DocKIE-Bench: Document Key Information Extraction Benchmark for Large Language Models

Anonymous ACL submission

Abstract

Document Key Information Extraction (KIE) transforms unstructured or semi-structured documents into structured data, typically key–value pairs or grouped entities, that support enterprise applications such as business workflow automation. While recent work explores the use of Large Language Models (LLMs) for KIE using prompting, rather than document-specific fine-tuning, progress is hindered by the lack of benchmarks tailored to this emerging paradigm. We introduce **DocKIE-Bench**, a benchmark specifically designed to evaluate KIE in the context of LLMs. DocKIE-Bench provides carefully designed schema with detailed descriptions, formats, and examples, covers 38 document types from diverse domains, and includes fine-grained component tags (tables, forms, handwritten regions, and others) that enable nuanced analysis of model performance. We evaluate both proprietary and open-source LLMs and conduct comprehensive ablation studies on schema design and input modality, offering practical insights into current strengths and limitations. The dataset will be publicly available.

1 Introduction

Key Information Extraction (KIE) refers to the task of identifying and extracting structured information, typically in the form of key-value pairs or grouped entities, from unstructured or semi-structured documents. This structured information underpins a wide range of downstream applications such as database population, robotic process automation, and automated business workflows. As such, KIE plays a central role in document understanding pipelines, especially in enterprise and industrial settings (Cui et al., 2021).

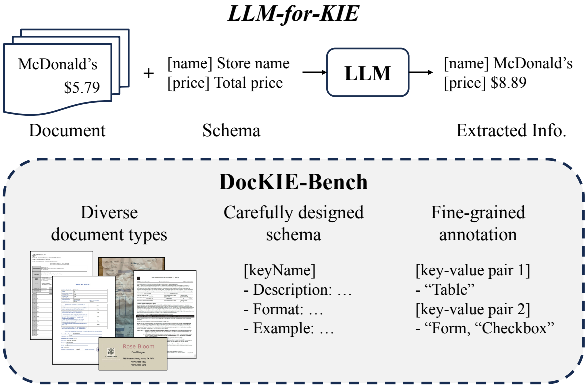

Figure 1: Overview of the *LLM-for-KIE* paradigm and the proposed benchmark, **DocKIE-Bench**. The illustration shows how a LLM performs KIE based on a document and a schema, and how DocKIE-Bench supports this process through diverse document types, carefully designed schemas, and fine-grained annotations.

With the emergence of Large Language Models (LLMs), there has been growing interest in leveraging their strong generalization capabilities for KIE tasks across diverse document types. While earlier fine-tuning-based approaches (Xu et al., 2020, 2021; Li et al., 2021; Hong et al., 2022; Wang et al., 2022; Kim et al., 2022; Huang et al., 2022) demonstrated effectiveness in controlled experimental settings, they face significant scalability challenges. These methods require substantial effort in data curation and model retraining whenever new document types are introduced, limiting their practicality in dynamic real-world environments.

To overcome these limitations, recent efforts have shifted toward the *LLM-for-KIE* paradigm (He et al., 2023; Perot et al., 2023; Liao et al., 2024; Zhang et al., 2025), where key information is extracted using an LLM without document-specific fine-tuning. As illustrated in Figure 1, an LLM is prompted with a document and information about entity keys for extraction (referred to as schema), and returns the corresponding values in a structured, machine-readable format such as JSON. Despite the promise of this paradigm, the lack of standardized benchmarks tailored for evaluating LLMs in KIE settings hinders rigorous assessment and meaningful comparison across approaches.

In the context of the *LLM-for-KIE* paradigm, a benchmark designed for LLM-based KIE must account for several factors that are largely overlooked in prior benchmarks. First, as the paradigm shifts away from document-specific training, the information provided in the schema has to be both clear and unambiguous, extending beyond mere key names to include descriptions or contextual cues. In earlier fine-tuning-based settings, where the test distribution closely matched the training data, models could learn to resolve ambiguities between key names and values during training, making such precision less critical. However, in the training-free setting of *LLM-for-KIE*, the clarity and specificity of the schema are essential for guiding the model and enabling meaningful evaluation of extraction performance.

Secondly, existing benchmarks typically target a narrow range of document types, often focusing on a single document type, under the assumption that models are fine-tuned for each specific extraction task. In contrast, the *LLM-for-KIE* paradigm eliminates the need for task-specific training, shifting the focus towards generalization across heterogeneous document structures and formats. As such, a benchmark for this setting should prioritize broad coverage of diverse document types to better assess the robustness and adaptability of LLM-based extractors in real-world scenarios.

Lastly, given the training-free nature of LLMs and their growing applicability in KIE tasks, there is a need for benchmarks that go beyond coarse-grained accuracy metrics. In particular, incorporating fine-grained annotations that specify, for each entity key, the type of document component from which the value is extracted (e.g., table, form field, checkbox, handwritten region) allows for more detailed evaluation. These component-level annotations provide an additional analytical dimension, enabling more insightful diagnostics of model behavior under the *LLM-for-KIE* paradigm. Rather than treating performance as a single aggregated score, such granularity reveals strengths and weaknesses of a model across different content types, offering a more comprehensive understanding of its generalization capabilities.

In view of the above considerations, we propose **DocKIE-Bench**, a novel benchmark specifically designed to evaluate LLM-based KIE systems. DocKIE-Bench features clearly defined schemas based on our proposed schema design, which includes a description, format specifications, and examples for each entity across a wide range of document types, enabling precise and unambiguous evaluation of LLM extraction capabilities. By incorporating a diverse set of document types, the benchmark supports robust assessment of model generalization. Additionally, it includes fine-grained annotations of document components, such as tables, forms, and handwritten texts, allowing for detailed analysis of model performance across different component types.

Using DocKIE-Bench, we conduct extensive experiments with both proprietary and open-source LLMs to assess their effectiveness on the KIE task. We also perform ablation studies to systematically examine the impact of key factors such as schema design and input modality. These studies provide deeper insight into model behavior, strengths, and limitations within the *LLM-for-KIE* setting.

Our contributions are summarized as follows:

- We introduce DocKIE-Bench, a benchmark tailored for evaluating KIE capabilities in LLMs with schema-guided structured outputs.

- We curate a diverse collection of documents spanning various domains and formats, with fine-grained annotations linking extracted values to specific document components (e.g., tables, forms, handwritten regions).

- We conduct extensive experiments with proprietary and open-source LLMs, including ablation studies, to provide practical insights into LLM behavior and guide future KIE system development.

2 Related Works

2.1 Benchmarks for KIE

Several benchmarks have been proposed to evaluate KIE systems, but they are mostly designed for fine-tuned models and lack critical features necessary for the emerging *LLM-for-KIE* paradigm.

Limited Schema and Document Diversity. Early benchmarks such as SROIE (Huang et al., 2019) and FUNSD (Jaume et al., 2019) adopt simple schemas with flat entity structures (i.e., no grouped entities) and focus on narrow document types like receipts and form-like documents. CORD (Park et al., 2019) introduces grouped entities and brief textual descriptions of entities, but remains limited to receipts. More recent datasets such as

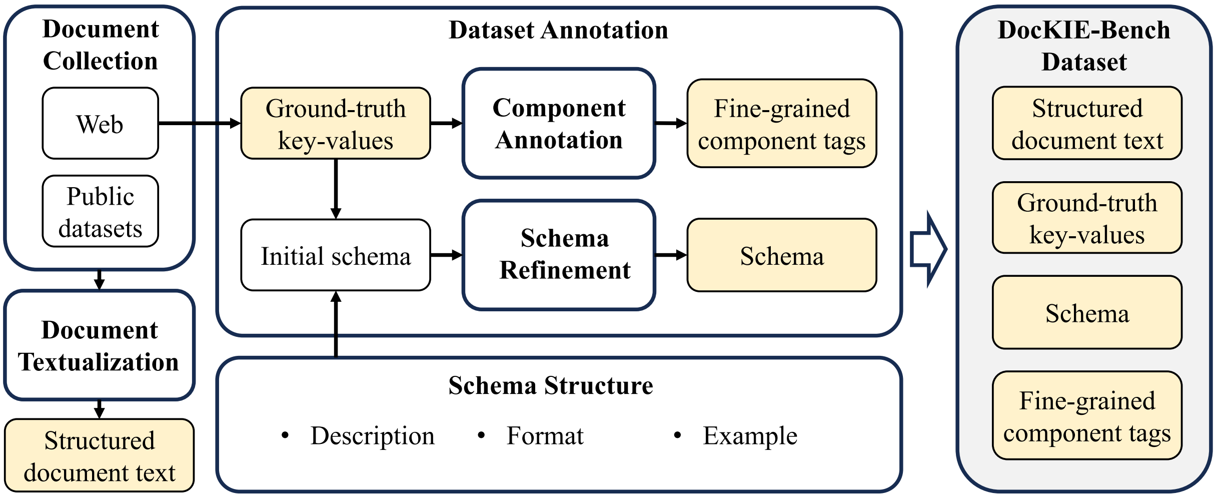

Figure 2: Overview of the dataset construction process for DocKIE-Bench. The process begins with the collection of documents from both manually curated sources and publicly available KIE benchmarks. These documents are then converted into structured text using document parsing tools. Next, key-value pairs are annotated and used to construct corresponding schemas with descriptions, formats, and examples. These schemas are then refined through LLM-based simulation to improve clarity and consistency. Finally, we apply fine-grained document component annotations, such as table, form, checkbox, to enable detailed evaluation of *LLM-for-KIE* paradigm.

POIE (Kuang et al., 2023) expand to product images captured in the wild, and the Kleister series (Stanisławek et al., 2021) focuses on NDA and charity documents. While these benchmarks are suitable for the fine-tuning paradigm, they lack key elements for *LLM-for-KIE*, including broad document-type coverage, detailed entity descriptions, and document component labeling, which are features essential for insightful evaluation of LLM-based KIE systems.

Benchmarks with Broader Document Coverage. Benchmarks like VRDU (Wang et al., 2023b), RealKIE (Townsend et al., 2024) and OmniAI OCR (OmniAI, 2025) improve along the axis of document diversity, covering wider range of formats such as registration form, FCC invoices, and bank check, making them better aligned with real-world industrial use-cases. However, despite their diversity, they still lack detailed schema descriptions and component-level annotations, limiting their applicability for evaluating LLM behavior in training-free settings.

DocKIE-Bench, on the other hand, is constructed from the ground up with the various aspects of *LLM-for-KIE* paradigm in mind. It curates documents from a wide range of sources, to ensure domain diversity, and applies rigorous annotation and revision to meet the paradigm's core requirements: detailed schema for KIE, broad document coverage, and fine-grained component labeling.

2.2 Evaluating LLMs for KIE

Recent efforts to apply LLMs to KIE have led to a wide range of evaluation methodologies. Prior works (He et al., 2023; Perot et al., 2023; Wang et al., 2023a; Luo et al., 2024; Zhang et al., 2025; Zhu et al., 2025) have adopted a variety of strategies to facilitate KIE evaluation of LLMs across prior KIE benchmarks. Some of the strategies include: modification of schema (e.g., rewriting entity keys into natural language); addition of few-shot examples to adapt the LLMs to KIE settings.

While these approaches highlight the potential of LLMs in KIE, the lack of consistency across evaluation strategies hinders fair and accurate comparisons of model performance. This challenge underscores the need for a standardized benchmark that is specifically designed for LLMs in KIE tasks. Without standardized schema descriptions, diverse document coverage, and component-level annotations, prior benchmarks fall short in supporting insightful and interpretable evaluations across different LLM-based works.

To address this gap, we introduce DocKIE-Bench, a benchmark purpose-built for evaluating KIE performance in the *LLM-for-KIE* paradigm.

3 DocKIE-Bench

In this section, we describe the dataset construction process of DocKIE-Bench. Figure 2 illustrates the overall workflow of this construction process.

Dataset	# Types	Dataset	# Types
CORD (2019)	1	DocILE (2023)	2
SROIE (2019)	1	VRDU (2023b)	2
PWC (2020)	1	AutoBench (2025)	4
DeepForm (2020)	1	RealKIE (2024)	5
WildReceipt (2021)	1	SIMARA (2023)	6
ETD500 (2021)	1	FUNSD (2019)	16
POIE (2023)	1	OmniAI OCR (2025)	37
Kleister (2021)	2	**DocKIE-Bench (Ours)**	**38**

Table 1: Number of document types covered by public KIE datasets compared with DocKIE-Bench. DocKIE-Bench comprises 38 types, the largest among all datasets. OmniAI OCR is the next most diverse, but 11 of those contain fewer than three documents.

3.1 Document Collection

We have collected various licensed documents from the web. Details regarding the sources of these documents are described in Appendix A.1. DocKIE-Bench consists of two main components. First, we curated a set of 20 document types commonly found in business operations, such as invoices, purchase orders, contracts, and application forms. For each type, we collected 5 representative samples, resulting in 100 documents. Due to potential privacy concerns in real-world documents, we synthetically generated or anonymized all field values to ensure there were no personally identifiable or sensitive contents.

Second, we expanded DocKIE-Bench with eight adapted public benchmark datasets used in prior KIE studies (Park et al., 2019; Kardas et al., 2020; Choudhury et al., 2021; Kuang et al., 2023; Šimsa et al., 2023; Wang et al., 2023b; Townsend et al., 2024; OmniAI, 2025). These datasets were selected based on their diversity and complementary coverage of document types mostly not represented in our manually collected set and variety of key types. From these benchmarks, we curated 19 distinct documents types and selected a total of 100 samples to augment DocKIE-Bench. Each selected dataset was reformatted to align with the DocKIE-Bench schema to ensure consistency and comparability during evaluation.

In total, DocKIE-Bench comprises 200 documents: 100 manually curated samples and 100 additional samples adapted from existing benchmark datasets. This combination ensures a balanced and diverse evaluation set spanning a wide range of document types. Notably, as shown in Table 1, DocKIE-Bench covers 38 distinct document types, the most among existing KIE benchmarks. Detailed dataset statistics are presented in Appendix A.2.

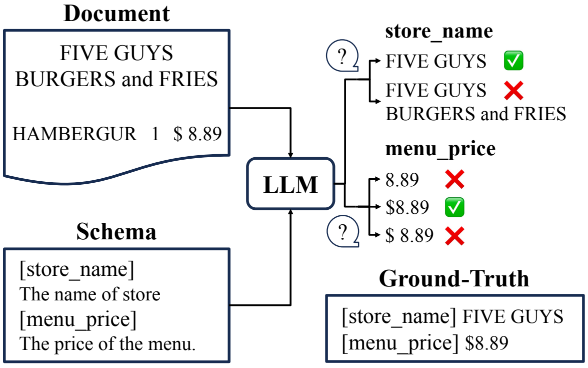

Figure 3: Ambiguous schema descriptions can cause LLMs to generate varied yet semantically valid answers that may be penalized for output differences.

3.2 Document Textualization

Since LLMs generally operate over text inputs, it is essential to convert documents, often stored as PDFs or image files (JPG, PNG), into a textual format. Importantly, since documents convey meaning not only through content but also through visual layout, preserving such structural information during text conversion is critical for accurate understanding and extraction. While traditional OCR tool like Tesseract[1] can extract raw text, they often lose important structural information. Moreover, in case of scanned images embedded in PDFs, PDF parsers alone cannot extract any text.

To address these limitations, we employed a combination of advanced document parsing tools of Upstage Document Parse[2] and LlamaParse[3] to extract structured representations of documents. These tools provide structured text that reflects layout cues (e.g., headers, columns, tables), enabling LLMs to better understand the document layout.

3.3 Schema

3.3.1 Role of Schema for Reliable Evaluation

In traditional KIE setup, key names have been the only information passed to models. However, the flexible output capabilities of modern LLMs often lead to varied yet valid answers, making consistent evaluation difficult. This flexibility, combined with minimal key information, can result in multiple plausible outputs beyond the intended golden answer, as illustrated in Figure 3.

[1] https://github.com/tesseract-ocr/tesseract

[2] https://www.upstage.ai/products/document-parse

[3] https://www.llamaindex.ai/llamaparse

To mitigate this issue, we introduce schema that provide detailed information for each key. The schema clarifies the expected answer format and intent, serving as a reference for determining whether a model's output aligns with user expectations.

3.3.2 Structure of Schema

To mitigate such ambiguity and to enhance consistency, we define each key in the schema using three components: a description, a format, and representative examples.

- **Description:** provides an explanation of what the key represents.
- **Format:** specifies the expected structure or representation of the value. For instance, if the format indicates that currency symbols must precede the amount (e.g., $100), then even if the document contains 100 USD, the model must normalize the output to the specified form.
- **Example:** offers concrete instances of valid values, which help guide the model's generation and support more accurate inference.

The description and format facilitate reliable evaluation of LLM outputs, while the example helps promote consistency in model inference.

In addition to these semantic features, we also incorporate structural aspects into the schema design. To enable flexible and expressive schema definitions, we adopt the JSON Schema standard[4]. This structured format supports a wide range of data types, such as integers, strings, and booleans, and allows for complex key structures, including arrays and nested objects. Such flexibility is particularly valuable for representing grouped keys in tabular documents, where multiple rows with column-wise data must be extracted in groups.

3.3.3 Schema Design Process

We begin by identifying key-value pairs for each document type using qualified human annotators. For public datasets, we leverage existing keys and ground-truth values while for internally collected documents, both keys and ground-truth values are annotated directly. Based on the annotated key-value pairs, we then design the corresponding schema based on the structure defined in Section 3.3.2.

To enhance the consistency and reduce potential ambiguity in schema specification, we performed schema LLM-based refinement process. Specifically, we prompt GPT-4.1 (gpt-4.1-2025-04-14) with the initial schema and generate five independent inferences. If semantically correct but structurally inconsistent outputs (e.g. "$100" vs. "100") appear across runs, we compare them with the ground-truth value and revise the corresponding description or format to enforce a unique, consistent target output. This refinement step helps us produce a more unambiguous schema, enabling more consistent evaluation in *LLM-for-KIE* settings.

3.4 Document Component Annotation

DocKIE-Bench includes document component annotations at the key-value pair level. These annotations help identify which types of visual or structural components are associated with each extracted value, thereby enabling more detailed error analysis. Each key-value pair is tagged with zero, one, or more of the following six component types:

- **table**: appears within a tabular structure.
- **form**: follows a header–content pair pattern, typical in forms.
- **checkbox**: presented using a checkbox element.
- **handwritten**: handwritten rather than machine-printed.
- **plain**: appears in plain text paragraphs.
- **chart**: displayed in chart images.

This document component annotation supports fine-grained evaluation and helps uncover performance trends across different document types and layouts. The categorization of these components was inspired by prior works in document understanding (Mathew et al., 2021, 2022), which emphasized the importance of visual elements.

4 Experiments

This section presents an extensive experimental study that evaluates the performance of LLMs on our DocKIE-Bench. The detailed experimental setup for the use of Structured Outputs, system and user prompts is presented in Appendix B.

4.1 Evaluation Metrics

KIE Metrics We adopt KIEval (Khang et al., 2025) as the metric to assess KIE, where the extracted key-value pairs evaluated with structural awareness (grouping) in mind. KIEval's Entity F1 measures the extraction performance of individual key-value

