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

Anonymous ACL submission

Abstract

Multimodal Large Language Models (MLLMs) enhance the potential of natural language processing. However, their actual impact on document information extraction remains unclear. In this work, we conduct a large-scale benchmarking study on information extraction for business documents, evaluating state-of-the-art MLLMs. To 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, receipts, insurance quotes, and financial statements, and turning it into structured information. This capability is used in various industries, including healthcare, insurance, and legal document processing. Using document information extraction techniques, businesses can automate and streamline document-based and scene-text workflows, significantly reducing manual effort, avoiding errors, and 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 a structured information following a schema (Wang et al., 2023b). This approach, while effective, is inherently complex and is prone to error propagation from the OCR to the downstream extraction step. Recent studies have explored OCR-free algorithms and MLLMs as potential alternatives (Kim et al., 2022; Ye et al., 2023; Liu et al., 2024). Although many of these newer models claim superior performance, their effectiveness in real-world industry scenarios remains highly uncertain, and research in this area is still sparse.

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. To the best of our knowledge, this is the first large-scale benchmarking study of MLLMs for document information extraction that is ecologically valid for real-world use and conditions.

Furthermore, we leverage large language model (LLM) 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 still necessary for document information extraction? Can MLLMs replace traditional pipelines? 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 negative impact. Our findings suggest that image-only input 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. General-purpose LLMs lack task-specific knowledge, highlighting the need for more carefully designed schema, exemplars, and instructions. We refine our approach and achieve measurable performance improvement by leveraging insights from our error analysis framework.

2 Related Work

2.1 Two-stage OCR-based Document Information Extraction

The two-stage document information extraction process involves OCR pre-processing to obtain textual content, followed by another model that extracts the structured information from the OCR-derived text (Katti et al., 2018; Hong et al., 2022). However, a key challenge is to preserve the spatial layout of the original document. To address this, some methods (He et al., 2023) introduce coordinate tokens (e.g. $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$) to integrate layout and text information into a unified sequence. Others, such as DocLLM (Wang et al., 2023a), leverage spatial attention mechanisms to enhance text-layout alignment. LayoutLLM (Huang et al., 2022) incorporates pre-trained layout-aware models to jointly process visual, textual, and spatial data. Despite these advancements, OCR-based methods rely on text extraction from external systems, increasing computational costs and processing complexity. In addition, they may inherit OCR errors and positional inaccuracies, which introduce challenges for downstream information extraction tasks.

2.2 OCR-free MLLMs for Document Information Extraction

Recent research has increasingly focused on developing streamlined end-to-end methodologies to process documents directly without relying on any OCR engines (Kim et al., 2022; Lee et al., 2023). Although these approaches successfully eliminate the dependency on OCR tools, they still require task-specific finetuning. In this context, MLLMs have emerged as transformative technologies. These powerful models are explicitly pretrained on large-scale, diverse image datasets and further fine-tuned with instruction-following capabilities. This enables them to effectively understand visual content and perform zero-shot predictions. For example, recent advances such as Qwen-VL (Bai et al., 2023) and Gemini (Team et al., 2023) have demonstrated strong visual reasoning skills, including extracting relevant information from images in response to user queries. However, a comprehensive benchmark for MLLMs on document information extraction remains absent. To bridge this gap, our aim is to provide a rigorous evaluation and to provide a fair comparison of their effectiveness in real-world applications.

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 which has been developed for business documents and achieves high accuracy in this domain. In addition, considerable effort was dedicated to ensuring high accuracy of the structured ground-truth annotations. Unlike existing open-source invoice or receipt datasets, which typically contain simplified structures with clear key-value pairs or fixed recognized structures, our dataset presents greater complexity, including nested information, stacked cells within line items, ambiguous dates, and inconsistent header formats, among other challenges. These factors introduce additional difficulties in document parsing, making our dataset a more realistic benchmark for evaluating document information extraction models in industrial applications.

Specifically, we want our model to parse and extract several *header fields* (i.e., Country, Currency, Invoice Number, Invoice Date, Invoice Amount, Supplier ID) as well as the list of purchased items, referred to as *line items*. Line items include details, such as ‘Description’ or ‘Number ID’, ‘Quantity’ and ‘Amount’, for each item. We note that while header fields may appear only once in one document, multiple line items may appear in the same document.

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 LLM-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 LLM 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 the Appendix A.

