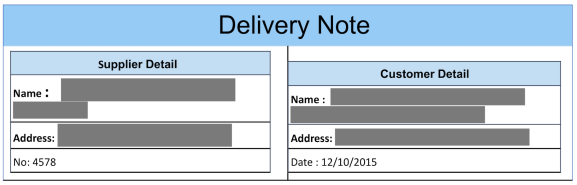

Figure 9: The corresponding image(cropped and censored) for example 6.

Example 6

For the data entry "deliveryNoteNumber", the ground truth indicates "4578" but the prediction yields an empty result. The cause analysis shows that the field is not recognized in the image text. In Figure 9, the ground truth "4578" appears under "Supplier Detail" rather than being explicitly labelled as "deliveryNoteNumber", presenting a challenge for the extraction model in terms of high-level layout comprehension and reasoning.

Example 6:

Data entry: `"deliveryNoteNumber"`
Ground truth: ["4578"]
Prediction: `""`
Cause: *"Prediction was empty because the field was not explicitly recognized in the image text."*

D.3 OCR schema ambiguity

Example 7

For the data entry "lineItem.quantity", the ground truth specifies "3" whereas the prediction inaccurately states "12" The cause analysis suggests that the error is due to incorrect logic or misalignment in OCR. In Figure 10, both "3" and "12" are located within the quantity column, but they appear in different rows. OCR misalignment or incomplete structured data led the prediction to mistakenly extract "12" from a neighboring row, rather than the correct value "3".

Example 7:

Data entry: `"lineItem.quantity"`
Ground truth: ["3"]
Prediction: `"12"`
Cause: *"Incorrect logic or misalignment in OCR could cause quantity mismatch."*

Example 8 & 9

For the data entries "lineItem.itemNumber" and "lineItem.quantity", the ground truth specifies "1" and "13", whereas the predictions are "8" and "7", respectively. The cause analysis suggests that the error results from OCR extracting both fields as adjacent tokens without clear separation or labeling. In the OCR output, the item number and quantity values appear consecutively in a single text segment or without distinct bounding boxes. As a result, when the LLM processes this unstructured or ambiguously segmented text, it may confuse the associations between values and fields. In this case, the model likely misaligned the detected numbers, attributing "8" to the item number and "7" to the quantity, rather than correctly mapping "1" and "13". Figure 11 shows that the close spatial proximity of numeric fields contributed to this misinterpretation.

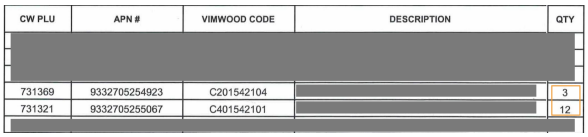

Figure 10: The corresponding image(cropped and censored) for example 7.

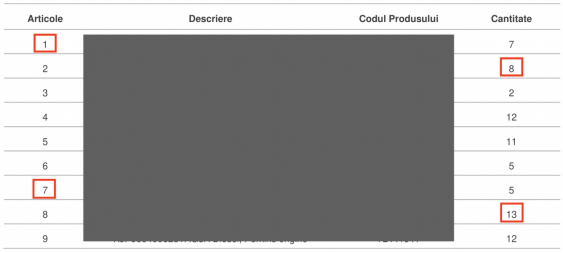

Figure 11: The corresponding image(cropped and censored) for example 8 and 9.

Example 8:

Data entry: `"lineItem.itemNumber"`
Ground truth: ["1"]
Prediction: `"8"`
Cause: *"The OCR data extracted the itemNumber and quantity as adjacent fields, which can lead to misinterpretation by the LLM."*

Example 9:

Data entry: `"lineItem.quantity"`
Ground truth: ["13"]
Prediction: `"7"`
Cause: *"The OCR data extracted the itemNumber and quantity as adjacent fields, which can lead to misinterpretation by the LLM."*

E Refined Prompt Template

We cannot disclose the format instructions and document schema information. Therefore, we have omitted these two variables, but all other details for our refined prompt template are presented below:

Prompt Template for Image-only Input:

You are a warehouse manager receiving a delivery. As an expert, you will go through the attached delivery note and carefully extract the data required to receive the shipped goods and process them in your ERP system. Focus on the actually received goods and quantities.

The document may be in English, German, or any other language. Some fields may be indicated by abbreviations. Extract only the information present in the document. If you have doubts about a field, skip it.

Format instructions: {modified format instructions}. {modified document schema}.

Return date fields in YYYY-MM-DD format. For country and currency, use ISO format. Do not include the schema in the answer. Ensure that all fields are returned as valid values or empty strings (‘"""’), rather than null. If a field does not have a value, return it as an empty string.

Always return valid JSON and do not wrap your response in backticks! Ensure that the JSON structure is valid and does not contain any extra commas or brackets. Each object should be properly closed without trailing commas.

Be attentive to abbreviations and language variations in the document, and ensure that you extract the correct information based on context. Validate the JSON structure before returning the output, checking for any syntax errors. Accuracy in the extraction process is crucial, ensuring that all relevant details are captured accurately.

Emphasize the importance of accuracy in the extraction process and encourage the model to double-check its outputs against the provided schema. Pay special attention to context clues in the document to accurately extract and interpret abbreviations and language variations. Your output must reflect the exact information present in the document, as inaccuracies can lead to significant operational issues.

Here is the document image: