From Pixels to Pairs: A Comprehensive Benchmark of LLM-Driven Key–Value Extraction in Noisy Document Settings

Zahra Anvari* Vassilis Athitsos†

August 4, 2026

Abstract

Large language models (LLMs) are increasingly used for structured information extraction from documents, yet their behavior under realistic OCR noise remains poorly understood. In this work, we present a systematic and controlled benchmark of open-source instruction-tuned LLMs for key–value pair (KVP) extraction under both clean-text and noisy OCR conditions.

We evaluate representative decoder-only models (Gemma, Mistral, Qwen2.5, LLaMA 3, and DeepSeek) on three document benchmarks (FUNSD, CORD, and SROIE), using both Gold-text annotations and OCR outputs from PaddleOCR, EasyOCR, and Tesseract. Our unified evaluation protocol isolates the effects of input quality, model design, and prompting under consistent conditions.

Our results show that modern LLMs act as strong semantic extractors when high-quality text is available, in some cases approaching supervised layout-aware systems without task-specific training. However, this capability does not translate reliably to realistic settings. Under OCR noise, performance degrades substantially, and performance gaps between models narrow as input corruption increases.

Across all datasets and models, we observe that extraction performance is governed by a two-stage bottleneck: (1) semantic reasoning over text and (2) preservation of textual fidelity under OCR noise. While model scaling improves results in the clean-text regime, these gains diminish under noisy inputs, where upstream text quality becomes the dominant factor.

We further identify recurring failure modes—including key–value misalignment, hallucination, and numeric corruption—that are amplified by OCR errors and prompt sensitivity.

Overall, our findings highlight a fundamental gap between clean-text evaluation and real-world deployment, and demonstrate that robust document extraction requires jointly addressing OCR quality, structural reasoning, and LLM-based semantic modeling. We release our benchmark to support reproducible and realistic evaluation of document understanding systems.

1 Introduction

Key–value pair (KVP) extraction is a core task in Document AI, enabling applications such as invoice processing, receipt understanding, form digitization, and enterprise automation [11, 9]. Traditional pipelines typically combine optical character recognition (OCR) with heuristic post-processing and domain-specific rules. While effective in controlled settings, these approaches degrade under real-world conditions where documents exhibit noise, irregular layouts, and non-standard field structures. In practice, OCR errors—such as token fragmentation, line merging, and numeric corruption—propagate directly to downstream extraction systems and reduce reliability.

Large language models (LLMs) have recently introduced a new paradigm for document understanding [1, 12, 26]. Rather than explicitly modeling layout or visual structure, instruction-tuned LLMs can infer semantic relationships directly from text using large-scale pretraining and in-context reasoning. Under clean input conditions, prior work has shown that LLMs can perform structured extraction tasks without task-specific fine-tuning [27, 25]. However, most existing evaluations rely on clean, human-curated text rather than OCR-generated inputs. This distinction is critical, as OCR noise remains a primary source of error in real-world document processing pipelines.

*Independent AI Researcher. Email: zahra.anvari.uta@gmail.com
†Professor, Computer Science and Engineering, University of Texas at Arlington. Email: athitsos@uta.edu

This discrepancy creates a gap between benchmark evaluation and deployment conditions. Reported improvements may reflect performance under idealized inputs rather than robustness to realistic noise. In addition, comparisons across models are often influenced by inconsistent prompting strategies, post-processing pipelines, or dataset-specific assumptions, making it difficult to isolate the effect of model capability. Existing studies also tend to evaluate datasets in isolation, despite the fact that forms and receipts impose distinct structural and semantic challenges.

In this work, we present a controlled benchmark of LLM-based KVP extraction under both clean-text and OCR-derived inputs. We evaluate models across multiple datasets representing diverse document types, including forms (FUNSD [11]) and receipts (SROIE [9], CORD [19]). To reflect realistic usage scenarios, we compare performance across input regimes using Gold-text annotation-derived text and OCR outputs from widely used engines such as Tesseract [23], EasyOCR [10], and PaddleOCR [5].

Our analysis examines how extraction performance varies with both model capability and input quality. While stronger models improve performance under clean-text conditions, these gains become less pronounced as input quality degrades. This suggests that improvements in model capacity do not consistently translate to proportional gains under realistic OCR conditions.

We evaluate a diverse set of instruction-tuned open-source LLMs, including Gemma [26], Mistral [12], Qwen2.5 [20], DeepSeek [3], and LLaMA 3 [1]. To ensure fair comparison, all models are evaluated using a unified prompting framework, deterministic decoding, and a strictly non-semantic canonicalization pipeline that standardizes output format without introducing heuristic extraction logic.

Performance is measured using complementary metrics including Key Recall, Exact Match (EM), and Value-level F1, capturing field coverage, strict correctness, and robustness to textual variation. This setup enables controlled analysis of model behavior across datasets and input regimes.

Our results show several consistent patterns. First, instruction-tuned LLMs achieve strong extraction performance under clean-text conditions across all datasets. Second, performance decreases under OCR inputs, with the magnitude of degradation varying by dataset and document structure. Third, differences between models become less pronounced as input quality decreases. Finally, few-shot prompting improves performance in many settings, but its impact depends on dataset regularity and input quality.

By releasing our prompts, inference pipeline, and evaluation framework, we aim to establish a reproducible benchmark for LLM-based document extraction. Overall, our findings highlight the importance of evaluating models under realistic input conditions and clarify when prompt-based LLM extraction is effective in practice.

Contributions. This paper makes the following contributions:

- We present a controlled benchmark for KVP extraction with LLMs under both clean-text and OCR-derived inputs, enabling systematic analysis of extraction robustness across input regimes.
- We evaluate multiple instruction-tuned LLMs across diverse document types (forms and receipts) and OCR pipelines using a unified experimental setup.
- We introduce a standardized evaluation pipeline with deterministic decoding and strictly non-semantic output canonicalization, ensuring fair and reproducible comparison across models.
- We provide empirical analysis showing how input quality, dataset structure, and model choice jointly influence extraction performance.
- We release our prompts, inference pipeline, and predictions to support reproducible research in document understanding.

2 Related Work

2.1 Key–Value Pair Extraction in Document AI

Key–value pair (KVP) extraction is a fundamental task in Document AI, enabling applications such as invoice processing, form digitization, identity document parsing, and enterprise automation. Early approaches relied on hand-crafted templates, regular-expression heuristics, and spatial rules [21, 6]. While effective in controlled settings, these systems

were brittle: minor layout changes, OCR artifacts, or domain shifts often caused failures. Their reliance on fixed patterns limited semantic generalization and required extensive manual effort to adapt across document types.

Deep learning significantly advanced this area. Chargrid [13] introduced grid-based representations that encode both text and spatial layout, enabling convolutional models to jointly reason over content and structure. The release of FUNSD [11] further accelerated progress by providing a benchmark for form understanding and key–value linking. However, many neural approaches still depend on large annotated datasets and struggle to generalize across unseen layouts or document domains.

2.2 Layout-Aware Architectures

Layout-aware transformers represent a major advancement by integrating textual content with spatial and visual features. LayoutLM [30] demonstrated that combining language modeling with 2D positional embeddings yields strong performance on key information extraction tasks. Subsequent variants such as LayoutLMv2 and LayoutLMv3 [31, 8] further improved results through multimodal pretraining and better alignment between visual and textual signals.

Follow-up models—including DocFormer [2], FormNet [15], StructuralLM [16], and UDOP [25]—extended this paradigm with more sophisticated fusion mechanisms and hierarchical representations. These approaches achieve strong performance on benchmarks such as FUNSD [11], CORD [19], and SROIE [9].

Despite their effectiveness, layout-aware models have practical limitations. They rely on accurate OCR bounding boxes, which degrade under noisy scans or low-resolution inputs. Training often requires large-scale annotated corpora, and deployment typically involves a full vision–language pipeline, increasing engineering complexity and computational cost. Moreover, their reliance on explicit layout signals can limit robustness when structure is degraded.

Recent multimodal generative systems such as Donut [14], DocVLM [22], and LayoutLLM [17] attempt to unify visual understanding with generative modeling. While these approaches can leverage full document images, they are often less compatible with existing OCR-based pipelines used in many real-world systems.

2.3 Large Language Models for Structured Extraction

Large language models (LLMs) have introduced a flexible paradigm for structured information extraction from semi-structured text. Modern decoder-only, instruction-tuned models—such as Mistral [12], Gemma [26], Qwen2.5 [20], DeepSeek [3], and LLaMA 3 [1]—demonstrate strong zero-shot and few-shot capabilities across a wide range of extraction tasks.

These models can infer key–value relationships, perform entity recognition, and generate structured outputs directly from natural-language prompts, without task-specific training. Instruction tuning [29] and unified extraction frameworks such as InstructUIE [27] further highlight the ability of LLMs to generalize across extraction tasks using a single prompting interface.

However, existing evaluations of LLM-based document extraction remain limited. Many studies rely on clean, human-curated text rather than OCR outputs, making them unrepresentative of real-world conditions. Others evaluate only a narrow set of models or use inconsistent prompting and post-processing strategies, hindering reproducibility and fair comparison. As a result, the robustness of modern open-source LLMs under realistic OCR noise remains insufficiently understood.

2.4 OCR Noise and Robustness

OCR-induced corruption is a primary source of failure in document processing pipelines. Errors such as character substitutions, token fragmentation, and line reordering directly affect downstream extraction. While robustness to noise has been studied in NLP [24, 18], relatively little work examines how LLMs behave under OCR-generated text.