In the final stage, we compute the evaluation metrics. We first compare the ground-truth annotations with model predictions and classify the results as true/false positives/negatives based on their correctness. The following categories are considered:

- **vc (value-correct)**: The predicted value matches one of the ground-truth values (true positive).
- **vn (value-none)**: The ground-truth contains a value, but the model prediction is empty (false negative).
- **vw (value-wrong)**: A value is predicted, but it does not match any ground-truth values (false positive).
- **nv (none-value)**: A value is predicted, but the ground-truth is explicitly empty (false positive).
- **nn (none-none)**: No value is predicted, and the ground-truth is also empty (true negative).

From these metrics, we derive aggregate scores, either at the field level, document level, or dataset-wide. Then, the overall evaluation metrics are computed as follows:

$$\text{accuracy} = \frac{vc + nn}{vc + vn + vw + nv + nn} \quad (1)$$

$$\text{recall} = \frac{vc}{vc + vn + vw} \quad (2)$$

$$\text{precision} = \frac{vc}{vc + vw + nv} \quad (3)$$

$$F1 = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (4)$$

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.

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 prediction 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 (similar to the **vc, vn, vw, nv** and **nn**), 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

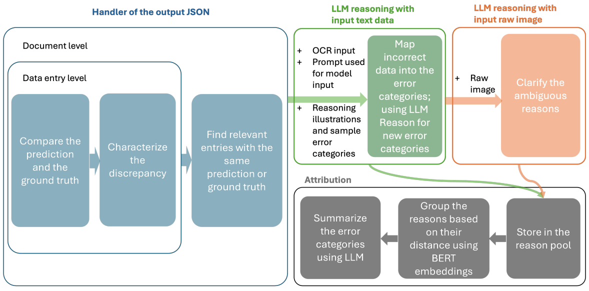

Handler of the output JSON
Document level
Data entry level
Compare the prediction and the ground truth
Characterize the discrepancy
Find relevant entries with the same prediction or ground truth
LLM reasoning with input text data
+ OCR input
+ Prompt used for model input
+ Reasoning illustrations and sample error categories
Map incorrect data into the error categories; using LLM Reason for new error categories
LLM reasoning with input raw image
+ Raw image
Clarify the ambiguous reasons
Attribution
Summarize the error categories using LLM
Group the reasons based on their distance using BERT embeddings
Store in the reason pool

Figure 1: Hierarchical Error Analysis Framework

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 error reasons, which forms 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 model behaviour across multiple documents to determine whether errors originate from OCR misrecognition, layout misinterpretation, prompt misalignment, model capability issues, or schema inconsistencies. This attribution process provides insights into systematic failure patterns, helping prioritize areas for improvement through prompt optimization, schema adjustment, or dataset augmentation.

4 Experiments

4.1 Baselines

For each MLLM, we evaluate three different input formats: document image-only, OCR-extracted text, and a combination of both. In contrast, for LLMs, we only provide OCR-extracted text as input. Due to confidentiality, we cannot provide the names and specific details of the LLM/MLLMs selected for evaluation. However, all models used in our experiments are flagship models from major companies.

As many companies categorize their models based on parameter size and capabilities, we adopt a standardized naming method to ensure anonymity and abstraction. Specifically, we use `Large` to denote the most powerful model from each company, `Middle` for mid-tier models, and `Small` for lightweight ones. Furthermore, if a model has undergone significant updates, we denote different versions using numerical markers such as *V1, V2*, etc.

To ensure the relevance of our evaluation to both the ongoing development of LLMs, we exclusively select models released in 2024. This choice represents the state-of-the-art advances in LLM technology at the time of our investigation.

4.2 Experiment Results

Table 1 presents a comparative analysis of different LLMs and MLLMs in the three evaluation settings: Image-only, OCR-only, and Image + OCR. Performance is measured using F1-score on our two business document datasets — C1 (from supply chain) and C2 (from finance) — with the arithmetic mean providing an overall assessment.

First, the mean scores of all models, either text-only or vision-enabled, for OCR-only input fall consistently within the range 66%-74%, with relatively low variance. However, image-only inputs widen the performance range and reveal greater differences among the large models from various providers. Finally, when image and OCR inputs are combined, models with relatively weaker visual processing capabilities tend to rely more on OCR-derived text, bringing their scores closer to the 70% average, presumably by leveraging the models' comparatively stronger text processing capabilities.