[4] https://json-schema.org

Model Name	Context Max	KIEval$_{\text{Aligned}}$	Entity F1	Group F1	API Error	Parsing Error	Format Error
gpt-4o-mini-2024-07-18	128,000	$64.22_{\pm 0.11}$	$67.62_{\pm 0.15}$	$30.60_{\pm 0.86}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gpt-4o-2024-11-20	128,000	$65.41_{\pm 0.09}$	$67.77_{\pm 0.07}$	$38.53_{\pm 0.31}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gpt-4.1-nano-2025-04-14	1,047,576	$52.70_{\pm 0.01}$	$56.95_{\pm 0.04}$	$23.64_{\pm 0.88}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gpt-4.1-mini-2025-04-14	1,047,576	$67.30_{\pm 0.21}$	$69.86_{\pm 0.17}$	$39.96_{\pm 0.57}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gpt-4.1-2025-04-14	1,047,576	$71.62_{\pm 0.26}$	$74.03_{\pm 0.15}$	$46.03_{\pm 0.22}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
o3-mini-2025-01-31	200,000	$64.68_{\pm 0.27}$	$67.41_{\pm 0.31}$	$42.26_{\pm 0.26}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
o4-mini-2025-04-16	200,000	$66.48_{\pm 0.15}$	$69.66_{\pm 0.15}$	$41.69_{\pm 0.00}$	$0.67_{\pm 0.47}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gemini-2.0-flash-lite	1,048,576	$53.61_{\pm 0.22}$	$56.27_{\pm 0.19}$	$16.12_{\pm 0.29}$	$0.00_{\pm 0.00}$	$6.00_{\pm 0.82}$	$0.33_{\pm 0.47}$
gemini-2.0-flash	1,048,576	$59.35_{\pm 0.32}$	$63.25_{\pm 0.29}$	$27.17_{\pm 0.62}$	$0.00_{\pm 0.00}$	$3.67_{\pm 0.47}$	$5.00_{\pm 0.00}$
gemini-2.5-flash-preview-04-17†	1,048,576	$63.25_{\pm 0.11}$	$65.30_{\pm 0.11}$	$29.46_{\pm 0.15}$	$3.67_{\pm 0.47}$	$4.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gemini-2.5-pro-preview-05-06†	1,048,576	$62.42_{\pm 0.33}$	$64.22_{\pm 0.31}$	$23.11_{\pm 0.62}$	$0.67_{\pm 0.94}$	$1.33_{\pm 0.47}$	$0.00_{\pm 0.00}$
claude-3-5-haiku-20241022	200,000	$63.53_{\pm 0.16}$	$66.83_{\pm 0.17}$	$37.54_{\pm 0.70}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$3.00_{\pm 0.00}$
claude-3-7-sonnet-20250219	200,000	$70.58_{\pm 0.19}$	$72.78_{\pm 0.19}$	$45.62_{\pm 0.63}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Qwen2.5-0.5B-Instruct	32,768	$19.07_{\pm 0.07}$	$22.52_{\pm 0.07}$	$8.12_{\pm 0.10}$	$0.00_{\pm 0.00}$	$9.00_{\pm 0.82}$	$0.00_{\pm 0.00}$
Qwen2.5-1.5B-Instruct	32,768	$35.25_{\pm 0.08}$	$38.81_{\pm 0.09}$	$12.85_{\pm 0.21}$	$0.00_{\pm 0.00}$	$8.33_{\pm 1.25}$	$0.00_{\pm 0.00}$
Qwen2.5-3B-Instruct	32,768	$47.24_{\pm 0.05}$	$51.51_{\pm 0.03}$	$16.80_{\pm 0.10}$	$0.00_{\pm 0.00}$	$7.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Qwen2.5-7B-Instruct	131,072*	$52.50_{\pm 0.14}$	$56.89_{\pm 0.11}$	$19.88_{\pm 0.14}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Qwen2.5-32B-Instruct	131,072*	$65.80_{\pm 0.13}$	$68.82_{\pm 0.09}$	$40.96_{\pm 0.16}$	$0.00_{\pm 0.00}$	$0.33_{\pm 0.47}$	$0.00_{\pm 0.00}$
Qwen2.5-72B-Instruct	131,072*	$67.17_{\pm 0.31}$	$69.94_{\pm 0.32}$	$41.42_{\pm 0.21}$	$0.00_{\pm 0.00}$	$1.33_{\pm 0.94}$	$0.00_{\pm 0.00}$
gemma-3-1b-it	32,768	$15.73_{\pm 0.18}$	$18.17_{\pm 0.20}$	$1.87_{\pm 0.26}$	$0.00_{\pm 0.00}$	$31.33_{\pm 1.25}$	$0.00_{\pm 0.00}$
gemma-3-4b-it	131,072	$49.90_{\pm 0.13}$	$53.99_{\pm 0.17}$	$19.37_{\pm 0.39}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gemma-3-12b-it	131,072	$59.64_{\pm 0.17}$	$62.56_{\pm 0.18}$	$29.38_{\pm 0.68}$	$8.00_{\pm 0.82}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
gemma-3-27b-it	131,072	$63.50_{\pm 0.08}$	$66.28_{\pm 0.06}$	$37.07_{\pm 0.40}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Mistral-Small-3.1-24B-Instruct	131,072	$68.33_{\pm 0.01}$	$70.97_{\pm 0.01}$	$38.09_{\pm 0.01}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Llama-3.3-70B-Instruct	131,072	$65.53_{\pm 0.06}$	$68.13_{\pm 0.07}$	$41.15_{\pm 0.03}$	$0.00_{\pm 0.00}$	$1.33_{\pm 0.47}$	$0.00_{\pm 0.00}$

Table 2: Performance comparison of various proprietary and open-source LLMs, based on key metrics such as KIEval$_{\text{Aligned}}$, Entity F1, and Group F1, demonstrating the dominance of GPT-4.1 and Claude-3-7-Sonnet, while highlighting the scaling trends within Qwen and Gemma series, as well as the notable performance of Mistral-Small-3.1-24B-Instruct relative to its size. Additionally, the table underscores the challenges faced by reasoning-focused models like Gemini and o-series in maintaining accuracy despite larger context. Some Qwen models leverage RoPE scaling to extend context length from 32K to 128K, as indicated by the * mark in the table. Gemini 2.5 models show API failures due to copyright issues, marked with † in the table to denote these occurrences.

pairs, while Group F1 evaluates the extraction performance at group-level (i.e. groups of related entities). Additionally, KIEval$_{\text{Aligned}}$, a modification of Entity F1 for industrial applications, focuses on the number of corrections required to fix incorrect predictions. More details can be found in Appendix B.3.

Reliability Metrics In addition to extraction performance, we assess model reliability by measuring the frequency of failures in generating valid structured outputs. There are three types of errors: API, Parsing and Format errors. API errors refer to cases where the API call fails due to server-side issues (e.g., timeouts). Parsing errors occur when the model's output cannot be parsed as valid JSON. Format errors, on the other hand, occur when the output is syntactically valid JSON but deviates from the expected schema structure.

All experiments are conducted across three independent runs, and we report the averaged results, along with the standard deviation.

4.2 Model Comparison

Table 2 presents the performance comparison of several proprietary and open-source LLMs under the text-only setting, where visual documents are converted to text using Upstage Document Parse, which yield better KIE results than LlamaParse (see Appendix C).

Among the models evaluated, GPT-4.1 achieves the highest performance across all key metrics, demonstrating superior accuracy and grouping capabilities. Claude 3.7 Sonnet follows closely, delivering consistently strong results across evaluation dimensions. Among open-source models, Mistral (MistralAI, 2025) delivers notably strong performance. While not as large as some of the parameter-heavy models (e.g., 70B+), its 24B scale positions it in the mid-sized range relative to other open models considered in this study. Its competitive performance suggests that larger parameter counts do not always translate directly to better KIE results, especially across different model providers. In contrast, models like Qwen2.5 (Yang et al., 2024) and Gemma 3 (Kamath et al., 2025) exhibit a clear positive scaling trend, with larger

Model Name	Schema	KIEval$_{\text{Aligned}}$	Parsing Error
gpt-4o-mini (2024-07-18)	D	$61.04_{\pm 0.27}$	$1.00_{\pm 0.00}$
	D + F	$62.17_{\pm 0.19}$	$1.33_{\pm 0.47}$
	D + F + E	$64.22_{\pm 0.11}$	$1.00_{\pm 0.00}$
gpt-4o (2024-11-20)	D	$62.70_{\pm 0.30}$	$0.00_{\pm 0.00}$
	D + F	$63.57_{\pm 0.19}$	$0.33_{\pm 0.47}$
	D + F + E	$65.41_{\pm 0.09}$	$0.00_{\pm 0.00}$
gpt-4.1-nano (2025-04-14)	D	$50.03_{\pm 0.33}$	$0.33_{\pm 0.47}$
	D + F	$51.11_{\pm 0.37}$	$0.33_{\pm 0.47}$
	D + F + E	$52.70_{\pm 0.01}$	$0.00_{\pm 0.00}$
gpt-4.1-mini (2025-04-14)	D	$63.68_{\pm 0.19}$	$0.33_{\pm 0.47}$
	D + F	$64.54_{\pm 0.16}$	$0.33_{\pm 0.47}$
	D + F + E	$67.30_{\pm 0.21}$	$0.00_{\pm 0.00}$
gpt-4.1 (2025-04-14)	D	$65.28_{\pm 0.21}$	$0.00_{\pm 0.00}$
	D + F	$66.38_{\pm 0.23}$	$0.00_{\pm 0.00}$
	D + F + E	$71.62_{\pm 0.26}$	$0.00_{\pm 0.00}$

Table 3: Comparison across different schema designs containing description (D), format (F), and example (E).

variants consistently outperforming their smaller counterparts.

Interestingly, reasoning-focused models such as Gemini 2.5, o3-mini, and o4-mini underperform despite extended context windows and large output capacities. For instance, the o-series models lag behind GPT-4.1 and are more comparable in performance to GPT-4o. While Gemini 2.5 improves over its predecessor (2.0 Flash), it still falls short of achieving top-tier scores. These results suggest that advanced reasoning capability alone does not strongly correlate with effective KIE performance.

A notable pattern is also observed among smaller models such as Qwen2.5-0.5B-Instruct and Gemma-3-1b-it, which exhibit significantly higher parsing error rates. This trend highlights a trade-off between model size and parsing robustness, suggesting that reliable KIE performance with minimal parsing errors generally requires models with sufficient capacity (e.g. 4B+).

4.3 Effect of Schema on Performance

Table 3 examines the impact of different schema configurations on LLM performance. As we incrementally enrich the schema with a description (D), formatting rules (F), and examples (E), we observe consistent gains in extraction accuracy.

Comparing the D and D + F settings reveals that adding format guidelines enhances extraction quality, particularly for structured fields such as dates, numerical values, and standardized formats. Format instructions help reduce ambiguity and guide the model toward consistent outputs. For example, fields like dates (e.g., 07/08/2024 vs. 07 / 08 / 24) and addresses benefit from clear formatting expectations. These findings highlight the value of structural guidance in improving model reliability for such keys.

Model Name	Modality	KIEval$_{\text{Aligned}}$	Parsing Error
gpt-4.1-nano (2025-04-14)	T	$52.70_{\pm 0.01}$	$0.00_{\pm 0.00}$
	I	$47.81_{\pm 0.47}$	$0.00_{\pm 0.00}$
	T + I	$55.83_{\pm 0.48}$	$0.67_{\pm 0.47}$
gpt-4.1-mini (2025-04-14)	T	$67.30_{\pm 0.21}$	$0.00_{\pm 0.00}$
	I	$76.04_{\pm 0.17}$	$0.67_{\pm 0.47}$
	T + I	$74.44_{\pm 0.20}$	$0.00_{\pm 0.00}$
gpt-4.1 (2025-04-14)	T	$71.62_{\pm 0.26}$	$0.00_{\pm 0.00}$
	I	$72.46_{\pm 0.15}$	$0.00_{\pm 0.00}$
	T + I	$77.42_{\pm 0.06}$	$0.00_{\pm 0.00}$

Table 4: Performance comparison of different input modalities across GPT 4.1 models. In Modality, T, I, and T + I indicated text-only, image-only and use text and image together, respectively.

Extending the schema further to include examples (D + F + E) provides an additional performance boost. Real-world examples convey nuanced cues, such as address delimiters, apartment labels, currency symbol placement, spacing, and capitalization, that are often difficult to fully articulate through descriptions alone. The inclusion of examples allow the model to infer these subtle patterns, complementing textual and structural instructions with concrete, context-rich signals.

5 Analysis

Building on the experimental results, we proceed with an analysis focused on the assessment of inherent visual understanding capabilities across models and the relationship between different modalities and document components in our benchmark.

5.1 Impact of Modality

While textualizing documents is the most convenient approach for applying LLMs to information extraction, recent advances in multi-modal LLMs allow models to directly process visual documents. This section investigates how well models can interpret and integrate textual and visual inputs.

Table 4 compares three input modalities: HTML text only (T), image only (I), and a combined setting with both text and image inputs (T + I). Overall, image-only inputs tend to outperform text-only inputs, suggesting that key information in our benchmark is primarily grounded in visual layout and structure. An exception is observed with GPT-4.1 nano, where the text-only input slightly outperforms the image-only setting, indicating that smaller models may struggle with visual comprehension in the absence of textual cues, possibly due

Model Name	Modality	Table	Form	Plain	Handwritten	Chart	Checkbox
gpt-4.1-nano (2025-04-14)	T	$72.34_{\pm 0.83}$	$72.10_{\pm 0.22}$	$75.01_{\pm 1.81}$	$37.73_{\pm 0.32}$	$11.11_{\pm 0.00}$	$7.46_{\pm 0.44}$
	I	$63.32_{\pm 1.97}$	$69.23_{\pm 0.33}$	$77.78_{\pm 1.88}$	$52.98_{\pm 1.68}$	$24.52_{\pm 3.30}$	$57.48_{\pm 6.08}$
	T + I	$69.59_{\pm 0.54}$	$74.86_{\pm 0.47}$	$74.29_{\pm 0.48}$	$58.54_{\pm 0.00}$	$16.05_{\pm 0.00}$	$57.37_{\pm 1.59}$
gpt-4.1-mini (2025-04-14)	T	$79.79_{\pm 0.41}$	$77.55_{\pm 0.30}$	$57.81_{\pm 1.35}$	$55.36_{\pm 0.00}$	$53.54_{\pm 2.46}$	$26.53_{\pm 0.00}$
	I	$87.64_{\pm 1.17}$	$83.47_{\pm 0.14}$	$74.01_{\pm 1.66}$	$80.86_{\pm 2.31}$	$57.08_{\pm 12.89}$	$58.08_{\pm 0.56}$
	T + I	$83.78_{\pm 1.67}$	$83.39_{\pm 0.42}$	$70.00_{\pm 0.22}$	$72.62_{\pm 1.68}$	$42.47_{\pm 11.75}$	$53.97_{\pm 2.59}$
gpt-4.1 (2025-04-14)	T	$89.90_{\pm 0.30}$	$80.80_{\pm 0.15}$	$66.05_{\pm 0.13}$	$56.79_{\pm 1.75}$	$44.25_{\pm 5.19}$	$31.93_{\pm 0.25}$
	I	$76.35_{\pm 0.77}$	$77.96_{\pm 0.11}$	$69.03_{\pm 0.84}$	$60.12_{\pm 0.84}$	$82.91_{\pm 2.82}$	$56.40_{\pm 0.21}$
	T + I	$91.46_{\pm 0.33}$	$83.99_{\pm 0.01}$	$66.42_{\pm 2.44}$	$75.00_{\pm 3.86}$	$63.24_{\pm 4.59}$	$50.00_{\pm 1.10}$