In Document AI, evaluations are typically performed either on clean annotations or using end-to-end multimodal systems that bypass OCR. Consequently, there is limited empirical understanding of how text-only LLM extraction pipelines perform under realistic OCR conditions produced by widely used engines such as Tesseract [23], EasyOCR [10], and PaddleOCR [5].

2.5 Positioning of This Work

This work addresses these gaps by providing a controlled and reproducible benchmark of open-source, instruction-tuned decoder-only LLMs for KVP extraction under both clean-text and noisy OCR conditions. We evaluate multiple model families across diverse document types (forms and receipts) and systematically vary OCR quality to isolate its impact on extraction performance.

Our framework uses a unified prompting strategy, deterministic decoding, and a strictly non-semantic canonicalization pipeline to ensure fair comparison across models. By separating model capability from input quality, we provide a clearer understanding of when prompt-based extraction succeeds and where it breaks down in realistic document processing settings.

2.5.1 Dataset Samples and Visual Characteristics

Figures 1, 2, and 3 show representative examples from the CORD, FUNSD, and SROIE datasets, illustrating the structural diversity and visual variability that drive many of the failure modes observed in our benchmark.

CORD (Receipts with dense numeric structure). CORD consists of Indonesian receipts with varying lengths and complexity. Documents contain multiple itemized entries along with aggregated fields such as Subtotal, Service Charge, VAT, and Total. As shown in Figure 1, receipts exhibit blur, uneven lighting, color cast, and background clutter.

These characteristics directly contribute to extraction errors observed in our results. In particular, dense numeric regions and repeated values increase the risk of key–value misalignment, while OCR-induced digit corruption (e.g., "193.00" $\rightarrow$ "19300") significantly degrades Value F1. Multi-line item lists further introduce ambiguity in associating quantities, prices, and totals.

FUNSD (Form-based documents with spatial dependencies). FUNSD contains scanned forms where key–value relationships are defined by spatial layout rather than sequential text. As illustrated in Figure 2, keys and values are often separated across regions (e.g., left/right alignment or distant blocks), requiring spatial reasoning to associate them correctly.

When converted to OCR text, this spatial structure is largely lost, resulting in flattened sequences that obscure key–value relationships. This leads to frequent value misalignment and reduced Exact Match scores in text-only extraction settings, particularly for fields that rely on positional cues rather than explicit lexical markers.

SROIE (Receipts with limited schema). SROIE consists of shorter receipts with a small set of canonical fields such as *total*, *date*, and *address*. As shown in Figure 3, these documents are structurally simpler than CORD but still exhibit OCR challenges including digit corruption, token merging, and inconsistent formatting.

While the reduced annotation schema lowers structural ambiguity, it introduces a different evaluation challenge: semantically correct extractions outside the annotated schema are ignored. In addition, even minor OCR variations in numeric or address fields can reduce both Exact Match and Value F1, making this dataset particularly sensitive to normalization errors.

3 Methodology and Experimental Setup

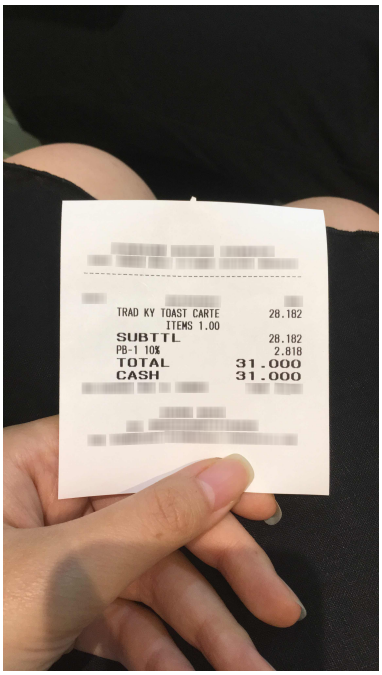
This section describes the methodological design and experimental configuration of our benchmark. We first formalize the key–value pair (KVP) extraction task, then detail the evaluated models, prompting strategy, and output canonicalization. Finally, we specify datasets, input regimes, and inference settings to ensure reproducibility and controlled comparison.

3.1 Task Formulation

Given a document image I, an OCR engine produces a textual transcript T, which may contain character errors, token fragmentation, and loss of structural ordering. The KVP extraction task aims to map this text into a structured set of semantic fields:

$$\mathcal{P} = \{(k_i, v_i)\}_{i=1}^{N},$$

(a) Short receipt



(b) Multi-item receipt

(c) Medium-complexity receipt

Figure 1: Representative CORD receipts illustrating dense numeric structure, multi-line item lists, and real-world imaging artifacts (e.g., blur, lighting, background clutter) that contribute to OCR-induced errors and key–value misalignment.

Table 1: Summary of evaluated LLMs.

Model	**Params**	**Family**	**Context**	**Tokenizer**
Gemma 2B Instruct	2B	Gemma	8k	SentencePiece
Gemma 7B Instruct	7B	Gemma	8k	SentencePiece
Mistral 7B Instruct v0.2	7B	Mistral	32k	BPE
Qwen2.5 7B Instruct	7B	Qwen	32k	BPE-style
DeepSeek LLM 7B Chat	7B	DeepSeek	4k	SentencePiece
LLaMA 3 8B Instruct	8B	LLaMA	8k	BPE-style

where k_i denotes a semantic key (e.g., `TOTAL`, `DATE`) and v_i is its corresponding value.

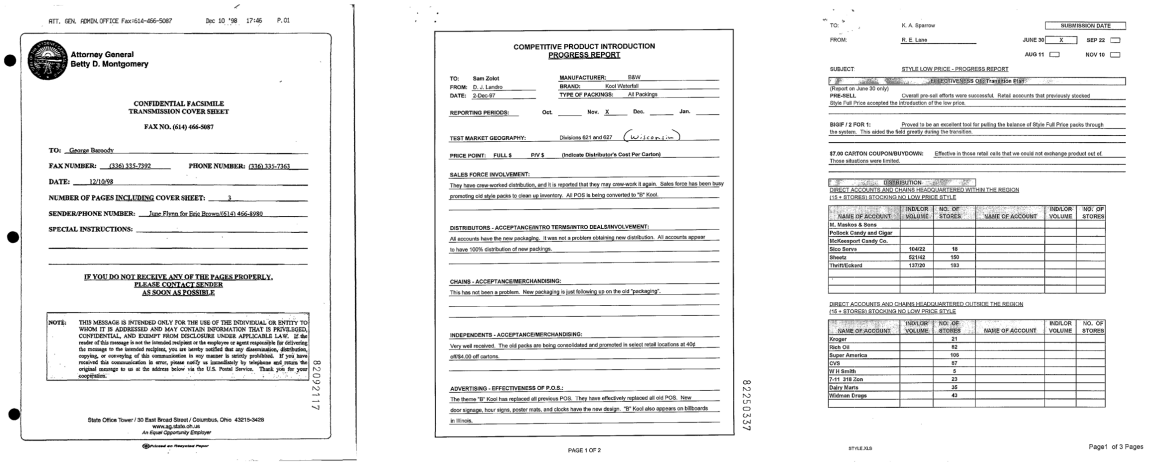
Compared to standard sequence labeling tasks such as named entity recognition, KVP extraction introduces additional challenges. Keys may be implicit, values may span multiple lines, and multiple candidate values may exist for a single key (e.g., subtotal vs. total). OCR further complicates the task by introducing numeric corruption, missing punctuation, and inconsistent token boundaries. In form-like documents, flattening spatial layout into text can also obscure relationships between semantically linked fields.

3.2 Models Evaluated

We evaluate six open-source instruction-tuned decoder-only LLMs spanning the Gemma, Mistral, Qwen2.5, DeepSeek, and LLaMA 3 families. These models were selected to cover a representative range of parameter scales, context lengths, and tokenizer designs while remaining fully reproducible.

Table 1 summarizes the evaluated models.

These factors are relevant for document extraction. Parameter scale affects semantic reasoning capacity, context length determines whether long receipts can be processed without truncation, and tokenizer design influences robustness to OCR artifacts such as broken numbers and irregular spacing.



(a) Form sample 1 (b) Form sample 2 (c) Form sample 3

Figure 2: Representative FUNSD forms illustrating spatially distributed key–value relationships. Layout-dependent associations are lost when converted to OCR text, leading to value misalignment in text-only extraction.

3.3 Prompting Strategy

A unified, dataset-agnostic prompt is used across all models. The prompt (i) defines the extraction task, (ii) enforces a JSON output schema, and (iii) explicitly discourages hallucinated fields.

We use a unified prompting framework with consistent JSON output constraints across datasets. While the overall structure of the prompt is shared, minor dataset-specific wording is used when required by annotation schema (e.g., limited field sets in SROIE). Models must infer the relevant keys directly from the input text. This ensures that performance reflects semantic extraction ability rather than memorization of dataset-specific field inventories.

For prompt sensitivity analysis, we evaluate 0–3 in-context examples. Few-shot examples are drawn from the training split of the same dataset but are strictly disjoint from evaluation samples.

3.4 Output Canonicalization

Model outputs frequently deviate from valid JSON formatting, especially under noisy inputs. To enable consistent evaluation, we apply a deterministic, non-semantic output parsing and normalization procedure.

We first attempt to parse a valid JSON object from the raw model output. If this fails, we apply a lightweight output-recovery step that only reads already-generated ”key” and ”value” fields from the model response. This step operates solely on the generated text, does not inspect the source document again, and does not introduce new content, semantic inference, rule-based extraction, or task-specific heuristics.

If no structured output can be recovered, the prediction is treated as empty.

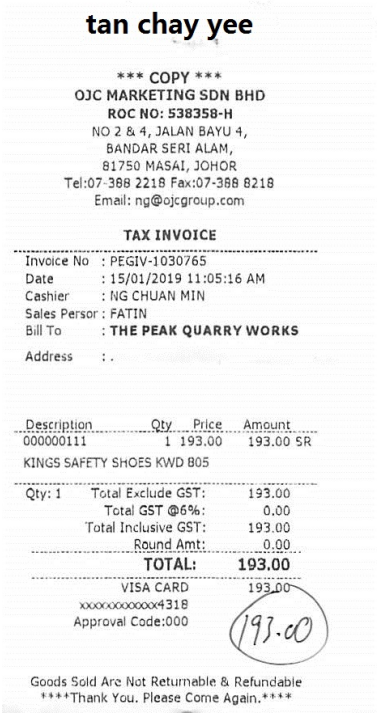
This design ensures that evaluation reflects model generation behavior rather than post-processing heuristics.

Importantly, no post-processing step extracts information from the input document; all recovered key–value pairs originate strictly from the model-generated output.

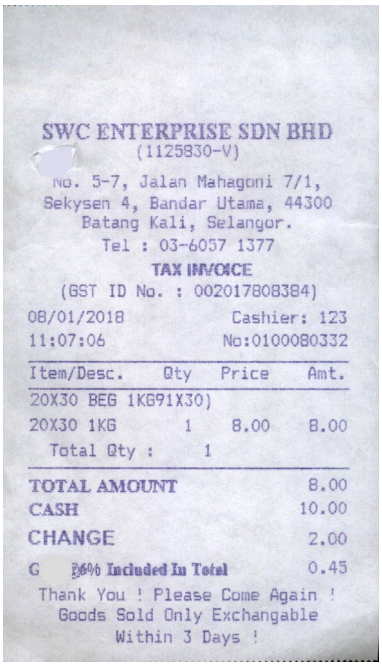
3.5 Datasets and Splits

We evaluate on three standard document understanding benchmarks:

- **FUNSD** (forms with key–value annotations),
- **CORD** (long, itemized receipts),
- **SROIE** (structured receipts with a limited field schema).



(a) Receipt sample 1



(b) Receipt sample 2

(c) Receipt sample 3

Figure 3: Representative SROIE receipts illustrating simpler structure but persistent OCR variability (digit corruption, token merging, formatting noise), which affects numeric and address extraction accuracy.

We use the official test splits for FUNSD and CORD. For SROIE, we evaluate on the full dataset due to the absence of an official split. These datasets collectively capture diverse structural properties, including spatially dependent forms and long, repetitive receipts.

3.6 Input Regimes

Each model is evaluated under two input regimes:

Gold-text: text derived from ground-truth annotations, representing an upper bound without OCR errors.

OCR-text: text generated using PaddleOCR, EasyOCR, and Tesseract, capturing varying levels of realistic noise.

This separation allows us to isolate semantic extraction performance from OCR robustness and quantify the effect of input degradation.

3.7 Inference Settings

All models are evaluated under identical deterministic decoding settings: greedy decoding, no sampling, and a maximum generation length of 256 tokens.

Models are invoked using their standard chat templates when applicable. No model-specific prompt tuning, decoding adjustments, or reranking is applied. This ensures that differences in performance arise from model behavior rather than inference-time optimization.

3.8 Implementation Details

All experiments are implemented using the `transformers` library within a unified pipeline. Each document is processed independently, and predictions are stored in JSONL format.

All components of the pipeline—prompting, decoding, and evaluation—are fixed across models and input regimes, enabling controlled and reproducible comparison.

Evaluation metrics and matching procedures are described in Section 4.

4 Evaluation Methodology

This section defines the evaluation protocol used to assess key–value pair (KVP) extraction performance. Our goal is to measure how accurately models recover annotated fields from document text under both clean and OCR-degraded conditions.

4.1 Evaluation Overview

Each document is provided as plain text, and the model generates a set of predicted key–value pairs in JSON format. Predictions are compared against ground-truth annotations using complementary metrics that capture: (i) coverage of annotated fields, (ii) exact correctness of extracted pairs, and (iii) robustness to textual variation.

The evaluation is performed in a text-only setting without access to layout or visual features, reflecting deployment scenarios where only OCR text is available.

4.2 Task Definition

Let the ground-truth set of key–value pairs be

$$\mathcal{P} = \{(k_i, v_i)\}_{i=1}^{N},$$

and the model prediction be $\hat{\mathcal{P}}$.

Evaluation is restricted to annotated keys in $\mathcal{P}$. Predicted pairs whose keys do not match any ground-truth key are ignored and do not affect the metrics. This design avoids penalizing models for extracting plausible but unannotated fields, while ensuring comparability across models.

4.3 Normalization and Matching

To reduce sensitivity to superficial formatting differences, both keys and values are normalized prior to comparison. Normalization consists of: (i) lowercasing, (ii) trimming leading/trailing whitespace, and (iii) collapsing repeated spaces.

Key Matching. Keys are matched using exact equality after normalization.

Value Alignment. For each matched key, the corresponding predicted and ground-truth values are aligned. Value comparison is performed at the token level using whitespace-based tokenization after normalization.

4.4 Metrics

We report three complementary metrics.

Key Recall. Let $\hat{K}$ denote the set of predicted keys. Key Recall measures the fraction of ground-truth keys that are successfully identified:

$$\text{KeyRecall} = \frac{|\{k \in \mathcal{P} : k \in \hat{K}\}|}{N}.$$

Exact Match (EM). A key–value pair is counted as correct only if both the key and its value match exactly after normalization:

$$\text{EM} = \frac{|\{(k, v) \in \mathcal{P} : (k, v) \text{ exactly matched}\}|}{N}.$$

This provides a strict measure of extraction accuracy.

Value F1. For each matched key, we compute token-level precision, recall, and F1 between predicted and ground-truth values:

$$\text{Precision} = \frac{|V_{\text{pred}} \cap V_{\text{gt}}|}{|V_{\text{pred}}|}, \quad \text{Recall} = \frac{|V_{\text{pred}} \cap V_{\text{gt}}|}{|V_{\text{gt}}|},$$

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The final Value F1 score for a document is computed as the average over all matched keys. Keys that are not predicted contribute zero to the average.

This metric provides partial credit when predicted values capture most of the ground-truth content despite minor formatting differences or OCR-induced noise.

Metric Complementarity. The three metrics capture distinct aspects of performance: Key Recall measures field detection, EM captures strict correctness, and Value F1 measures approximate value recovery under noise.

4.5 Macro Averaging

Dataset-level scores are computed using macro averaging:

$$\text{MacroAverage} = \frac{1}{M} \sum_{i=1}^{M} \text{Metric}(i),$$

where M is the number of documents. This ensures that each document contributes equally, preventing long or densely annotated documents from dominating the results.

4.6 Evaluation Tracks

We evaluate models under two input regimes:

Gold-text track. Inputs consist of clean text derived from ground-truth annotations, providing an upper bound on extraction performance.

Noisy OCR track. Inputs are generated using PaddleOCR, EasyOCR, and Tesseract, introducing realistic errors such as token fragmentation, numeric corruption, and loss of structure.

Comparing these tracks isolates the effect of OCR noise on extraction quality.

4.7 Discussion of Design Choices

Our evaluation design makes two important choices.

First, we ignore predicted keys that do not match annotated fields. This avoids penalizing models for producing additional plausible fields, which is especially important for datasets such as SROIE with limited annotation schemas.

Second, we use token-level Value F1 instead of strict string matching to account for minor OCR-induced variations. This provides a more robust estimate of value recovery while still penalizing substantial deviations.

4.8 Reproducibility

All evaluation results are produced using a unified pipeline with fixed normalization, matching, and scoring procedures. Per-document predictions are stored in JSONL format, enabling full reproducibility and detailed error analysis.

Implementation details are provided in Appendix A.

5 Results

This section presents quantitative and qualitative results for all evaluated models across the three datasets (FUNSD, SROIE, CORD) and the two evaluation tracks: (1) Gold-text and (2) Noisy OCR Text. We report Key Recall, Exact Match (EM), and Value F1 following the protocol in Section 4.

Table 2: Representative supervised document understanding baselines on FUNSD (entity-level F1 on the official test split). These models are fully supervised and leverage layout and/or visual features; they serve as supervised upper-bound anchors rather than direct comparisons.

Model	F1	Source
BERT	0.60	[4]
LayoutLM-base	0.78	[30]
LayoutLMv2-base	0.85	[31]
LayoutLMv3-base	0.89	[8]
LiLT-base	0.81	[28]
BROS-base	0.83	[7]
DocFormer-base	0.86	[2]
FormNet-A2	0.90	[15]
UDOP	**0.92**	[25]

5.1 Overall Trends

Across datasets and input regimes, three broad empirical patterns recur.

Consistently strong 0-shot models. Across the three datasets, Qwen2.5–7B, Mistral–7B, and LLaMA–3 8B are consistently among the strongest 0-shot performers, particularly under Gold-text conditions. In Table 6, Qwen2.5–7B achieves the best 0-shot Gold-text Value F1 on all three datasets, and it also provides the best 0-shot PaddleOCR result on FUNSD, SROIE, and CORD.

Performance degradation under OCR is driven primarily by value corruption and key–value misalignment rather than failure to detect relevant fields. Across all datasets, the transition from Gold-text to OCR text results in lower EM and Value F1. Using the best-performing 0-shot model in each setting (Table 6), Value F1 decreases from 0.6371 to 0.5781 on FUNSD, from 0.9262 to 0.8964 on SROIE, and from 0.9700 to 0.8267 on CORD under PaddleOCR. Under EasyOCR and Tesseract, the decline is substantially larger.

Model gaps narrow as OCR noise increases. Under Gold-text conditions, differences between models are more clearly separated, especially on SROIE and CORD. Under noisier OCR inputs, these gaps become smaller. This pattern is most visible on CORD under Tesseract, where all models fall below 0.30 Value F1.

These trends appear across forms (FUNSD), short receipts (SROIE), and long item-heavy receipts (CORD), despite their differing structural properties.

5.2 Comparison to Prior Work

We compare our benchmark results to representative prior work in two categories: (i) supervised document information extraction models evaluated with entity-level F1, and (ii) LLM/MLLM-based document understanding systems evaluated with ANLS. Because these metrics are not directly interchangeable with our Key Recall / EM / Value F1 protocol, we treat prior work as *performance anchors* rather than as directly comparable numbers.

Supervised layout-aware baselines (entity-level F1). Prior supervised approaches such as the LayoutLM family, UDOP, FormNet, and DocFormer are trained end-to-end on dataset-specific annotations and typically leverage layout cues and document structure [30, 31, 8, 25, 15, 2]. As a result, they provide strong upper bounds under full supervision. Tables 2–4 report representative entity-level F1 results on the official test splits. We use these as ceiling references rather than direct comparisons.

LLM/MLLM baselines reported with ANLS. Recent LLM/MLLM document understanding systems report ANLS, which offers a more comparable anchor for generative systems if we compute ANLS on our own outputs. Table 5 lists

Table 3: Representative supervised baselines on CORD (entity-level F1 on the official test split). These results reflect fully supervised training with layout-aware architectures and serve as upper-bound references.

Model	F1	Source
BERT	0.87	[4]
LayoutLM-base	0.91	[30]
LayoutLMv2-base	0.95	[31]
LayoutLMv3-large	0.97	[8]
DocFormer-base	0.95	[2]
FormNet-A3	0.97	[15]
UDOP	**0.97**	[25]

Table 4: Representative supervised baselines on SROIE (entity-level F1 on the official test split).

Model	F1	Source
BERT	∼0.90	[30]
LayoutLM-base	∼0.94	[30]
LayoutLM-large	∼**0.95**	[30]

representative zero-shot ANLS values reported in LayoutLLM [17]. In the final version, we will also report ANLS for our own models to enable closer comparison between prompt-only text extraction and layout-/vision-aware systems.

Takeaway for comparisons. We interpret supervised entity-level F1 as supervised upper bounds and ANLS as LLM/MLLM anchor metrics. Our primary contribution is not to outperform these systems under their native protocols, but to provide a controlled benchmark of prompt-only LLM extraction across Gold-text and multiple OCR regimes on the official test sets of FUNSD, SROIE, and CORD.

5.3 FUNSD (Gold-Text)

FUNSD Gold-text isolates model ability to infer key–value relationships from clean, human-annotated text without OCR distortion. Results are shown in Table 7.

Observations. On FUNSD Gold-text, the gap between Key Recall and Exact Match remains visible for all models, indicating that retrieving candidate fields is easier than correctly aligning their corresponding values. Qwen2.5–7B achieves the strongest overall results, particularly in Value F1. Gemma–7B and LLaMA–3 8B form the next strongest group, while Gemma–2B and DeepSeek–7B remain substantially lower on all three metrics.

Many remaining errors involve mismatches between question and answer spans, which is consistent with the form-based structure of FUNSD after conversion to plain text.

Effect of few-shot prompting. To study prompt sensitivity under idealized conditions, we vary the number of in-context examples from 0 to 3 for a subset of models. Results are given in Table 8.

Few-shot prompting yields substantial gains on FUNSD Gold-text for both models. LLaMA–3 8B improves steadily from 0-shot to 3-shot across all three metrics. Qwen2.5–7B also improves with additional examples, with its highest Key Recall and EM at 2-shot and highest Value F1 at 3-shot.

Table 9 quantifies OCR severity across the three datasets using Character Error Rate (CER) and Word Error Rate (WER), computed against gold-standard transcriptions. Two patterns are especially important. First, PaddleOCR consistently produces the cleanest transcripts, while Tesseract produces the most severe corruption. Second, receipt-style datasets, especially CORD, exhibit substantially higher error rates than FUNSD. On CORD, Tesseract approaches a WER of 1.0, indicating that nearly every word is either corrupted, deleted, or misplaced.

Table 5: Zero-shot LLM and MLLM baselines reported with ANLS on FUNSD, CORD, and SROIE (official test splits). All values are taken from LayoutLLM and serve as LLM-family anchors.

Model	FUNSD	CORD	SROIE
LLaMA2-7B	0.19	0.15	0.06
LLaMA2-13B	0.23	0.23	0.07
Vicuna-13B	0.26	0.25	0.08
LLaVA-1.5-7B	0.31	0.25	0.08
Qwen-VL-Chat	0.46	0.55	0.35
mPLUG-DocOWL	0.47	0.58	0.37
LayoutLLM-7B	**0.80**	**0.89**	**0.71**

Table 6: Best 0-shot Value F1 per dataset and regime. For each setting, we report the strongest 0-shot model using Gold-text or PaddleOCR (best-performing OCR engine).

Dataset	Regime	Model	Key Recall	EM	Value F1
FUNSD	Gold-text	Qwen2.5 7B	0.2939	0.2249	**0.6371**
	PaddleOCR	Qwen2.5 7B	0.3516	0.1574	0.5781
SROIE	Gold-text	Qwen2.5 7B	0.9950	0.6902	**0.9262**
	PaddleOCR	Qwen2.5 7B	0.9950	0.5684	0.8964
CORD	Gold-text	Qwen2.5 7B	0.8118	0.7724	0.9700
	PaddleOCR	Qwen2.5 7B	0.6345	0.4545	**0.8267**

5.4 FUNSD (Noisy OCR)

Applying OCR to FUNSD introduces distortions absent from the Gold-text annotations. As shown in Table 9, PaddleOCR yields the lowest CER and WER on FUNSD, while EasyOCR and Tesseract produce substantially noisier transcripts. Table 10 summarizes Key Recall, EM, and Value F1 for all OCR–model combinations.

Observations. Under OCR inputs, all FUNSD results are lower than their Gold-text counterparts. Under PaddleOCR, Qwen2.5–7B achieves the strongest results on all three metrics. Under EasyOCR and Tesseract, Qwen2.5–7B remains the strongest Value F1 model, while LLaMA–3 8B attains the highest Key Recall under Tesseract.

The largest changes appear in EM and Value F1, which indicates that OCR noise affects value alignment more strongly than broad field detection. The drop is especially pronounced under EasyOCR and Tesseract.

Few-shot robustness under PaddleOCR. We additionally probe the effect of few-shot prompting on FUNSD PaddleOCR for LLaMA–3 8B and Qwen2.5–7B using Table 11.

Under noisy OCR, few-shot prompting still improves FUNSD performance for both models. LLaMA–3 8B increases steadily from 0-shot to 3-shot across all three metrics. Qwen2.5–7B also improves overall, with a small dip at 2-shot before reaching its highest values at 3-shot.

5.5 SROIE (Gold-text)

Although SROIE contains only four annotated ground-truth fields, real receipts contain many more semantically meaningful fields (e.g., tax, subtotal, payment method), so Key Recall is inherently shaped by the narrow evaluation schema. Results on Gold-text are shown in Table 12.

Observations. On SROIE Gold-text, all strong models achieve very high Key Recall, while EM remains lower than Key Recall across the board. Qwen2.5–7B is the strongest overall model, followed by Mistral–7B and DeepSeek–7B. Gemma–2B remains competitive in Value F1 relative to some larger models, despite lower EM.

Table 7: FUNSD (Gold-text): Key–Value Extraction Results.

Model	Key Recall	EM	Value F1
Gemma 2B	0.0394	0.0183	0.1292
Gemma 7B	0.2475	0.1817	0.5500
DeepSeek 7B	0.1246	0.0586	0.2526
LLaMA–3 8B	0.2729	0.1917	0.4676
Mistral–7B	0.1794	0.1304	0.4135
Qwen2.5 7B	**0.2939**	**0.2249**	**0.6371**

Table 8: Few-shot performance on FUNSD using Gold-text. Results are shown for 0–3 in-context examples.

Model	Shots	Key Recall	EM	Value F1
LLaMA–3 8B	0-shot	0.2729	0.1917	0.4676
	1-shot	0.3974	0.3075	0.6961
	2-shot	0.4052	0.3120	0.7403
	3-shot	**0.4148**	**0.3226**	**0.7504**
Qwen2.5 7B	0-shot	0.2939	0.2249	0.6371
	1-shot	0.3942	0.3040	0.6686
	2-shot	**0.4161**	**0.3152**	0.6820
	3-shot	0.4077	0.3013	**0.7074**

Because SROIE contains only four annotated fields, the reported metrics reflect performance on a narrow schema rather than the full semantic content of each receipt.

Few-shot prompts on SROIE. Because SROIE contains short receipts with relatively stereotyped structure, it provides a useful setting for prompt sensitivity analysis. We compare 0–3-shot prompts for LLaMA–3 8B and Qwen2.5–7B using Gold-text in Table 13.

Few-shot prompting is highly effective on SROIE Gold-text. Both LLaMA–3 8B and Qwen2.5–7B improve substantially over their already strong 0-shot baselines, with near-perfect Key Recall and large gains in both EM and Value F1. Performance remains strong across 1–3 shots for both models.