4.3 Analysis

4.3.1 Is OCR necessary for MLLMs?

From Table 1, we observe that models tend to perform better when both modalities are involved. This aligns with the intuition that incorporating additional sources of information should enhance performance by providing a more comprehensive understanding of the input.

However, an interesting phenomenon emerges when analyzing Company E's flagship model. Unlike the models from Companies C and D, its per-

Table 1: Performance comparison of different LLMs 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
LLM:										
Company A	Large V1		-		76.0	69.5	72.8		-	
Company B	Large V1		-		68.7	65.1	66.9		-	
MLLM:										
Company C	Large V1	68.7	57.4	63.1	75.3	71.2	73.3	72.7	68.0	70.4
Company D	Large V1	43.8	56.4	50.1	72.0	68.2	70.1	74.0	69.1	71.5
	Large V2	65.0	69.3	67.2	73.7	**72.6**	72.8	73.6	69.6	71.6
Company E	Large V1	**87.3**	66.4	**76.8**	**78.4**	69.8	**74.1**	**86.2**	65.0	**75.6**
	Large V2	75.2	**73.3**	74.3	77.6	69.5	73.6	77.1	**73.2**	75.2

formance does not significantly degrade when using only images as input, without OCR-extracted text. In some cases, it even shows improvements. Initially, we considered this to be a coincidence. However, the trend consistently persists after multiple reevaluations using different sampling methods. This implies that Company E's model can directly extract structured information from textual images and effectively comprehend content, without relying on OCR as an intermediary. OCR-generated text does not provide additional benefits to Company E's model.

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 2, the overall performance improves as the size of the model increases. 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 Middle V1 is already capable of a rather high baseline performance score.

A particularly interesting observation is that for the Small V2 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 model sizes exhibit varying levels of dependence on OCR-extracted text. These findings provide new insights into the scalability of MLLMs and highlight the potential of vision models to process textual information effectively.

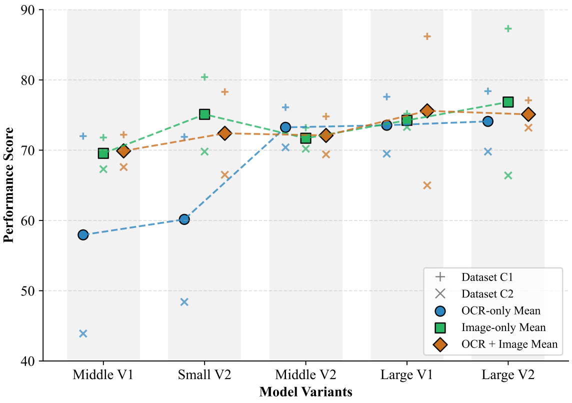


Figure 2: Performance comparison on various size models across different input types. The small shape (●, ■, ◆) denotes the arithmetic mean across two different categories of dataset. + is the F1-score in C1, while × is for C2.

5 Discussion

We employ our hierarchical error analysis framework to categorize the underlying causes of errors. Figure 3 presents the results and representative failure cases for each category are listed in Appendix

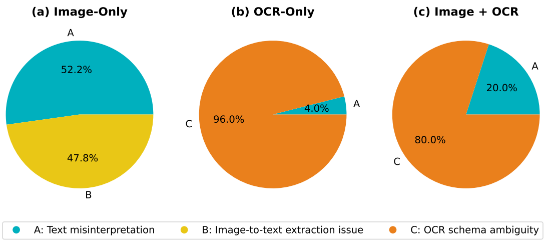


Figure 3: Error analysis results for three different input modalities.

Company E Large V1	Initial	Final
Dataset C1	87.3	**89.1**
Dataset C2	66.4	**68.6**
Mean	76.8	**78.9**

Table 2: Performance results for the optimized prompt template with image-only input.

B. 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-related issues (Error **C**), which stem from inaccuracies in text recognition and confusion in document schema description.

Based on these, we apply several enhancements to improve performance:

- 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 C) for the input of only images. As shown in Table 2, 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 required in some cases. Furthermore, our automated error analysis demonstrates the impact of well-designed schema, exemplars, and instructions on improving MLLMs performance. We believe that these findings offer valuable insight to advance research in document information extraction.