Table 5: Component-level analysis of model performance. Each column represents a distinct component, and the rows provide KIEval$_{\text{Aligned}}$ for different models. Notable trends include the substantial performance gains in visually dominant components (Checkbox, Chart, Handwritten) when incorporating image data and the consistent performance across structured components (Table, Form, Plain) regardless of modality.

to limited model capacity.

For the combined modality (T + I), results show a general trend of improvement, demonstrating synergy between text and image inputs. However, GPT-4.1 mini presents a slight drop in performance compared to its image-only setting. This suggests that the inclusion of textual input may, in some cases, introduce noise or conflicting signals that interfere with accurate visual referencing.

In contrast, GPT-4.1 achieves its best performance in the T + I setting, showing a substantial margin over both single-modality inputs, indicating a more advanced capability to effectively integrate multi-modal inputs while filtering out noise. Such findings suggest that higher-performing models may employ more refined mechanisms to align visual and textual cues and prioritize salient information for robust information extraction.

5.2 Document Component

Our benchmark includes component-type annotations for each key-value pair, enabling a more granular analysis of model behavior. The component-level results in Table 5 reveal both the relative difficulty of different components and how input modality influences performance across them.

The superior performance of image-based inputs over text-only inputs is primarily driven by visually dominant components such as Checkbox, Chart, and Handwritten, where visual information plays a crucial role in accurate extraction. In contrast, components with more structured or text-heavy layouts, such as Table, Form, and Plain, exhibit minimal performance differences between text-only and image-only inputs. This suggests that the modality gap is concentrated within a specific subset of components, rather than being uniformly present across all component types.

The combined modality (T + I) proves especially effective for components that depend on both textual labels and spatial structure, including Form, Table, and Handwritten. In these cases, the largest model (GPT-4.1) shows substantial gains from multi-modal integration, whereas smaller models demonstrate limited or inconsistent benefits. This pattern supports the notion that successful multimodal fusion is closely tied to model capacity, with smaller models often struggling to meaningfully leverage both input types effectively.

Overall, these findings highlight the value of our benchmark in analyzing how text, layout, and visual content contribute to model performance across diverse components. This structure serves a basis for more targeted assessments of future models, enabling deeper insights into text and visual processing capabilities of LLMs.

6 Conclusion

We present **DocKIE-Bench**, a benchmark tailored to evaluate LLMs on document KIE. It fills key gaps in existing benchmarks by providing structured schemas with descriptions, formatting rules, and examples; diverse coverage of 38 real-world document types; and fine-grained component annotations linking each value to its visual context.

Our experiments show that proprietary models outperform open-source models, but even the best models fall short of high accuracy, highlighting room for improvement. Ablation studies demonstrate that well-specified schemas significantly improve performance, and multi-modal inputs offer further gains when paired with sufficiently capable models. These results highlight the value of DocKIE-Bench as a comprehensive benchmark for advancing *LLM-for-KIE* paradigm through more reliable evaluation and deeper analysis.

Limitations

While **DocKIE-Bench** is designed to be a comprehensive and practical benchmark for LLM-based KIE, certain limitations remain, primarily due to the inherent complexity of real-world document understanding.

First, the definition of "key information" in a document is inherently use-case dependent. Different practitioners may define varying schemas for the same document, for instance, an invoice could be labeled with a single *invoice_total* field or broken down into *subtotal*, *tax*, and *grand_total*. Exhaustively covering all valid schema interpretations is impractical. To maintain consistency and enable controlled evaluation, DocKIE-Bench adopts a single, carefully curated schema per document type. While this narrows the evaluation scope, it ensures comparability across models in benchmarking.

Second, to provide LLMs with structured, machine-readable input, visual documents are converted to HTML or Markdown via automated parsing tools. As with any learned system, these parsers may occasionally introduce noise that can affect absolute performance. However, since all models are evaluated on the same parsed outputs, such artifacts impact all systems equally, preserving the validity of relative comparisons and performance trends.

Overall, these design choices reflect practical trade-offs aimed at building a usable, reproducible, and extensible benchmark for advancing the study of LLMs in document-level information extraction.

References

Muntabir Hasan Choudhury, Himarsha R Jayanetti, Jian Wu, William A Ingram, and Edward A Fox. 2021. Automatic metadata extraction incorporating visual features from scanned electronic theses and dissertations. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 230–233. IEEE.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.

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.

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

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

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. Axcell: Automatic extraction of results from machine learning papers. *arXiv preprint arXiv:2004.14356*.

Minsoo Khang, Sang Chul Jung, Sungrae Park, and Teakgyu Hong. 2025. Kieval: Evaluation metric for document key information extraction. *arXiv preprint arXiv:2503.05488*.

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.

Jianfeng Kuang, Wei Hua, Dingkang Liang, Mingkun Yang, Deqiang Jiang, Bo Ren, and Xiang Bai. 2023. Visual information extraction in the wild: practical dataset and end-to-end solution. In *International Conference on Document Analysis and Recognition*, pages 36–53. Springer.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. *arXiv preprint arXiv:2105.11210*.

Wenhui Liao, Jiapeng Wang, Hongliang Li, Chengyu Wang, Jun Huang, and Lianwen Jin. 2024. Doclayllm: An efficient and effective multi-modal extension of large language models for text-rich document understanding. *arXiv preprint arXiv:2408.15045*.

Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. 2024. Layoutllm: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15630–15640.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

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

MistralAI. 2025. Mistral small 3.1. `https://mistral.ai/news/mistral-small-3-1`.

Nanonets. 2025. Document processing automation benchmark.

OmniAI. 2025. Omniai ocr benchmark. `https://getomni.ai/ocr-benchmark`.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, and 1 others. 2023. Lmdx: Language model-based document information extraction and localization. *arXiv preprint arXiv:2309.10952*.

Štěpán Šimsa, Milan Šulc, Michal Uřičář, Yash Patel, Ahmed Hamdi, Matěj Kocián, Matyáš Skalickỳ, Jiří Matas, Antoine Doucet, Mickaël Coustaty, and 1 others. 2023. Docile benchmark for document information localization and extraction. In *International Conference on Document Analysis and Recognition*, pages 147–166. Springer.

Jonathan Stray Stacey Svetlichnaya. 2020. Project deepform: Extract information from documents.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, pages 564–579. Springer.

Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenhao Lin, and Wayne Zhang. 2021. Spatial dual-modality graph reasoning for key information extraction. *arXiv preprint arXiv:2103.14470*.

Solène Tarride, Mélodie Boillet, Jean-François Moufflet, and Christopher Kermorvant. 2023. Simara: a database for key-value information extraction from full-page handwritten documents. In *International Conference on Document Analysis and Recognition*, pages 421–437. Springer.

Benjamin Townsend, Madison May, and Christopher Wells. 2024. Realkie: Five novel datasets for enterprise key information extraction. *arXiv preprint arXiv:2403.20101*.

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*.

Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *arXiv preprint arXiv:2202.13669*.

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.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, and 1 others. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv e-prints*, pages arXiv–2412.

Jinyu Zhang, Zhiyuan You, Jize Wang, and Xinyi Le. 2025. Sail: Sample-centric in-context learning for document information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25868–25876.

Zhaoqing Zhu, Chuwei Luo, Zirui Shao, Feiyu Gao, Hangdi Xing, Qi Zheng, and Ji Zhang. 2025. A simple yet effective layout token in large language models for document understanding. *arXiv preprint arXiv:2503.18434*.

A Dataset Details

A.1 Dataset Composition

Table 8 presents detailed information about the datasets included in DocKIE-Bench, including their dataset sources, names, document types, and brief descriptions. This table provides a comprehensive view of both manually curated datasets and adapted public benchmarks used in DocKIE-Bench.

A.2 Dataset Statistics

	Mean ± Std	Range (Min - Max)
# Pages	12.05 ± 49.85	1 - 474
# Tokens	10,179.40 ± 34,097.75	142 - 234,542
# Keys	14.66 ± 7.45	3 - 34
# Values	48.80 ± 90.64	4 - 651
# Groups	5.51 ± 11.51	0 - 58

Table 6: Document-level statistics of DocKIE-Bench. Each row represents a different statistics computed over 200 documents. Token counts were computed using the GPT-4.1 tokenizer on documents parsed into HTML format using Upstage Document Parse.

DocKIE-Bench consists of 200 documents from manually curated and public KIE benchmarks. Table 6 summarizes the key statistics of DocKIE-Bench. The number of pages per document varies, with some documents consisting of a single page and others containing up to 474 pages, reflecting realistic variability found in practical settings. In cases where the number of pages is large, the resulting token-length could be as long as 230K tokens, exceeding the context window of many LLMs. This underscores one of the key challenges of LLMs in realistic KIE applications, reflected in our benchmark. Similarly, the number of target keys, ground-truth values, and groups ranges from 3 to 34, 4 to 651, and 0 to 58, indicating the diversity in information density across different document types. This diversity allows DocKIE-Bench to support comprehensive evaluation across a wide variety of document structures and extraction challenges.

B Experimental Details

B.1 Structured Outputs

To practically utilize KIE from LLM inference results, it is essential to be able to automatically parse values with corresponding keys. At the moment, the most effective method for this is to utilize Structured Outputs. Recent LLM inference APIs, including proprietary services like OpenAI[5], as well as open-source libraries like vLLM[6], provide support for Structured Outputs, a feature that guides LLM decoding to adhere to a predefined JSON schema. Since our schema format follows JSON schema, we can easily integrate Structured Outputs with our benchmark, allowing model responses to be generated in a format that conforms exactly to the schema.

B.2 System and User Prompts

Both system and user prompts play a crucial role in guiding the LLM to perform the task as intended. To ensure a fair comparison, we designed prompts to be as concise and general as possible while maintaining clarity in task-specific output.

Figure 4 and Figure 5 illustrate the system and user prompts used for proprietary and open-source models, respectively. For proprietary models (e.g., GPT, Gemini), the JSON schema used for Structured Outputs is typically handled internally by the API provider through opaque mechanisms. In contrast, for open-source models (e.g., Mistral, Gemma), which are commonly deployed via inference frameworks such as vLLM, the schema must be explicitly provided by the user. So, for open-source models, we explicitly include the stringified JSON schema in the system prompt to ensure consistency in output structure.

The textualized document is included as part of the user prompt to provide the model with the necessary input for extraction. In cases where the combined input exceeds the model context window, we truncate the textualized document from the end to fit within the maximum token limit.

B.3 KIEval

Conventional KIE evaluation approaches typically focus on entity-level F1 scores. However, as discussed in (Khang et al., 2025), these methods often overlook the structural grouping of related entities, an essential aspect for downstream applications such as saving the extracted result to relational databases. To address this, we adopt KIEval (Khang et al., 2025), an evaluation metric designed to assess key-value extraction with structural awareness (grouping) in mind. This metric first aligns predicted and ground-truth groups when group enti-

[5] https://platform.openai.com/docs/guides/structured-outputs

[6] https://docs.vllm.ai/en/latest/features/structured_outputs.html

Model Name	**Parser**	**$\text{KIEval}_{\text{Aligned}}$**	**Parsing Error**
gpt-4o-mini (2024-07-18)	DP Llamaparse	$64.22_{\pm 0.11}$ $50.48_{\pm 0.07}$	$1.00_{\pm 0.00}$ $0.00_{\pm 0.00}$
gpt-4o (2024-11-20)	DP Llamaparse	$65.41_{\pm 0.09}$ $50.62_{\pm 0.10}$	$0.00_{\pm 0.00}$ $0.67_{\pm 0.47}$
gpt-4.1-nano (2025-04-14)	DP Llamaparse	$52.70_{\pm 0.01}$ $45.04_{\pm 0.11}$	$0.00_{\pm 0.00}$ $0.00_{\pm 0.00}$
gpt-4.1-mini (2025-04-14)	DP Llamaparse	$67.30_{\pm 0.21}$ $51.63_{\pm 0.38}$	$0.00_{\pm 0.00}$ $1.33_{\pm 0.47}$
gpt-4.1 (2025-04-14)	DP Llamaparse	$71.62_{\pm 0.26}$ $53.07_{\pm 0.16}$	$0.00_{\pm 0.00}$ $0.00_{\pm 0.00}$

Table 7: Performance comparison using Upstage Document Parse (DP) and LlamaParse across multiple GPT model configurations. Note that LlamaParse failed to produce textual outputs for seven documents, resulting in incomplete input for those cases.

ties are present, and then compute both entity-level and group-level scores based on the matched pairs.

C Evaluation with Different Document Parsers

The method used to convert documents into text has a significant impact on the downstream performance of KIE systems. To investigate this, we compare two representative parsing approaches: HTML outputs from Upstage Document Parse (DP) and Markdown outputs from LlamaParse using its default “Balanced” mode. Table 7 summarizes the performance of LLMs under each setting.

Across all GPT models, DP consistently outperforms LlamaParse by a notable margin, demonstrating the importance of accurate and structured parsing for reliable extraction. It is worth noting that LlamaParse failed to generate usable Markdown for seven documents, resulting in incomplete textual inputs for those cases, which negatively impacted downstream extraction performance.

Prompt format for proprietary models

System:
You are a helpful assistant that extracts structured information from documents.

User:
Content to analyze: {content}

1. If you cannot find the information or the value is not mentioned, return nothing.
2. If you can find more than one value for a key, return all the values in an array.
3. Return the value only if the given key's value exists in the provided content. If it does not exist, return empty string.

Figure 4: Prompt format for proprietary models typically accessed via APIs (e.g., GPT, Gemini), with the textualized document inserted at {content}.

Prompt format for open-source models

System:
You are a helpful assistant that extracts structured information from documents.

Your responses should follow the schema:
[Start of schema]
{JSON schema}
[End of schema]
Please ensure your answers adhere to this format and do not contain any unnecessary text.

User:
Content to analyze: {content}

1. If you cannot find the information or the value is not mentioned, return nothing.
2. If you can find more than one value for a key, return all the values in an array.
3. Return the value only if the given key's value exists in the provided content. If it does not exist, return empty string.

Figure 5: Prompt format for open-source models (e.g., Mistral, Gemma) typically executed with inference libraries such as vLLM. JSON schema and textualized document are inserted into {JSON schema} and {content}, respectively.

Dataset source	Dataset name	Document type	Description
Curated	-	air waybill	An air waybill is a receipt issued by an international airline for goods and an evidence of the contract of carriage.
Curated	-	annuity account withdrawal application	An annuity account withdrawal application is a form used by individuals to request the withdrawal of funds from their annuity accounts.
Curated	-	application for individual annuity	An application for individual annuity is a document used to apply for a personal annuity plan.
Curated	-	bank statement	A bank statement is a summary issued by a bank that details a customer's account transactions and balances over a specific period.
Curated	-	bill of lading	A bill of lading is a legal document issued by a carrier to acknowledge receipt of cargo for shipment and acts as a shipment contract.
Curated	-	business card	A business card is a small printed card that contains a person's name, company affiliation, contact details, and professional title.
Curated	-	commercial invoice	A commercial invoice is a customs document used in international trade that provides details about the sale transaction and goods shipped.
Curated	-	driver license	A driver license is an official document permitting a person to operate a motor vehicle and serves as a form of personal identification.
Curated	-	f1040	Form 1040 is an IRS tax form used by individuals to file annual income tax returns in the United States.
Curated	-	f9465	Form 9465 is an IRS document used to request a monthly installment plan for paying off tax debts.
Curated	-	fss4	Form SS-4 is used by entities in the United States to apply for an Employer Identification Number (EIN) from the IRS.
Curated	-	fw2	Form W-2 is an IRS tax form used by employers to report wages paid to employees and the taxes withheld from them.
Curated	-	fw9	Form W-9 is a tax form used in the United States to request a taxpayer identification number (TIN) and certification.
Curated	-	medical report	A medical report is a document prepared by healthcare professionals detailing a patient's medical history, diagnosis, and treatment.
Curated	-	packing list	A packing list is a shipping document that itemizes the contents of a package or shipment, used for inventory and customs purposes.
Curated	-	passport	A passport is an official government-issued document that certifies a person's identity and nationality for international travel.
Curated	-	receipt	A receipt is a document acknowledging that a person has received money or goods in exchange for a product or service.
Curated	-	resume	A resume is a document created by an individual to present their background, skills, and accomplishments for job applications.
Curated	-	shipping request	A shipping request is a document submitted to initiate the shipment of goods, detailing the sender, recipient, and contents.
Curated	-	travel insurance claim	A travel insurance claim is a form submitted to an insurer requesting compensation for losses incurred during travel, such as medical emergencies or cancellations.
Public	CORD	Receipt	A receipt is a document acknowledging that a person has received money or goods in exchange for a product or service.
Public	RealKIE	SEC S1 Filings	SEC S-1 filings are registration documents submitted to the U.S. Securities and Exchange Commission for companies planning to go public, detailing financial and business information.
Public	RealKIE	US Non-Disclosure Agreements	A US Non-Disclosure Agreement is a legal contract that prevents parties from disclosing confidential information shared during business activities.
Public	RealKIE	UK Charity Reports	UK Charity Reports are documents submitted by charitable organizations in the UK outlining financial statements, activities, and compliance with charity regulations.
Public	RealKIE	FCC Invoices	FCC invoices are billing documents related to regulatory services or fines issued by the U.S. Federal Communications Commission.
Public	RealKIE	Resource Contracts	Resource contracts are legal agreements that define the terms for the extraction, use, or allocation of natural or organizational resources.
Public	OmniAI OCR	Staff Shift Schedule	A staff shift schedule is a document that outlines the working hours and assigned shifts of employees over a given time period.
Public	OmniAI OCR	DEMOCRATIC DESIGNATING PETITION	A Democratic Designating Petition is a political document used to gather signatures for placing a candidate on the ballot in a Democratic primary election.
Public	OmniAI OCR	Glossary	A glossary is a list of terms and their definitions, typically used to explain technical or domain-specific vocabulary.
Public	OmniAI OCR	Real Estate Transaction	A real estate transaction document contains statistical data summarizing property sales, prices, and market trends within a specific region or period.
Public	OmniAI OCR	CALIFORNIA COMMERCIAL LEASE AGREEMENT	A California Commercial Lease Agreement is a legally binding contract outlining terms for renting commercial property in the state of California.
Public	OmniAI OCR	Bank Check	A bank check is a written, dated, and signed instrument that directs a bank to pay a specific sum of money to the bearer or a designated person.
Public	OmniAI OCR	Money Flow Report	A money flow report details the movement of funds within an organization or account over a specific period.
Public	OmniAI OCR	Medical Equipment Inspection Checklist	A medical equipment inspection checklist is a document used to verify the functionality and safety compliance of medical devices.
Public	VRDU	Ad-buy Forms	Ad-buy forms are documents used to request or confirm the purchase of advertising space across various media platforms.
Public	VRDU	Registration Forms	Registration Forms are government documents filed by foreign agents with the US government, containing essential details such as agent names, bureau addresses, activity purposes.
Public	POIE	Product Info	The Product Info documents consist of camera-captured images of real-world product packaging.
Public	PWC	Machine Learning Papers	Machine learning papers are academic or technical documents that present research, methodologies, and findings in the field of machine learning.
Public	FUNSD	Noisy Scanned Documents	Noisy Scanned Documents are low-resolution grayscale images from the RVL-CDIP collection, containing realistic noise introduced through repeated scanning and printing.
Public	SROIE	Receipt	A receipt is a document acknowledging that a person has received money or goods in exchange for a product or service.

Table 8: Detailed information on the datasets included in DocKIE-Bench. Each entry presents the dataset source, name, document type, and a brief description, offering a comprehensive overview of the benchmark, which comprises both manually curated datasets and adapted public benchmarks.