5.6 SROIE (Noisy OCR)

SROIE OCR noise is substantially more severe than that observed in FUNSD, with frequent issues such as broken numeric fields (e.g., "193.00" → "19300"), corrupted decimal points, and address fragmentation across inconsistent line breaks. As indicated by the SROIE rows in Table 9, even the best OCR engine (PaddleOCR) operates at non-trivial error rates, while EasyOCR and Tesseract introduce substantially more corruption. Table 14 summarizes results across OCR engines.

Observations. Under OCR inputs, SROIE still yields relatively high Key Recall for the strongest models, but EM and Value F1 decrease compared with Gold-text. Qwen2.5–7B achieves the strongest Value F1 under all three OCR engines and the strongest EM under PaddleOCR, EasyOCR, and Tesseract. Mistral–7B and DeepSeek–7B remain competitive across engines.

The reduction from Gold-text to PaddleOCR is smaller on SROIE than on FUNSD or CORD, but the decline under EasyOCR and Tesseract remains clearly visible.

5.7 CORD (Gold-text)

CORD presents a more challenging scenario due to long receipts, dense item-level content, and repeated numerical patterns. Models frequently confuse item-level amounts with global totals. Results on Gold-text are shown in

Table 9: OCR error rates (Character Error Rate, CER, and Word Error Rate, WER) for FUNSD, SROIE, and CORD test sets. Lower values indicate better OCR quality. PaddleOCR consistently yields the lowest error rates across all datasets, while Tesseract produces the noisiest transcripts, particularly on long receipts (CORD).

Dataset	OCR Engine	Mean CER	Mean WER
FUNSD	PaddleOCR	**0.4199**	**0.5675**
	EasyOCR	0.4974	0.7318
	Tesseract	0.5485	0.7197
SROIE	PaddleOCR	**0.3099**	**0.4577**
	EasyOCR	0.4145	0.7148
	Tesseract	0.4334	0.6615
CORD	PaddleOCR	**0.4844**	**0.6199**
	EasyOCR	0.5044	0.8519
	Tesseract	0.8031	0.9825

Table 15. For CORD few-shot prompting, the in-context examples were drawn from the training split using receipts `receipt_00760`, `receipt_00730`, and `receipt_00710`.

Observations. On CORD Gold-text, Value F1 is higher than on FUNSD or SROIE for all strong models. Qwen2.5–7B achieves the strongest overall results, followed by LLaMA–3 8B. Gemma–2B remains well below the larger models, while Gemma–7B, Mistral–7B, and DeepSeek–7B form a middle group.

The gap between Key Recall and EM remains present, indicating that some errors still arise from assigning values to the wrong receipt fields, especially in the presence of repeated numeric content.

Few-shot prompting on CORD (Gold-text). We further examine prompt sensitivity on CORD Gold-text for Qwen2.5–7B and LLaMA–3 8B in Table 16.

On CORD Gold-text, few-shot prompting improves both models relative to 0-shot. The largest gains occur between 0-shot and 1-shot, after which performance saturates. Qwen2.5–7B remains slightly stronger than LLaMA–3 8B across most settings, although the gap narrows substantially under few-shot prompting.

5.8 CORD (Noisy OCR)

We next evaluate CORD under realistic OCR conditions using EasyOCR, PaddleOCR, and Tesseract. As shown in Table 9, CORD exhibits the highest OCR error rates among the three datasets, particularly under Tesseract. Results are summarized in Table 17.

Few-shot prompting under PaddleOCR (CORD). On CORD PaddleOCR text, few-shot prompting yields moderate but noticeable gains over the already strong 0-shot baselines. For Qwen2.5–7B, Value F1 improves from 0.8267 to 0.8618 at 2-shot, while LLaMA–3 8B rises from 0.7547 to 0.8489 at 3-shot.

Observations. CORD under OCR yields the lowest results among the three datasets under the same OCR engines, especially under Tesseract. Qwen2.5–7B remains the strongest model across all three OCR settings, with LLaMA–3 8B usually closest under EasyOCR and PaddleOCR. Under Tesseract, however, all models fall sharply, and the differences between them narrow.

The gap between PaddleOCR and Tesseract is especially large on CORD. For example, Qwen2.5–7B drops from 0.8267 Value F1 under PaddleOCR to 0.2679 under Tesseract.

5.9 Prompt Sensitivity and Saturation Effects

Across datasets, few-shot prompting provides gains that depend strongly on both document type and input quality. On CORD Gold-text and CORD PaddleOCR, additional examples generally improve performance before saturating. On

Table 10: FUNSD (Noisy OCR Text): Key–Value extraction performance across OCR engines and models. Results are grouped by OCR engine to highlight the impact of OCR quality on downstream LLM-based extraction.

OCR	Model	Key Recall	EM	Value F1
PaddleOCR	Gemma 2B	0.0809	0.0229	0.2192
	Gemma 7B	0.3060	0.1284	0.5357
	Mistral–7B	0.2393	0.1103	0.4933
	Qwen2.5 7B	**0.3516**	**0.1574**	**0.5781**
	DeepSeek 7B	0.1404	0.0416	0.2435
	LLaMA–3 8B	0.3131	0.1362	0.5402
EasyOCR	Gemma 2B	0.0348	0.0032	0.1119
	Gemma 7B	0.2068	0.0385	0.3094
	Mistral–7B	0.2027	0.0320	0.2524
	Qwen2.5 7B	**0.2333**	**0.0368**	**0.3118**
	DeepSeek 7B	0.0925	0.0112	0.0740
	LLaMA–3 8B	0.2138	0.0331	0.3030
Tesseract	Gemma 2B	0.0223	0.0063	0.1019
	Gemma 7B	0.1104	0.0071	0.1477
	Mistral–7B	0.1218	0.0385	0.2408
	Qwen2.5 7B	0.2222	**0.0673**	**0.3656**
	DeepSeek 7B	0.0859	0.0112	0.1751
	LLaMA–3 8B	**0.2341**	0.0478	0.3471

FUNSD, both Gold-text and PaddleOCR settings benefit from few-shot prompting, with the strongest results typically reached at 2–3 shots. On SROIE, few-shot prompting is especially effective and remains stable across 1–3 shots under both Gold-text and PaddleOCR inputs.

These results indicate that the effect of in-context examples depends on both dataset structure and input quality. The gains are largest on the more regular receipt datasets and remain visible, though smaller, under OCR inputs.

5.10 Cross-Dataset Comparison

Comparing behavior across FUNSD, SROIE, and CORD highlights that each dataset stresses a different part of the extraction pipeline. On FUNSD, errors are driven primarily by semantic ambiguity and the loss of layout structure after OCR flattening. Models must infer relationships between loosely related "question" and "answer" spans without spatial cues, which tends to inflate the gap between Key Recall and EM.

In SROIE, noisy vendor names, addresses, and numeric corruption dominate the error profile. Even strong models such as Qwen2.5 and Mistral experience clear drops in EM and Value F1 relative to Gold-text, although the compact and regular receipt schema makes this dataset more amenable to few-shot prompting.

CORD exposes a different failure mode: distinguishing line items from global receipt fields when values repeat across the document. Models often attach the correct key (e.g., `total`) to an item-level amount rather than the global total, producing substantial EM penalties even when Key Recall remains relatively high.

Taken together, these differences show that KVP extraction cannot be adequately assessed on a single benchmark. Forms, short receipts, and long item-heavy receipts reveal distinct semantic and structural failure modes.

Figure 4 aggregates the strongest 0-shot Value F1 achieved in each dataset and regime. Even when selecting the best-performing model per condition, the degradation from Gold-text to OCR remains visible: FUNSD drops by roughly 6 points, SROIE by about 3 points, and CORD by about 14 points under PaddleOCR alone.

5.11 Qualitative Error Analysis

To complement the quantitative results, we present representative qualitative examples that illustrate how model behavior changes across clean-text and OCR-degraded inputs. Table 19 shows selected predictions highlighting both successful

Table 11: Few-shot performance on FUNSD using PaddleOCR text. Results are grouped by model to highlight prompt sensitivity under noisy OCR.

Model	Shots	Key Recall	EM	Value F1
LLaMA–3 8B	0-shot	0.3131	0.1362	0.5402
	1-shot	0.4223	0.1958	0.7176
	2-shot	0.4609	0.2160	0.7393
	3-shot	**0.4692**	**0.2234**	**0.7526**
Qwen2.5 7B	0-shot	0.3516	0.1574	0.5781
	1-shot	0.4216	0.1870	0.6691
	2-shot	0.4252	0.1846	0.6653
	3-shot	**0.4446**	**0.2037**	**0.7032**

Table 12: SROIE (Gold-text): Key–Value extraction performance under clean, human-annotated text. Best results per metric are highlighted.

Model	Key Recall	EM	Value F1
Gemma 2B	0.8098	0.4006	0.8581
Gemma 7B	0.9604	0.5468	0.8520
Mistral–7B	0.9914	0.5764	0.8921
Qwen2.5 7B	**0.9950**	**0.6902**	**0.9262**
DeepSeek 7B	0.9878	0.5677	0.8680
LLaMA–3 8B	0.7608	0.4467	0.6926

extraction and common failure modes. These examples correspond directly to the failure categories summarized in Table 20.

Under clean-text conditions, models typically recover key–value pairs with correct alignment and formatting. In these cases, both Exact Match and Value F1 are high, reflecting accurate semantic mapping from input text to structured output.

In contrast, OCR-degraded inputs introduce systematic failure modes that are not fully captured by aggregate metrics. Three recurring patterns are evident: numeric corruption, key–value misalignment, and hallucination or over-extraction.

Numeric corruption. OCR errors frequently distort numeric values, such as decimal points or digit boundaries. These errors often preserve partial token overlap, resulting in moderate Value F1 but complete failure under Exact Match.

Key–value misalignment. Models often identify the correct set of keys but associate them with incorrect values. This is particularly common in form-like documents such as FUNSD, where spatial relationships between fields are lost in flattened text.

Hallucination and over-extraction. Models sometimes generate additional fields that are not supported by the input text, especially in receipt-style documents. These hallucinated outputs reflect implicit schema priors induced by the prompt or pretraining data.

More broadly, these examples illustrate that OCR noise does not merely reduce value accuracy, but fundamentally alters the structure of the extraction task. Errors in token integrity, line ordering, and grouping disrupt the cues that models rely on for semantic alignment.

A manual examination of several hundred model predictions across FUNSD, SROIE, and CORD reveals systematic failure patterns that explain much of the remaining gap between LLM extraction and specialized document models.

One prominent failure mode is *hallucination*, where models generate keys or values not present in the document. This is most common on receipts, where models sometimes invent fields such as “subtotal” or “invoice number” based on prior expectations. Hallucinations introduce false positives and directly degrade EM.

A second recurring issue is *key–value misalignment*. Models often identify the correct key inventory but attach incorrect values, particularly in dense numerical regions such as tax, total, and item-price sections. Without layout infor-

Table 13: Few-shot prompting on SROIE (Gold-text). Results for LLaMA–3 8B and Qwen2.5–7B under 0–3-shot prompting.

Model	Shots	Key Recall	EM	Value F1
LLaMA–3 8B	0-shot	0.7608	0.4467	0.6926
	1-shot	**0.9993**	0.7111	0.9481
	2-shot	0.9978	**0.7795**	**0.9668**
	3-shot	0.9935	0.7630	0.9586
Qwen2.5 7B	0-shot	0.9950	0.6902	0.9262
	1-shot	0.9914	0.7320	0.9534
	2-shot	**0.9950**	0.7608	0.9592
	3-shot	0.9921	**0.7644**	**0.9625**

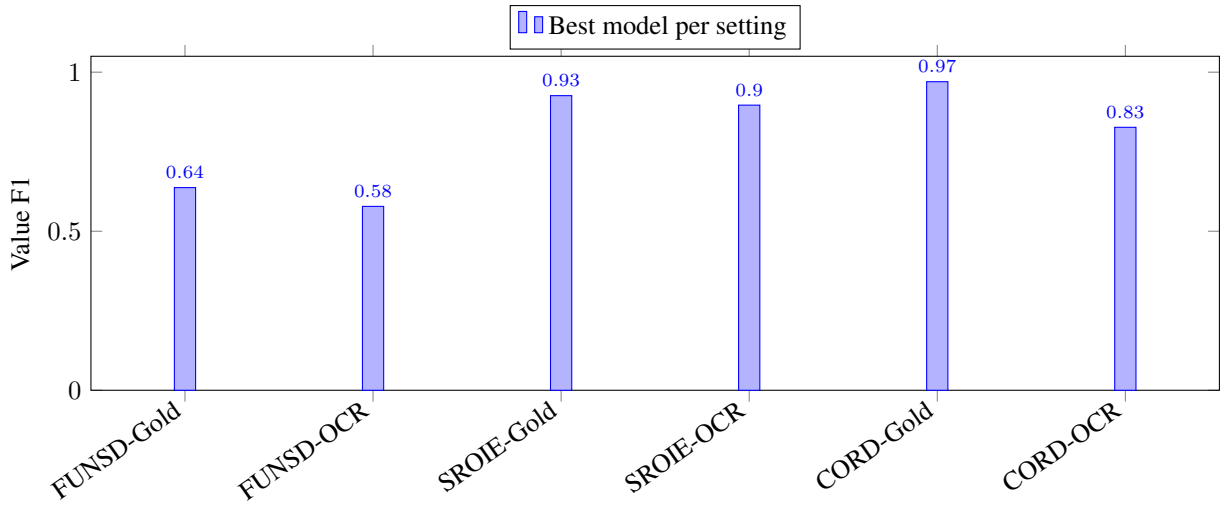


Figure 4: Best 0-shot Value F1 achieved in each dataset and regime. For each bar, we report the strongest 0-shot model under Gold-text or PaddleOCR. The Gold-to-OCR gap is smallest on SROIE and largest on CORD, highlighting that OCR degradation affects datasets differently depending on structural complexity and numeric density.

mation, models appear to rely on local textual proximity or distributional regularities that are not always semantically correct.

A third class of errors involves *address fragmentation*. Multi-line addresses are often truncated, split, or merged with neighboring text, reducing Value F1 and harming downstream usability. OCR artifacts make this especially difficult by breaking line structure.

Models also exhibit *over-extraction*, especially on long receipts, where they attempt to assign labels to nearly every number in the document. Finally, a non-trivial fraction of errors arise from *JSON formatting problems*, including unbalanced braces, multiple JSON blocks, or explanatory text mixed with structured output.

As summarized in Figure 5, these errors can be grouped into five interrelated classes. Hallucinations and address fragmentation are closely tied to OCR artifacts and missing layout cues, while misalignment connects directly to downstream over-extraction and JSON failures.

5.12 Summary of Results

Across datasets, strong 0-shot performance is observed under Gold-text conditions, especially for Qwen2.5–7B and, in several settings, Mistral–7B and LLaMA–3 8B. Under OCR inputs, EM and Value F1 decline across all datasets, with the largest absolute drop occurring on CORD and the smallest on SROIE under PaddleOCR.

Few-shot prompting generally improves results on all three datasets, although the magnitude of improvement varies

Table 14: SROIE: Key–Value Extraction performance across OCR engines. For each OCR regime, the strongest scores among evaluated models are highlighted.

OCR	Model	Key Recall	EM	Value F1
PaddleOCR	Gemma 2B	0.8112	0.3408	0.8189
	Gemma 7B	0.9733	0.4986	0.8413
	Mistral–7B	0.9733	0.4733	0.8614
	Qwen2.5 7B	**0.9950**	**0.5684**	**0.8964**
	DeepSeek 7B	0.9921	0.5014	0.8586
	LLaMA–3 8B	0.7385	0.3696	0.6655
EasyOCR	Gemma 2B	0.7378	0.1693	0.6146
	Gemma 7B	0.9712	0.2983	0.6955
	Mistral–7B	0.9683	0.3336	0.7370
	Qwen2.5 7B	**0.9957**	**0.4229**	**0.7720**
	DeepSeek 7B	0.9719	0.2990	0.6999
	LLaMA–3 8B	0.9114	0.3631	0.7222
Tesseract	Gemma 2B	0.7903	0.2262	0.6944
	Gemma 7B	0.9460	0.3617	0.7698
	Mistral–7B	0.9416	0.3559	0.7780
	Qwen2.5 7B	0.9402	**0.4280**	**0.8076**
	DeepSeek 7B	**0.9611**	0.3725	0.7627
	LLaMA–3 8B	0.7803	0.3213	0.6554

Table 15: CORD (Gold-text): Key–Value Extraction Results.

Model	Key Recall	EM	Value F1
Gemma 2B	0.2735	0.2312	0.6286
Gemma 7B	0.6057	0.5385	0.7800
Mistral–7B	0.6263	0.4752	0.7885
DeepSeek 7B	0.5962	0.4322	0.7489
LLaMA–3 8B	0.7796	0.6631	0.8900
Qwen2.5 7B	**0.8118**	**0.7724**	**0.9700**

by dataset and input regime. The strongest few-shot gains are observed on SROIE and FUNSD, while CORD shows steadier but smaller improvements before saturation.

6 Discussion

Our benchmark reveals that text-only LLM-based key–value pair (KVP) extraction is governed by a two-stage bottleneck: (i) semantic reasoning over textual content, and (ii) preservation of that content under OCR-induced corruption. While modern instruction-tuned models perform strongly in the first stage, their overall effectiveness in realistic pipelines is ultimately constrained by the second. This distinction clarifies why clean-text evaluations often overestimate real-world performance.

6.1 Separation Between Semantic Capacity and Input Fidelity

Under Gold-text conditions, several models achieve high Key Recall and Value F1 across all datasets, indicating that LLMs can recover a large fraction of annotated fields without task-specific supervision or layout information. This suggests that, when textual fidelity is preserved, extraction errors arise primarily from alignment and normalization rather than from insufficient semantic capacity.

Table 16: Few-shot prompting on CORD (Gold-text). Each model is listed once, with performance shown for 0–3 in-context examples.

Model	Shots	Key Recall	EM	Value F1
Qwen2.5 7B	0-shot	0.8118	0.7724	0.9700
	1-shot	0.8213	0.7992	0.9850
	2-shot	**0.8247**	**0.8035**	**0.9880**
	3-shot	0.8216	0.8004	**0.9880**
LLaMA–3 8B	0-shot	0.7796	0.6631	0.8900
	1-shot	0.8051	0.7280	0.9556
	2-shot	0.8262	**0.7539**	**0.9856**
	3-shot	**0.8276**	0.7453	**0.9856**

Table 17: CORD: Key–Value Extraction Performance Across OCR Engines. Each OCR engine is listed once, with results shown for all evaluated models.

OCR	Model	Key Recall	EM	Value F1
EasyOCR	Gemma 2B	0.1244	0.0586	0.2567
	Gemma 7B	0.4023	0.1584	0.4617
	Mistral–7B	0.3669	0.1666	0.4160
	Qwen2.5 7B	0.4619	**0.2507**	**0.5594**
	LLaMA–3 8B	**0.4800**	0.1822	0.4978
PaddleOCR	Gemma 2B	0.1769	0.1203	0.4600
	Gemma 7B	0.5008	0.3040	0.6928
	Qwen2.5 7B	**0.6345**	**0.4545**	**0.8267**
	DeepSeek 7B	0.4496	0.2789	0.6000
	LLaMA–3 8B	0.6307	0.3817	0.7547
Tesseract	Gemma 2B	0.1171	0.0452	0.1700
	Gemma 7B	0.1649	0.0519	0.1553
	Qwen2.5 7B	**0.2462**	**0.1187**	**0.2679**
	DeepSeek 7B	0.1929	0.0695	0.2087
	LLaMA–3 8B	0.2261	0.0863	0.2288

However, this regime does not reflect typical deployment settings. Once OCR noise is introduced, the dominant source of error shifts away from reasoning and toward input corruption. This transition highlights a key limitation of text-only pipelines: strong semantic modeling cannot compensate for degraded or incomplete input signals.

6.2 OCR Noise Alters the Structure of the Task

The effect of OCR noise extends beyond simple degradation in accuracy. It fundamentally changes the structure of the extraction problem. Under clean text, models operate on well-formed sequences with clear lexical boundaries. Under OCR inputs, token fragmentation, numeric corruption, and loss of ordering disrupt these cues.

This shift is reflected in the disproportionate drop in EM and Value F1 relative to Key Recall. Models often continue to identify relevant fields (high recall), but fail to correctly align keys with their corresponding values. In this sense, OCR noise converts the problem from semantic extraction to a more fragile alignment task.

Importantly, the impact of this shift is dataset-dependent. FUNSD suffers from loss of spatial relationships, SROIE from numeric corruption, and CORD from compounded ambiguity in dense numeric regions. These differences indicate that OCR noise interacts with document structure, rather than acting as uniform random corruption.

Table 18: Few-shot prompting on CORD using PaddleOCR text. Each model is listed once, with performance shown for 0–3 shot prompts.

Model	Shots	Key Recall	EM	Value F1
Qwen2.5 7B	0-shot	0.6345	**0.4545**	0.8267
	1-shot	**0.6548**	0.4543	0.8485
	2-shot	0.6370	0.4534	**0.8618**
	3-shot	0.6327	0.4301	0.8117
LLaMA–3 8B	0-shot	**0.6307**	0.3817	0.7547
	1-shot	0.5993	0.3946	0.8061
	2-shot	0.6170	0.4153	0.8093
	3-shot	0.6166	**0.4237**	**0.8489**

Table 19: Qualitative examples of LLM-based KVP extraction across datasets and input regimes. We show representative success and failure cases, highlighting the impact of OCR noise and structural ambiguity.

Input (Excerpt)	Ground Truth	Prediction (Qwen2.5–7B)	Notes
FUNSD (Gold-text) "Date: 12/10/98 To: George Baroody"	{DATE: 12/10/98, TO: George Baroody}	{DATE: 12/10/98, TO: George Baroody}	Correct extraction (high EM and Value F1)
SROIE (OCR) "TOTAL: 19300 DATE: 09/12/98"	{TOTAL: 193.00, DATE: 09/12/98}	{TOTAL: 19300, DATE: 09/12/98}	Numeric corruption (EM fails, Value F1 partially preserved)
FUNSD (OCR) "Invoice Date Customer Name: John"	{DATE: Invoice Date, NAME: John}	{DATE: John, NAME: John}	Key–value misalignment (correct keys, wrong pairing)
CORD (OCR) "Total: 25.30 Item: Coffee 5.00"	{TOTAL: 25.30}	{TOTAL: 25.30, SUBTOTAL: 5.00}	Hallucination / over-extraction

6.3 Scaling Effects Diminish Under Noisy Inputs

Increasing model size improves performance under Gold-text conditions, but these gains diminish as OCR noise increases. Under noisy inputs, model rankings compress and differences between architectures become less pronounced.

This pattern suggests that scaling primarily enhances semantic reasoning, but provides limited benefit when the underlying signal is corrupted. Larger models can partially compensate for minor inconsistencies, but cannot reliably recover information that has been removed or distorted upstream. As a result, model capacity alone is insufficient to address the challenges posed by OCR-heavy pipelines.

6.4 Prompting Improves Performance Within Structural Limits

Few-shot prompting consistently improves performance, but its effectiveness is bounded by input quality and dataset structure. Gains are strongest on SROIE, where documents are short and the field inventory is limited, and more moderate on FUNSD and CORD, where structural ambiguity is higher.

This suggests that in-context examples primarily help models exploit regular patterns in the data rather than recover missing structure. When OCR noise disrupts key spans or relationships, additional examples provide diminishing returns. Thus, prompting is most effective when the task is ambiguous but the input remains structurally intact.

6.5 Structural Limitations of Text-Only Pipelines

A central limitation of this benchmark is the reliance on flattened text. Many document understanding tasks depend on spatial relationships, hierarchical grouping, and visual context, which are lost when documents are converted to linear

Table 20: Common failure modes in LLM-based KVP extraction.

Failure Type	Description
Hallucinated fields	Plausible but unsupported keys or values are generated, often influenced by prompt priors.
Key–value misalignment	Correct keys are detected but paired with the wrong values, especially in dense numeric regions.
OCR-induced corruption	Broken numbers, fragmented tokens, and missing punctuation lead to incorrect or partial value recovery.
Address fragmentation	Multi-line addresses are split, truncated, or merged with neighboring content.
Format instability	Outputs contain malformed JSON, duplicated blocks, or inconsistent field naming.

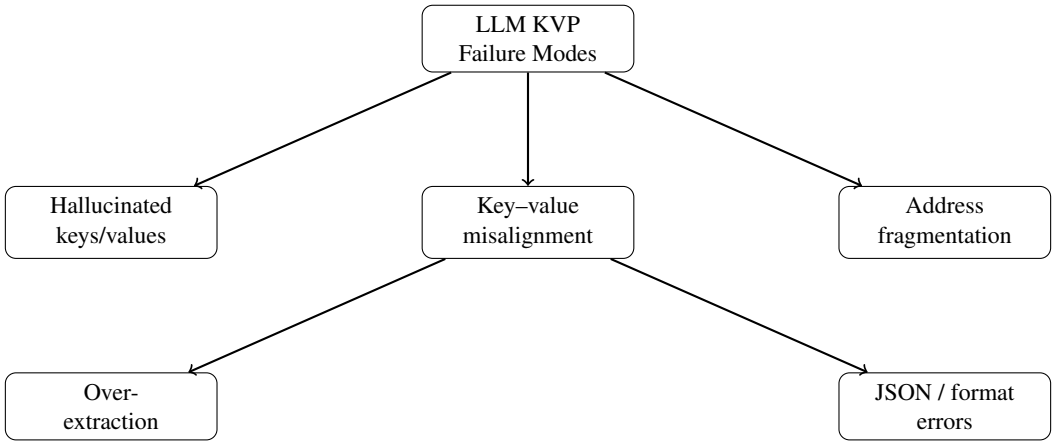


Figure 5: Conceptual taxonomy of recurring failure modes. Hallucinations and address fragmentation are strongly tied to OCR corruption and missing layout cues, while misalignment, over-extraction, and JSON failures reflect limitations in structural reasoning and output control.

text sequences.

The observed failure modes—key–value misalignment, hallucination, address fragmentation, and format instability—are consistent with this limitation. These errors often arise not from incorrect reasoning, but from insufficient structural cues in the input representation. This highlights a fundamental gap between text-only LLM pipelines and layout-aware or multimodal approaches.

6.6 Positioning Relative to Layout-Aware Models

The comparison with prior work suggests that text-only LLMs and layout-aware models address complementary aspects of document understanding. Layout-aware models leverage explicit spatial information and supervised training to achieve high accuracy on structure-sensitive tasks. In contrast, LLMs offer flexibility, schema generalization, and strong zero-shot performance.

Rather than viewing these approaches as competing paradigms, the results suggest that they are best understood as complementary components. Hybrid systems that combine OCR, layout modeling, and LLM-based semantic reasoning may offer the most robust solution for real-world document extraction.

6.7 Evaluation and Dataset Effects

The benchmark also highlights that evaluation outcomes depend strongly on dataset design. In SROIE, the limited annotation schema constrains the maximum achievable recall and does not reward extraction of additional valid fields. In FUNSD and CORD, structural complexity introduces ambiguity that is not fully captured by text-only representations.

Similarly, metric choice influences interpretation. EM is highly sensitive to formatting deviations, while Value F1 provides a more tolerant measure under OCR noise but still depends on token-level overlap. These factors suggest that evaluation metrics should be interpreted in the context of both dataset characteristics and input quality.

6.8 Implications for Deployment

From a practical perspective, these findings suggest that the reliability of LLM-based extraction depends more on upstream data quality than on model choice alone. Text-only LLM pipelines are well suited to settings with clean or born-digital text, where semantic reasoning dominates. In contrast, OCR-heavy pipelines require additional mechanisms for error correction, structural recovery, or validation.

This does not diminish the utility of LLMs, but clarifies their role within a larger system. They are most effective as semantic interpreters operating on high-quality inputs, rather than as standalone solutions for noisy document processing.

6.9 Key Takeaways

Three conclusions emerge. First, modern LLMs exhibit strong semantic extraction capability under clean-text conditions. Second, OCR noise fundamentally alters the extraction task and becomes the dominant source of error in realistic settings. Third, improvements from scaling and prompting are conditional on the availability of reliable input signals.

Overall, progress in document understanding will require not only advances in LLMs, but also improvements in OCR robustness, structural representation, and evaluation protocols that reflect real-world conditions.

7 Conclusion and Future Work

We presented a controlled benchmark of open-source instruction-tuned large language models for key–value pair (KVP) extraction under both clean-text and OCR-degraded input conditions. By evaluating representative decoder-only models (Gemma, Mistral, Qwen2.5, LLaMA 3, and DeepSeek) across FUNSD, CORD, and SROIE using a unified protocol, this work isolates the interaction between semantic modeling capacity and input quality in text-only extraction pipelines.

Our results demonstrate that modern instruction-tuned LLMs possess strong semantic extraction capabilities when the textual signal is preserved. Under Gold-text conditions, models recover a large fraction of annotated fields without task-specific training, indicating that prompt-based inference alone is often sufficient for structured extraction in clean settings. However, this capability does not fully transfer to realistic scenarios.

Across all datasets, OCR-induced corruption consistently reduces Exact Match and Value F1, with the largest degradation observed in structurally complex and numerically dense documents such as CORD. More importantly, the results show that OCR noise does not simply lower performance, but shifts the dominant error regime from semantic inference to key–value alignment and value corruption. This establishes a clear two-stage bottleneck: while LLMs are effective semantic interpreters, their performance is ultimately constrained by the fidelity of the input text.

We further observe that scaling and prompting provide conditional improvements. Larger models yield stronger results when the input signal remains intact, but their relative advantage diminishes as OCR noise increases. Similarly, few-shot prompting improves performance across datasets, but its effectiveness depends on the availability of consistent structural patterns and saturates quickly in noisy settings. These findings suggest that neither scaling nor prompting alone can overcome upstream degradation.

From a system perspective, these results position text-only LLM extraction as a powerful but incomplete solution for document understanding. While these models offer strong zero- and few-shot generalization, they lack explicit access to spatial structure and remain sensitive to OCR-induced errors. As a result, robust real-world pipelines will likely require integration with complementary components rather than relying on prompt-based extraction in isolation.

A central contribution of this work is reproducibility. We introduce a standardized evaluation framework with unified prompting, deterministic decoding, non-semantic output canonicalization, and consistent benchmarking across datasets and OCR engines. This enables controlled comparison across models and input regimes and provides a foundation for future research on robust document extraction.

Future Directions. Our findings suggest several promising directions for further work.

Hybrid document understanding systems. Combining layout-aware encoders or vision–language models with LLM-based semantic decoders may provide a principled way to integrate structural and semantic information, improving robustness on complex documents.

Robustness to OCR degradation. Improving performance under noisy inputs remains a key challenge. Future work could explore OCR post-correction, numeric repair, structure recovery, and noise-aware adaptation strategies that explicitly account for corrupted inputs.

Structure-aware prompting and decoding. While few-shot prompting is effective, its benefits are bounded. Incorporating schema constraints, structured decoding, or output regularization may improve stability without relying on heuristic post-processing.

Evaluation beyond clean benchmarks. Current benchmarks do not fully capture deployment conditions. Expanding annotation schemas, incorporating more diverse document types, and evaluating under controlled degradation levels would enable more realistic assessment of document extraction systems.

Bridging clean and noisy regimes. A more systematic understanding of how performance degrades with input quality could help align model improvements with real-world gains, particularly in OCR-heavy pipelines.

Final Remark. Overall, instruction-tuned LLMs already provide strong semantic extraction capabilities when the input signal is intact. Achieving reliable document understanding in practice, however, will require systems that jointly address language, structure, and input uncertainty, rather than treating these factors in isolation.

A Hyperparameters and Implementation Details

This appendix provides implementation details necessary to ensure full reproducibility of our experiments.

A.1 Model Configurations

All models are evaluated using publicly available HuggingFace checkpoints:

- `google/gemma-2b-it`
- `google/gemma-7b-it`
- `mistralai/Mistral-7B-Instruct-v0.2`
- `Qwen/Qwen2.5-7B-Instruct`
- `deepseek-ai/deepseek-llm-7b-chat`
- `meta-llama/Meta-Llama-3-8B-Instruct`

All models are used in their instruction-tuned variants without any fine-tuning or parameter updates.

A.2 Decoding Parameters

All models are evaluated under identical deterministic decoding settings:

- Decoding strategy: greedy decoding
- Maximum generation length: 256 tokens
- Temperature: 0.0
- Top-p: disabled
- Sampling: disabled

These settings eliminate stochastic variation and ensure consistent outputs across runs.

A.3 Prompt Template

All models are evaluated using a unified prompt template consisting of:

- Task instruction defining key–value extraction
- Explicit JSON output requirement
- Anti-hallucination constraint
- Optional few-shot examples (1–3)

The same prompt structure is applied across all models and datasets to ensure fair comparison. The full prompt template and examples are provided in the accompanying code repository.

A.4 OCR Engines

We evaluate three OCR systems:

- PaddleOCR (primary evaluation)
- EasyOCR
- Tesseract

OCR outputs are used directly without any correction, filtering, or post-processing.

A.5 Evaluation Pipeline

All predictions are processed using a deterministic JSON-first output parsing pipeline. We first attempt JSON parsing of the model response; if this fails, we recover only explicitly generated "key" / "value" fields from the response text itself. No rule-based extraction from the source document, heuristic completion, semantic correction, or post-hoc field inference is applied. Thus, all extracted content originates from the model output.

Metrics (Key Recall, Exact Match, Value F1) are computed using a shared normalization and matching framework across all datasets.

A.6 Implementation Details

All experiments are implemented using the `transformers` library within a unified inference pipeline.

Each document is processed independently, and model outputs are stored in JSONL format to support reproducibility and downstream analysis.

A.7 Hardware Setup

All experiments are conducted on a single-GPU environment (e.g., Colab Pro+). Models in the 7–8B parameter range are evaluated sequentially to ensure consistent resource usage.

For larger models, CPU offloading is used when necessary, which may increase inference latency but does not affect output quality.

References

[1] AI@Meta. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[2] Srikar Appalaraju et al. Docformer: End-to-end transformer for document understanding. In *ICCV*, 2021.

[3] DeepSeek-AI et al. Deepseek llm. *arXiv preprint arXiv:2401.02954*, 2024.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[5] Yuning Du et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

[6] Jaekyu Ha, Robert M. Haralick, and Ihsin T. Phillips. Document analysis and understanding for paper-based forms. In *Document Analysis Systems Workshop*, 2000.

[7] Teakgyu Hong et al. Bros: A pre-trained language model focusing on text and layout for better key information extraction. In *AAAI*, 2022.

[8] Yupan Huang et al. Layoutlmv3: Unified text and image masking for document ai. *arXiv preprint arXiv:2204.08387*, 2022.

[9] Zhenyong Huang, Lei Gao, Long Liu, et al. Icdar 2019 competition on scanned receipt ocr and information extraction. In *ICDAR Workshops*, 2019.

[10] Jaided AI. Easyocr. https://github.com/JaidedAI/EasyOCR, 2020.

[11] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *ICDAR Workshops*, 2019.

[12] Albert Q. Jiang et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[13] Anoop R. Katti et al. Chargrid: Understanding 2d documents. In *EMNLP*, 2018.

[14] Geewook Kim et al. Ocr-free document understanding transformer. In *ECCV*, 2022.

[15] Jaewon Lee et al. Formnet: Structural encoding for form understanding. In *EMNLP*, 2021.

[16] Yiheng Li et al. Structurallm: Structural pre-training for form understanding. *arXiv preprint arXiv:2105.11210*, 2021.

[17] Chuwei Luo et al. Layoutllm: Layout instruction tuning for document understanding. *arXiv preprint arXiv:2404.05225*, 2024.

[18] John Miller et al. Transformer sensitivity to noise. *arXiv preprint arXiv:2106.03930*, 2021.

[19] Seunghyun Park, Minhyeok Shin, Bokyung Lee, Junyeop Lee, Jonghwan Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. In *NeurIPS Document Intelligence Workshop*, 2019.

[20] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[21] Michael Shilman, Jie Liang, and Robert Haralick. Structured document image analysis using prior knowledge and heuristics. In *ICDAR*, 2005.

[22] Maayan Shpigel Nacson et al. Docvlm: Efficient vision-language model for documents. *arXiv preprint arXiv:2412.08746*, 2024.

[23] Ray Smith. An overview of the tesseract ocr engine. In *ICDAR*, 2007.

[24] Haitian Sun et al. Llm robustness under noisy text. *arXiv preprint arXiv:2305.13289*, 2023.

[25] Zhengyuan Tang et al. Udop: Unified document processing. *arXiv preprint arXiv:2212.02623*, 2022.

[26] Gemma Team et al. Gemma: Open models based on gemini research. *arXiv preprint arXiv:2403.08295*, 2024.

[27] Xinyu Wang et al. Instructuie: Unified information extraction. *arXiv preprint arXiv:2304.08085*, 2023.

[28] Yucheng Wang et al. Lilt: A simple yet effective language-independent layout transformer. In *ACL*, 2022.

[29] Jason Wei et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[30] Yiheng Xu et al. Layoutlm: Pre-training of text and layout for document understanding. In *KDD*, 2020.

[31] Yiheng Xu et al. Layoutlmv2: Multi-modal pre-training for visually-rich documents. In *ACL*, 2021.