Limitations

Despite promising results, our approach has several limitations. First, the LLM reasoning process could potentially be enhanced for the error analysis pipeline by incorporating a dedicated reasoning model, such as O1 (Jaech et al., 2024) or DeepSeek R1 (Guo et al., 2025). Second, we were not able to perform few-shot fine-tuning on MLLMs due to time constraints. Incorporating lightweight adapters could potentially allow MLLMs to achieve even better performance.

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*.

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. ICL-D3IE: In-context learning with diverse demonstrations updating for document information extraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19485–19494.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775.

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 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.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

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*.

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*.

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

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023b. 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:

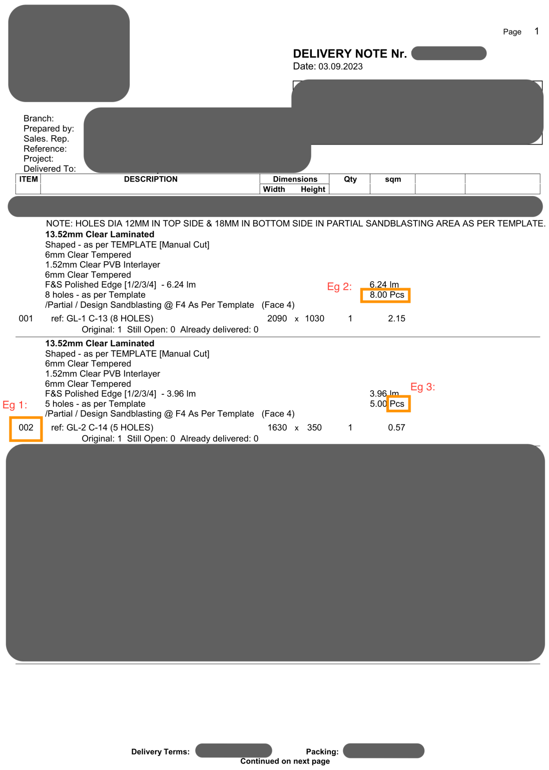
```
{
"deliveryDate": [""],
"deliveryNoteNumber": ["ID"],
"documentDate": ["YYYY-MM-DD"],
"purchaseOrderNumber": [""],
"supplierId": [""],
"lineItems": [
{
"lineItem.customerMaterialNumber": "",
"lineItem.itemNumber": "1",
"lineItem.purchaseOrderItemNumber": "",
"lineItem.purchaseOrderNumber": "",
"lineItem.quantity":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

B Failure Case Study

B.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 4 is "002" confirms the correct OCR extraction. This suggests that the error is due to omission in the interpretation of the format guideline.



[redacted]

Page 1

DELIVERY NOTE Nr. [redacted]
Date: 03.09.2023

[redacted]

Branch:
Prepared by:
Sales. Rep.
Reference:
Project:
Delivered To:
[redacted]

ITEM	DESCRIPTION	Dimensions		Qty	sqm		
		Width	Height				
[redacted]							
	NOTE: HOLES DIA 12MM IN TOP SIDE & 18MM IN BOTTOM SIDE IN PARTIAL SANDBLASTING AREA AS PER TEMPLATE. **13.52mm Clear Laminated** Shaped - as per TEMPLATE [Manual Cut] 6mm Clear Tempered 1.52mm Clear PVB Interlayer 6mm Clear Tempered F&S Polished Edge [1/2/3/4] - 6.24 lm 8 holes - as per Template /Partial / Design Sandblasting @ F4 As Per Template (Face 4)				6.24 lm 8.00 Pcs		
001	ref: GL-1 C-13 (8 HOLES) Original: 1 Still Open: 0 Already delivered: 0	2090 x	1030	1	2.15		
	13.52mm Clear Laminated Shaped - as per TEMPLATE [Manual Cut] 6mm Clear Tempered 1.52mm Clear PVB Interlayer 6mm Clear Tempered F&S Polished Edge [1/2/3/4] - 3.96 lm 5 holes - as per Template /Partial / Design Sandblasting @ F4 As Per Template (Face 4)				3.96 lm 5.00 Pcs		
002	ref: GL-2 C-14 (5 HOLES) Original: 1 Still Open: 0 Already delivered: 0	1630 x	350	1	0.57		

[redacted]

Delivery Terms: [redacted] **Packing:** [redacted]
Continued on next page

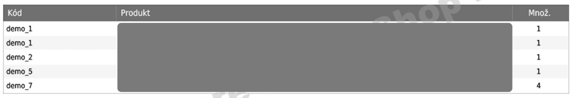
Figure 4: The corresponding image(cropped and censored) for example 1,2 and 3.

Example 1:

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

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 4 and correctly identifies it as the quantity attribute, suggesting a text misinterpretation problem.



Kód	Produkt	Množ.
demo_1	[redacted]	1
demo_1	[redacted]	1
demo_2	[redacted]	1
demo_5	[redacted]	1
demo_7	[redacted]	4

Figure 5: The corresponding image(cropped and censored) for example 4.

Example 2:

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

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"*

B.2 Image-to-text extraction issue

Example 4

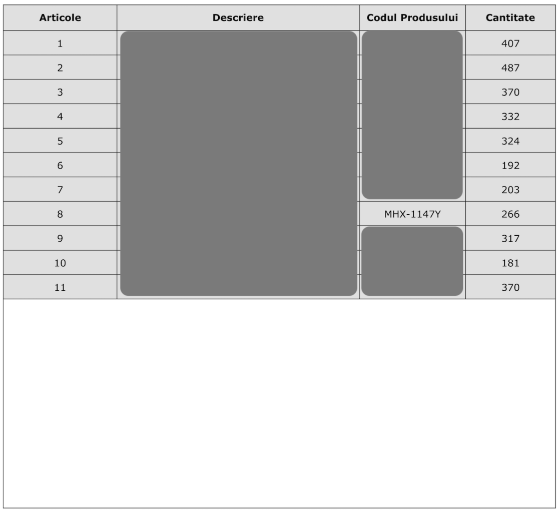
For the data entry "lineItem.itemNumber," the prediction specifies the item number as "1", but the ground truth is empty, indicating that the model fails to extract correct text for the corresponding field, as shown in Figure 5. The identified cause indicates that the model mistakenly assigned the quantity field's value as the item number due to their proximity or contextual confusion within the document. This demonstrates the effectiveness of the error analysis framework in systematically identifying and reasoning through the misalignment based on related data entries.

Example 4:

Data entry: "lineItem.itemNumber"
Ground truth: [""]
Prediction: "1"
Cause: *"The model misinterpreted the quantity field as the item number due to their close proximity within the document."*

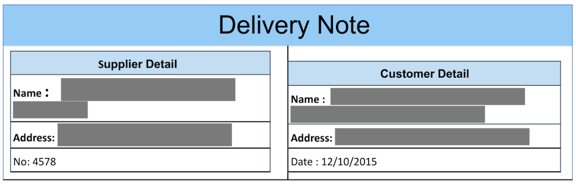
Example 5

As shown in Figure 6, 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.



Articole	Descriere	Codul Produsului	Cantitate
1	[redacted]	[redacted]	407
2	[redacted]	[redacted]	487
3	[redacted]	[redacted]	370
4	[redacted]	[redacted]	332
5	[redacted]	[redacted]	324
6	[redacted]	[redacted]	192
7	[redacted]	[redacted]	203
8	[redacted]	MHX-1147Y	266
9	[redacted]	[redacted]	317
10	[redacted]	[redacted]	181
11	[redacted]	[redacted]	370

Figure 6: The corresponding image(cropped and censored) for example 5.



Delivery Note

Supplier Detail
Name : [redacted]
Address: [redacted]
No: 4578

Customer Detail
Name : [redacted]
Address: [redacted]
Date : 12/10/2015

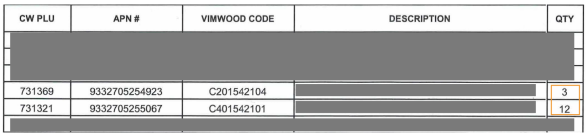
Figure 7: The corresponding image(cropped and censored) for example 6.

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 ’X’."*

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 7, 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.



CW PLU	APN #	VIMWOOD CODE	DESCRIPTION	QTY
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
731369	9332705254923	C201542104	[redacted]	3
731321	9332705255067	C401542101	[redacted]	12
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Figure 8: The corresponding image(cropped and censored) for example 7.

Example 6:

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

B.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 8, both "3" and "12" are located within the quantity column, but they appear in different rows. The OCR misalignment likely caused the prediction to capture "12" from an adjacent row instead of the correct "3."

Example 7:

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

C Updated Prompt Template

We cannot disclose the format instructions and document schema information. Therefore, we have omitted these two variables, but all other details 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: