21 November 2025

Large Language Models for Structured Document Understanding: A Comprehensive Survey of Tasks, Models, and Challenges

Zahra Anvari[1], Vassilis Athitsos

1. Independent AI Researcher

Abstract

Structured document understanding (SDU) plays a critical role in numerous industrial and enterprise applications, including finance, healthcare, legal, and insurance domains. While layout-aware transformer models have long defined the state of the art, the emergence of large language models (LLMs), particularly instruction-tuned and multimodal variants, has introduced new paradigms for solving SDU tasks such as key information extraction (KIE), named entity recognition (NER), document classification, and document question answering (DocQA). This paper presents a structured survey and interpretive review of both LLM-based and traditional non-LLM models across these core tasks. Rather than performing new empirical benchmarking, we synthesize results reported in prior works to analyze trends in architecture, modality fusion, task adaptability, and prompt sensitivity. Our analysis highlights the evolving trade-offs between generative and discriminative modeling, layout-aware vs. OCR-free strategies, and model efficiency vs. generalization. We conclude by outlining open research challenges related to layout grounding, multilingual robustness, evaluation protocols, and instruction-following stability. This survey aims to serve as a foundational reference for researchers and practitioners navigating the rapidly evolving document AI landscape.

Keywords

computer science, computer vision, computing and processing, document intelligence, large language models, nlp

Large Language Models for Structured Document Understanding: A Comprehensive Survey of Tasks, Models, and Challenges

Zahra Anvari [*], Vassilis Athitsos [†]

November 15, 2025

Abstract

Structured document understanding (SDU) plays a critical role in numerous industrial and enterprise applications, including finance, healthcare, legal, and insurance domains. While layout-aware transformer models have long defined the state of the art, the emergence of large language models (LLMs), particularly instruction-tuned and multimodal variants, has introduced new paradigms for solving SDU tasks such as key information extraction (KIE), named entity recognition (NER), document classification, and document question answering (DocQA).

This paper presents a structured survey and interpretive review of both LLM-based and traditional non-LLM models across these core tasks. Rather than performing new empirical benchmarking, we synthesize results reported in prior works to analyze trends in architecture, modality fusion, task adaptability, and prompt sensitivity. Our analysis highlights the evolving trade-offs between generative and discriminative modeling, layout-aware vs. OCR-free strategies, and model efficiency vs. generalization. We conclude by outlining open research challenges related to layout grounding, multilingual robustness, evaluation protocols, and instruction-following stability. This survey aims to serve as a foundational reference for researchers and practitioners navigating the rapidly evolving document AI landscape.

1 Introduction

Structured document understanding (SDU) underpins critical workflows in domains such as finance, healthcare, and legal compliance, where extracting accurate information from visually complex and semantically rich documents is essential. Traditional approaches, often reliant on rule-based heuristics and OCR systems, have given way to more powerful solutions driven by deep learning—particularly layout-aware transformers and multimodal vision-language models.

Simultaneously, instruction-tuned large language models (LLMs) have redefined the boundaries of generalization in both NLP and vision-language tasks. This evolution prompts a critical inquiry: Can LLMs outperform traditional layout-specific models in SDU tasks? What are the comparative strengths in terms of accuracy, generalizability, efficiency, and adaptability? And to what extent can generative reasoning replace token-level structured extraction?

In response, this paper delivers a comprehensive survey and comparative benchmark of both LLM-based and non-LLM approaches across four foundational SDU tasks: key-value pair extraction, named entity recognition (NER), document classification, and document question answering (DocQA). Rather than proposing new models, we synthesize and interpret findings from over 20 prominent architectures, including LayoutLM [1, 2], UDOP [3], DocLLM [4], and MoE-LLaVA [5], providing a unified lens into architectural trends, modality integration strategies, and evaluation outcomes.

Our goal is not merely to catalog models, but to establish a conceptual framework that contextualizes current advancements. We classify systems by architectural paradigm (encoder, encoder-decoder, decoder), input modality (text, layout, image), and learning strategy (prompt-driven, instruction-tuned, or multimodal fusion). We also spotlight persistent challenges, including hallucination, layout misalignment, and prompt instability.

[*]Independent AI Researcher, Email: zahra.anvari.uta@gmail.com

[†]Professor at The University of Texas at Arlington, Computer Science and Engineering, Email: athitsos@uta.edu

The scope of this survey spans:

- **key information extraction (KIE)** – retrieving structured field-value pairs from forms and receipts [6, 7]
- **Named Entity Recognition (NER)** – identifying entities in visually structured documents [8, 9]
- **Document Classification** – categorizing documents by layout or content [10, 11]
- **Document Question Answering (DocQA)** – answering free-form queries grounded in document content and layout [12, 13, 14]

Unlike prior reviews that limit themselves to taxonomy or qualitative comparisons, our work presents task-wise quantitative benchmarks across over 10 datasets and 20 models. We evaluate performance using standard metrics such as F1, EM, accuracy, and ANLS, and analyze trends in prompt sensitivity, OCR dependency, and domain generalization.

Key contributions:

- A taxonomy of LLM and non-LLM models organized by architecture, input modality, and learning style.
- Task-level performance comparisons across layout-aware transformers, graph models, and instruction-tuned LLMs.
- A critical analysis of dataset biases, benchmark saturation, and generalization bottlenecks.
- Synthesis of emerging trends in instruction tuning [15], OCR-free modeling [16], and multilingual document processing [8].
- A forward-looking discussion on future research challenges in prompt stability, lightweight adaptation, and benchmark standardization.

While our benchmark results are drawn from published sources and official repositories, we acknowledge that reported metrics may vary due to prompt design, OCR quality, and evaluation methodology. Section 7 addresses these caveats in detail.

Finally, we synthesize architectural trade-offs across model families, highlighting differences in layout grounding, generative flexibility, and deployment feasibility (Section 8).

We intend this survey to serve as both a reference and a roadmap for researchers and practitioners building next-generation document AI systems.

2 Task Formulation

Structured document understanding (SDU) encompasses a range of tasks aimed at extracting, classifying, or interpreting information from documents that combine textual, spatial, and often visual elements. These tasks typically operate on scanned PDFs, form-like images, and OCR-rendered document content. We formalize four key tasks central to this survey, along with their subtasks, data structures, and evaluation metrics.

2.1 Key Information Extraction (KIE)

The KIE task involves extracting field-value pairs such as `Name: John Doe` or `Total: $234.50` from structured or semi-structured documents like invoices, receipts, and tax forms.

Input: Scanned image or OCR-rendered text and layout.

Output: A set of `<key, value>` pairs.

Example:

```
Invoice Date: July 1, 2023
Total Amount: $1200
Customer Name: Alice Wong
```

Evaluation Metrics. Key Information Extraction (KIE; KVP-style) is primarily evaluated using the F1-score, computed either at the span level or based on full entity matches. This metric captures both precision and recall in identifying correct key-value alignments. In addition, Exact Match (EM) is often used to assess strict string-level equivalence between predicted and reference values, offering a more rigid assessment of correctness in structured outputs.

Variants. KIE extraction tasks can be categorized into two primary types. In *form-centric extraction*, key-value fields are spatially aligned, typically appearing in structured formats such as tables, forms, or receipts—where positional layout cues aid model performance. Conversely, *free-form KIE extraction* deals with open-domain documents where keys and values are embedded within natural language, requiring the model to infer fields from surrounding textual or semantic context. This setting poses greater challenges in both alignment and generalization.

2.2 Named Entity Recognition (NER)

NER aims to detect and classify entities such as `person names`, `organizations`, `dates`, and `monetary values` from documents.

Input: OCR-extracted text and optionally layout/vision features.

Output: Entities labeled with a class tag.

Example:

```
"Amazon Inc." => ORG
"July 1, 2023" => DATE
"$500" => MONEY
```

Evaluation Metrics. NER models are primarily evaluated using *token-level* or *span-level* F1-scores, depending on whether predictions are assessed per-token or as complete entity spans. These scores can be *micro-averaged*—favoring frequent classes—or *macro-averaged* to balance across all entity types, including rare categories.

Variants. NER tasks include *flat NER*, where entities do not overlap, and *nested NER*, where entities may be embedded within others (e.g., a `PERSON` inside an `ORG`). Additionally, *domain-specific NER* is common in specialized contexts such as *biomedical*, *legal*, or *financial* documents, where custom entity schemas and terminology present unique modeling challenges.

2.3 Document Classification

This task assigns a label (or multilabel) to the entire document based on its content and layout.

Input: Document-level text, layout, and/or image.

Output: Class label(s) such as `invoice`, `passport`, `resume`

Example:

```
Document Type => Purchase Order
Document Type => Utility Bill
```

Evaluation Metrics. Document classification tasks are typically assessed using standard metrics such as *accuracy*, *precision*, *recall*, and *F1-score*, which offer a balanced view of performance across balanced and imbalanced datasets. In multi-label scenarios—where documents may belong to multiple classes simultaneously—metrics like *macro-averaged F1* or *Area Under the Curve (AUC)* are used to capture performance across all labels and account for varying class distributions.

Variants. Document classification can be formulated in several forms. *Single-label classification* assigns a document to one exclusive category, while *multi-label classification* allows for multiple simultaneous labels, which is common in complex document corpora like legal or medical records. *Hierarchical classification* organizes labels in a taxonomy, requiring models to reason over both coarse and fine-grained levels. *Zero-shot classification* evaluates a model's ability to generalize to unseen classes using natural language descriptions or prompt-based inference, a setting increasingly supported by large language models.

2.4 Document Question Answering (DocQA)

DocQA focuses on answering natural language questions using the document content, layout, and visual context.

Input: Document (OCR+layout or image) and a natural language query.

Output: Answer string or bounding box span.

Example:

```
Q: What is the total amount?
A: $325.00

Q: What is the invoice number?
A: INV-2023-0045
```

Evaluation Metrics. Structured document understanding tasks, particularly document question answering (DocQA), are evaluated using a range of task-specific metrics. *Exact Match (EM)* and *character-level accuracy* are commonly used in extractive settings to assess whether the model output exactly corresponds to the ground truth. For cases where exact overlap is too strict, *Average Normalized Levenshtein Similarity (ANLS)* offers a softer alignment by measuring string edit distance. In generative scenarios, especially when answers are open-ended or composed of multiple tokens, metrics such as *BLEU* and *ROUGE* are employed to quantify overlap between predicted and reference sequences.

Variants. Document QA is typically categorized into three major variants. *Extractive QA* involves selecting a specific span of text or bounding box from the document as the answer, and is common in datasets like DocVQA. *Generative QA* permits open-ended responses, often structured in natural language or JSON, and is supported by models like Donut and UDOP. Lastly, *multimodal QA* encompasses questions that require integrating visual features such as layout, font, or color—demanding holistic understanding of both the document's content and its presentation.

3 Datasets

Robust evaluation of document understanding models requires access to diverse, well-annotated datasets encompassing a wide variety of formats, layouts, and tasks. This section presents a comprehensive overview of benchmark datasets commonly used in the field. These datasets span four core task categories: key-value pair extraction, named entity recognition (NER), document classification, and document question answering (QA).

Dataset Summary

Table 1 highlights key characteristics of widely adopted document understanding datasets.

Terminology. We use *key information extraction (KIE)* as an umbrella term for extracting structured fields from documents. *Key–value pair (KVP) extraction* refers to the special case, common in forms and receipts, where fields appear as (key, value) pairs. Unless stated otherwise, we use KIE.

3.1 Key Information Extraction (KIE) (incl. KVP)

The **FUNSD** [6], **SROIE** [9], and **CORD** [7] datasets serve as foundational benchmarks for KIE (KVP-style) extraction and NER in visually structured documents. These datasets provide OCR-extracted text and bounding box annotations, enabling layout-aware modeling.

FUNSD focuses on scanned forms and provides semantic annotations across four entity types—'question,' 'answer,' 'header,' and 'other'—along with relationship labels, making it ideal for evaluating field linkage. SROIE, by contrast, centers on scanned receipts with flat annotations for four fields: company name, address, date, and total. Despite its simpler structure, its clean layout and task specificity make it well-suited for real-world receipt parsing.

Table 1: Key characteristics of document understanding datasets.

Dataset	Task(s)	Modality	Language	Size
FUNSD	KIE (KVP-style), NER	OCR+Layout	English	199 forms
SROIE	KIE (KVP-style), NER	OCR+Layout	English	1000 receipts
CORD	KIE (KVP-style), Classification	OCR+Layout	English	1000 receipts
XFUND	Multilingual NER	OCR+Layout	10 languages	1000 forms
Kleister-NDA	NER, QA	OCR+Layout	English	254 NDAs
RVL-CDIP	Classification	Document Image	English	400K images
Tobacco3482	Classification	Image	English	3,482 documents
DocBank	Layout Segmentation	PDF+Layout	English	500K pages
DocVQA	Visual QA	OCR+Image	English	12K QA pairs
OCR-VQA	Visual QA	Image+OCR	English	1M QA pairs
InfographicVQA	Multimodal QA	Image+Layout+OCR	English	30K QA pairs
ChartQA	Table/Chart QA	Image+Text	English	42K QA pairs
PubTables-1M	Table Structure Recognition	HTML+Image	English	1M+ tables
TableBank	Table Detection	Image+LaTeX	English/Chinese	417K tables
DocLayNet	Layout Segmentation	Image+Mask	English	80K pages

CORD extends the complexity by including hierarchical annotations for menu items, categories, and prices, along with classification metadata. It supports both KIE extraction and document classification, introducing structural variety and realistic layout diversity. Together, these datasets reflect a progression from flat (SROIE), to semantically linked (FUNSD), to hierarchically structured formats (CORD), enabling evaluation across increasingly complex document forms.

3.2 Named Entity Recognition (NER)

XFUND [8] and **Kleister-NDA** [17] broaden NER evaluation beyond English and generic domains. XFUND, a multilingual extension of FUNSD, includes forms in 10 languages with bounding boxes and entity annotations—supporting layout-aware multilingual NER research.

Kleister-NDA, by contrast, targets legal domain documents—specifically NDAs—with annotations for entities such as parties and clauses. While XFUND emphasizes multilingual layout generalization, Kleister-NDA challenges models with long-form, domain-specific language. These datasets demonstrate the importance of both linguistic and domain diversity in NER benchmarks.

3.3 Document Classification

RVL-CDIP [10], **Tobacco3482** [11], and **DocBank** [18] provide complementary classification datasets. RVL-CDIP is the most widely used, offering 400,000 grayscale images categorized into 16 classes. Tobacco3482 is smaller (3,482 samples) and focuses on litigation documents—ideal for low-resource and few-shot evaluation.

DocBank provides fine-grained PDF annotations for layout segmentation (e.g., titles, captions, lists) and supports weakly supervised classification and layout-aware training. Collectively, these datasets test generalization (RVL-CDIP), low-resource robustness (Tobacco3482), and structural understanding (DocBank).

3.4 Document Question Answering (QA)

Document QA tasks are evaluated using **DocVQA** [12], **OCR-VQA** [14], **InfographicVQA** [13], and **ChartQA** [19].

DocVQA presents 12K question-answer pairs across scanned manuals and brochures, requiring both textual and layout reasoning. OCR-VQA tests OCR-based models using natural scene images. InfographicVQA introduces stylistic and semantic complexity with infographic-style inputs. ChartQA focuses on reasoning over structured graphical data, such as bar and line charts. These datasets collectively support evaluation across layout, OCR, and multimodal reasoning tasks.

3.5 Tabular and Layout Analysis

Table and Layout Segmentation. Datasets in this category enable structural understanding of documents. PubTables-1M [20] provides over a million tables annotated with HTML structure. TableBank [21] offers 417K tables from Word and LaTeX sources with bounding boxes and layout labels, spanning English and Chinese.

DocLayNet [22] expands beyond tables with 80K pages labeled at the pixel level for elements like text, figures, and lists. These datasets support both table-specific parsing and full document layout segmentation, serving both layout-aware and vision-based models.

Key Observations

Task Coverage. Most benchmarks are concentrated around KIE and NER, while QA and layout segmentation tasks are gaining importance with datasets like DocVQA and DocLayNet.

Modality Diversity. Traditional OCR+Layout datasets dominate, but OCR-free visual datasets—such as those used by Donut and SynthDoG [23]—are emerging as critical resources.

Size Limitations. KIE/NER datasets like FUNSD and XFUND are relatively small, whereas QA datasets such as OCR-VQA and InfographicVQA offer broader supervision.

Multilinguality. Multilingual support remains limited. XFUND and TableBank offer rare examples, signaling an important research gap.

Dataset Samples

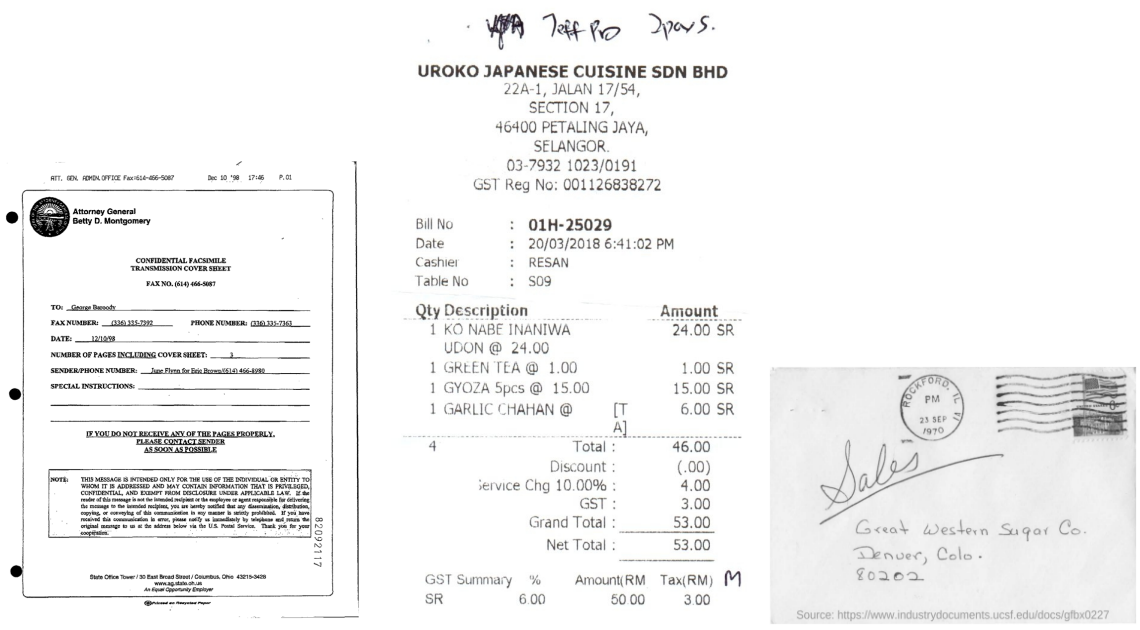


Figure 1: Sample annotations from key datasets. Left: FUNSD (KIE/NER), Center: SROIE (receipt KIE), Right: DocVQA (free-form document QA).

4 Methods Overview

Document understanding models have evolved from early layout-aware encoders to modern large language models (LLMs) with instruction-following and multimodal capabilities. This section offers a comparative overview of non-LLM and LLM-based approaches, emphasizing architectural strategies, fusion mechanisms, and task adaptability.

4.1 Non-LLM Approaches

4.1.1 Comparative Summary: Layout-Aware Encoders

Layout-aware models like **LayoutLMv2**, **LayoutLMv3**, **DocFormer**, and **FormNet** all extend transformer architectures with spatial structure awareness. LayoutLMv2 fuses text, layout, and visual patches using a ResNet backbone. LayoutLMv3 and DocFormer eliminate CNNs, leveraging ViT-style embeddings and gated attention, respectively, for tighter text-image integration. FormNet differs by forgoing vision entirely and modeling tabular structure through Row-Column Enhanced Attention (RCE-Attn). These models target form understanding and structured prediction tasks but lack generative decoding or instruction-tuning.

Model	Vision Input	Layout Fusion	Strength
LayoutLMv2	ResNet features	Cross-attention	Robust with image + layout
LayoutLMv3	ViT patches	Joint transformer	Unified multimodal token stream
DocFormer	Visual tokens	Gated attention	Efficient vision-text fusion
FormNet	None	Row-column attention	Tabular layout bias

Table 2: Comparison of layout-aware encoder-based document understanding models.

To visualize the architectural distinctions, see Figure 2. It shows how each model integrates text, layout, and vision.

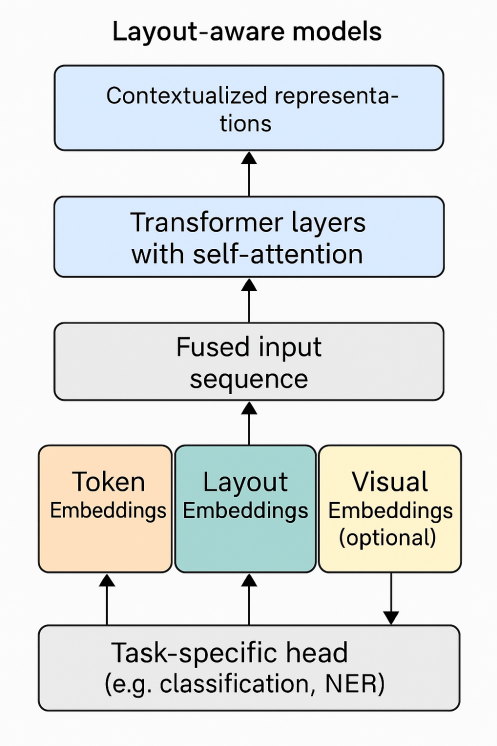


Figure 2: Comparative diagram of layout-aware models showing fusion strategies across LayoutLMv2, LayoutLMv3, DocFormer, FormNet, and LayoutLLM.

In contrast to the encoder-only architectures above, newer models like Donut and DocLLM adopt generative frameworks, which we discuss next.

4.1.2 PICK

PICK (Processing Key Information Extraction) [24] adopts a graph-based modeling approach for document understanding, focusing specifically on KIE extraction. It diverges from traditional token-level transformers by treating each text box as a node in a graph, thereby aligning more naturally with document layout.

Architecture and Reasoning. Each text box is encoded with a BiLSTM for textual features, bounding box coordinates for layout, and additional metadata. A Graph Attention Network (GAT) [25] updates each node by attending to its neighbors based on spatial proximity and semantic content. A classification head assigns roles such as key or value to each node.

Comparison. PICK is lightweight and excels in structured KIE tasks like those in SROIE and CORD. However, it lacks token-level granularity and is not suited for tasks requiring generative reasoning or instruction following. While it is less scalable than transformer-based models, its graph formulation offers strong interpretability.

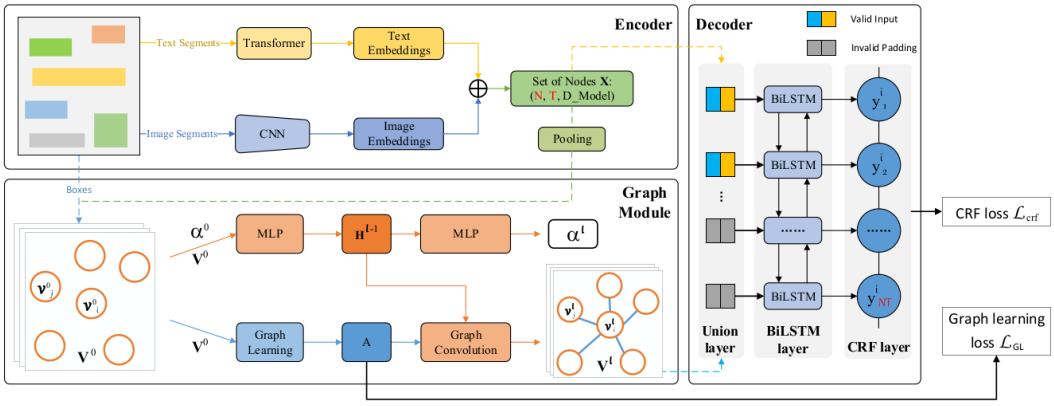


Figure 3: PICK architecture [24]: Graph-based key information extraction model that uses layout-aware attention between text boxes.

PICK excels at modeling document structure explicitly, especially when text boxes align well with semantic fields. Its lightweight architecture performs well on datasets like SROIE and CORD. However, PICK lacks token-level granularity and generative capabilities, limiting its adaptability to free-form document types or tasks requiring detailed output structures. Compared to transformer-based models like LayoutLM, PICK offers interpretability through graph interactions but is less scalable to complex layouts or long documents.

4.1.3 FormNet

FormNet [26] is tailored for structured form understanding. It introduces inductive biases via Row-Column Enhanced Attention (RCE-Attn) to explicitly model the spatial layout of documents, particularly tabular structures.

Architecture and Inductive Bias. Unlike token-level models, FormNet operates on text boxes as semantic units. Each box is encoded using textual features, bounding box coordinates, and segment role indicators. The RCE-Attn mechanism independently captures row-wise and column-wise dependencies, mimicking human reading patterns in grid-like documents.

Comparison. FormNet achieves strong performance on tabular datasets such as FUNSD and CORD. However, it is highly specialized for form-like documents and underperforms on free-form or visually complex layouts. It does not include visual embeddings or generative capabilities, limiting its generality compared to Donut or DocLLM.

Strengths and Limitations: FormNet achieves state-of-the-art results on datasets like FUNSD and CORD, thanks to its inductive layout bias. It generalizes well in low-data regimes where tabular alignment is consistent. However, it is specialized for form-like documents and may underperform on free-form text or visually complex layouts like contracts. It does not incorporate image-level features or instruction-following capabilities, limiting its flexibility compared to models like Donut or DocLLM.

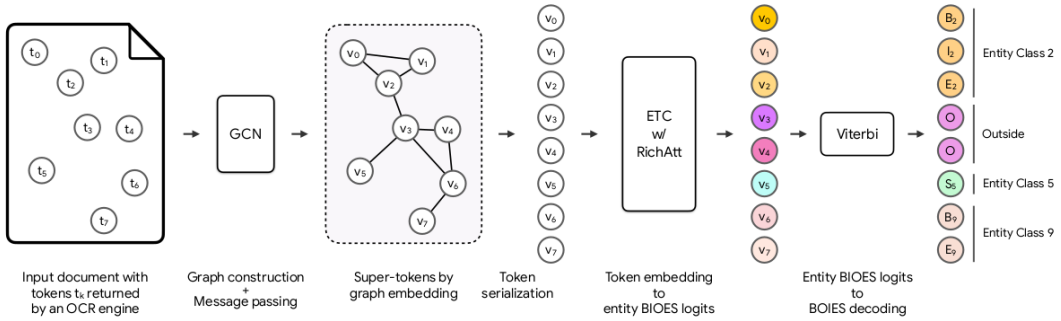


Figure 4: FormNet architecture [26]: Transformer encoder with row-column enhanced attention for table-like layouts.

4.1.4 Donut

Donut [16] represents a paradigm shift by introducing an OCR-free, end-to-end vision-to-sequence model. It directly processes document images using a Swin Transformer encoder and generates outputs using an autoregressive decoder, eliminating the need for OCR-based preprocessing.

Architecture and Vision-Centric Design. The input document is rendered as an image and divided into patches. These are embedded using a Swin Transformer and passed to a transformer decoder, which generates structured outputs such as classification tags, key-value pairs, or answers. Pretraining is performed on synthetic document data using SynthDoG [27].

Comparison. Donut is robust to OCR errors and flexible across tasks due to its generative design. It achieves state-of-the-art performance on datasets like CORD. However, as a decoder-only model, it lacks interpretability and is more prone to hallucination. It also incurs higher computational cost compared to encoder-based models like LayoutLMv2 or FormNet.

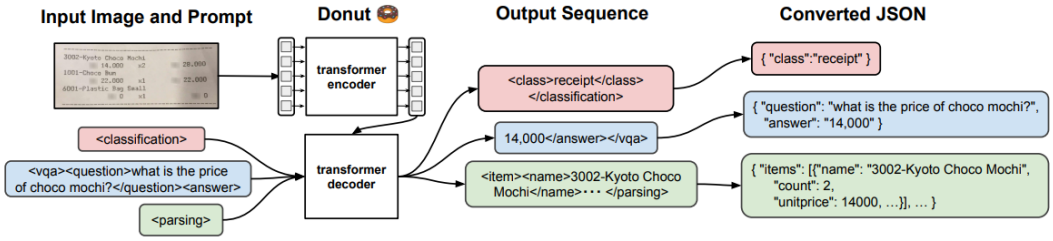


Figure 5: The Donut architecture [16]: An OCR-free vision-to-sequence model where a Swin Transformer encoder maps the document image to embeddings, and a decoder generates structured outputs such as key-value pairs or answers.

Strengths and Limitations: Donut demonstrates strong performance on datasets like CORD (96.9 F1), especially when OCR quality is poor or layout inconsistencies exist. Its ability to learn from rendered pixels enables robust document understanding without layout tokenization. However, being decoder-only, it lacks fine-grained token-level reasoning and may hallucinate outputs in visually ambiguous scenarios. Moreover, it is less interpretable than alignment-based models such as LayoutLM.

We next review representative LLM-based document understanding models, which extend the above techniques with instruction-following and generative capabilities.

4.2 LLM-Based Approaches

The emergence of instruction-tuned and multimodal large language models (LLMs) has redefined the landscape of document understanding. Unlike traditional models that rely on static pipelines and task-specific heads, LLMs offer dynamic, instruction-following capabilities with the potential to generalize across tasks, formats,

and domains. This section reviews representative LLM-based models, categorizing them by input modality and generative capacity while analyzing their architectural design and practical implications.

For completeness, we later compile reported zero-shot LLM baselines and document-LLM comparisons in Table 6 (from LayoutLLM) and Table 7 (from DocLLM), respectively.

Text-only IE (FLAN-T5/mT5). InstructUIE [15] (built on FLAN-T5/mT5) reframes IE tasks (NER/RE/EE) as instruction-following text-to-text generation. It reports strong results on *text-only* IE benchmarks, but since it does not use layout or vision inputs, we cite it as a text-only LLM baseline rather than a layout-aware document model.

4.2.1 InstructUIE

InstructUIE [15] is a lightweight, text-only encoder-decoder model based on mT5, fine-tuned to follow natural language instructions across tasks such as named entity recognition (NER), relation extraction, and classification. It frames document understanding as a text-to-text problem, where instructions and input text are concatenated and decoded into task-specific outputs.

While InstructUIE's zero-shot generalization and multilingual support make it highly portable, it lacks access to layout and visual information. As a result, its effectiveness declines in documents where spatial structure is critical. Compared to layout-aware LLMs like LayoutLLM or DocLLM, InstructUIE offers simplicity and low resource demands but trades off spatial reasoning and fine-grained control.

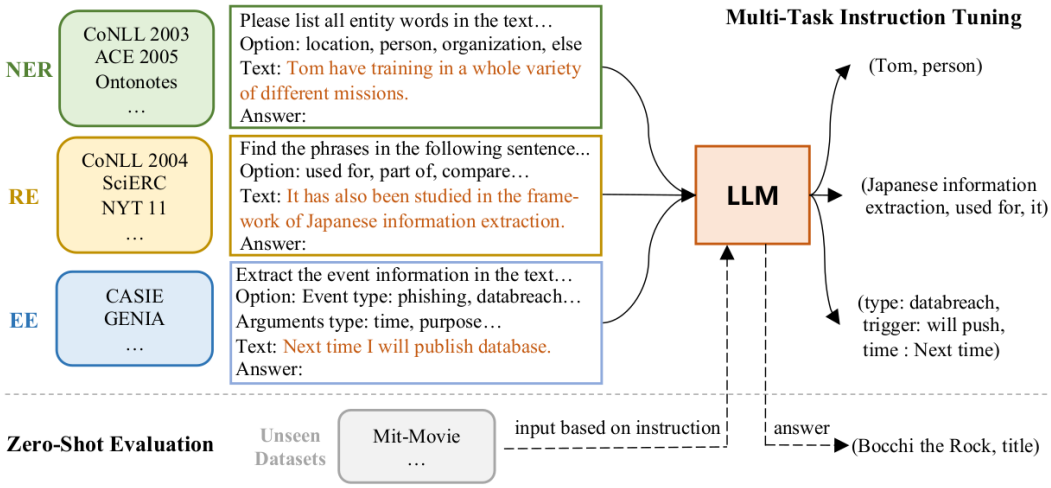


Figure 6: InstructUIE [15]: Instruction-tuned mT5 model for information extraction across multiple task types.

Strengths and Limitations: InstructUIE's main strength lies in its flexibility and portability. It supports a wide range of task formats and works well in few-shot scenarios without requiring retraining. The model is also multilingual, thanks to its mT5 backbone. However, it operates purely on text and does not incorporate any visual or layout information, which limits its effectiveness on visually complex documents. Additionally, its performance is sensitive to prompt phrasing and may degrade when instructions are ambiguous or misaligned with document structure.

Comparison: InstructUIE differs from models like LayoutLMv2 or DocFormer by omitting layout and image cues entirely. Compared to LLMs like LayoutLLM and DocLLM, which fuse layout information with prompts, InstructUIE is significantly more lightweight and deployable but less robust on layout-sensitive tasks. It serves as a strong baseline for purely text-driven document understanding in multilingual and low-resource settings.

4.2.2 LayoutLLM

LayoutLLM [28] introduces layout-awareness into decoder-only LLMs by rendering bounding box layouts as grayscale layout maps. These maps are passed through a vision encoder and fused with textual input, allowing the model to ground generation in spatial structure.

This design strikes a balance between instruction-following flexibility and layout sensitivity. Unlike full multimodal models such as DocVLM, LayoutLLM does not require full document images—only layout maps—reducing visual processing overhead. However, its reliance on handcrafted layout renderings introduces variance depending on preprocessing quality and resolution.

Architecture and Input Modalities: LayoutLLM accepts two inputs: (1) flattened text from OCR or PDF sources, and (2) a 2D layout map rendered as a grayscale image, where bounding boxes or regions are encoded visually to represent spatial structure. This layout map is processed through a lightweight vision encoder (e.g., CNN or ViT), whose outputs are linearly projected into embedding space and prepended to the token sequence. This allows the model to attend to both textual and spatial cues during generation. Instructions are issued in natural language prompts, making the system flexible across tasks.

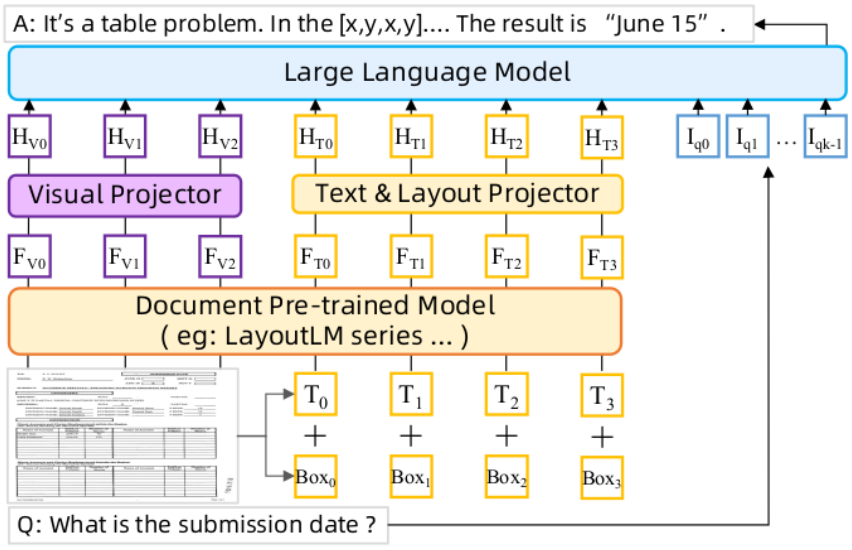


Figure 7: LayoutLLM [28]: A layout-aware instruction-following model that fuses layout maps with token embeddings for structured document tasks.

Training and Strengths: LayoutLLM is instruction-tuned using labeled datasets such as FUNSD, SROIE, CORD, and DocVQA. It supports end-to-end generation for tasks like structured extraction or classification. Its key strengths include robustness to OCR errors, thanks to layout supervision, and compatibility with standard decoder-only LLMs. The architecture requires minimal changes to existing LLMs, making it an efficient drop-in extension for document tasks.

Limitations and Comparison: LayoutLLM's effectiveness depends on the quality and expressiveness of the layout map, which currently lacks a standardized rendering protocol. The additional vision encoder also increases inference overhead. Compared to InstructUIE, LayoutLLM introduces layout understanding for improved accuracy on structured forms. While it lacks the full visual grounding of models like DocVLM or Donut, it offers a middle ground between pure-text instruction models and fully multimodal architectures.

4.2.3 DocLLM

DocLLM [4] builds on decoder-only transformers (e.g., LLaMA) by injecting token-level layout features such as bounding box coordinates directly into position embeddings. It supports flexible output formats via generative decoding and has shown state-of-the-art performance in key-value extraction and document QA.

Unlike LayoutLLM, which uses rendered layout maps, DocLLM discretizes layout numerically, simplifying the fusion pipeline. This makes it highly effective for structured generation while remaining lightweight.

However, the absence of visual embeddings limits performance in vision-heavy tasks such as form recognition or figure interpretation.

Architecture and Input Representation: DocLLM adopts a LLaMA-style decoder architecture with explicit layout embeddings. Each input token is embedded with its textual content and positional layout features, including x, y coordinates and bounding box dimensions. These layout features are discretized and incorporated into positional encodings to guide attention. Optionally, rendered document images can be processed into visual tokens and appended to enhance context.

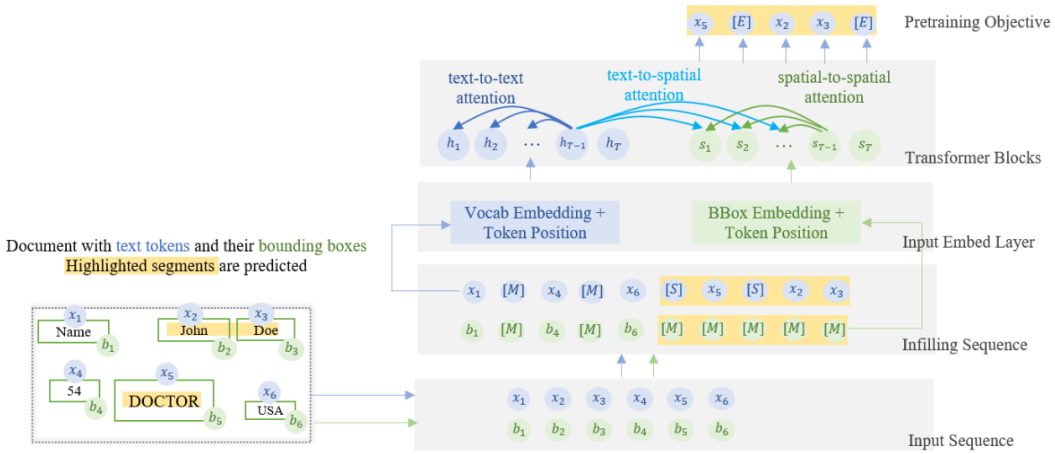


Figure 8: DocLLM [4]: A decoder-only transformer that incorporates layout and positional embeddings for structured generation across document tasks.

Training Objectives and Capabilities: DocLLM is trained on a large corpus of documents, including receipts, forms, and academic papers. Its multitask generative training allows it to handle key-value extraction, classification, and document QA within a single framework. The layout-guided decoding mechanism enhances spatial reasoning, enabling the model to handle irregular layouts and visually structured content more effectively.

Strengths and Limitations: DocLLM achieves state-of-the-art results on FUNSD and CORD, outperforming traditional models like LayoutLMv2 by a significant F1 margin. Its flexibility allows for task switching without architecture changes. However, being a decoder-only model, it incurs higher inference costs. The model also requires extensive pretraining data and may struggle with visually rich cues that are better captured by ViT-based encoders.

Comparison: Compared to encoder-based LayoutLMs, DocLLM provides stronger layout integration and generative versatility. It offers better layout resolution than InstructUIE and higher structural precision than Donut, while retaining compatibility with LLM instruction-following paradigms.

4.2.4 UDOP

UDOP (Unified Document Pretraining) [3] introduces a multimodal encoder-decoder framework designed to unify vision, text, and layout for document understanding. It reframes traditional document tasks—such as visual question answering (VQA), layout prediction, and classification—into a generative sequence-to-sequence setting using pretraining objectives inspired by T5.

Architecture and Input Representation: The model is built on a T5-style encoder-decoder architecture. Input consists of a rendered document image and its corresponding OCR tokens, which include layout coordinates. The image is processed by a vision transformer (e.g., ViT), while text tokens are overlayed on the image to create a combined multimodal representation. These image and token features are concatenated and passed through the encoder. The decoder then generates structured outputs—such as answers to document-related questions or predicted key-value pairs—guided by task-specific prompts.

Training Objectives and Capabilities: UDOP uses a combination of masked language modeling, visual-token alignment, and instruction-driven decoding. These objectives enable it to learn rich multimodal

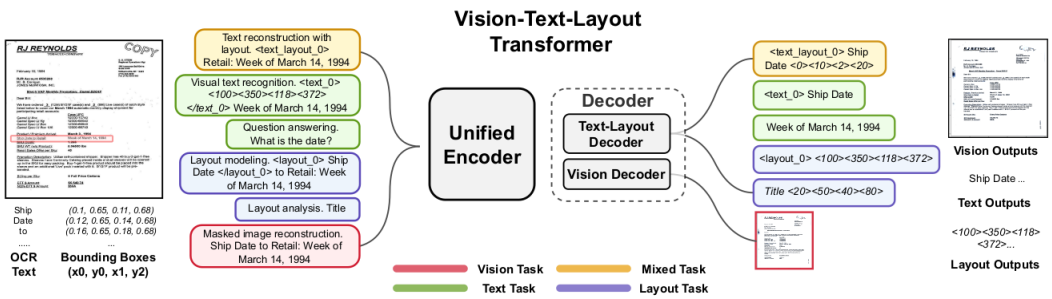


Figure 9: UDOP architecture [3]: Multimodal encoder-decoder unifying vision, text, and layout via rendered OCR-text over image inputs.

representations while generalizing across various downstream tasks. It supports key-value extraction, visual QA, and layout prediction using unified prompts in the decoder.

Strengths and Limitations: UDOP demonstrates strong performance on document QA benchmarks, particularly when spatial and visual reasoning is needed. Its encoder-decoder design supports flexible generative outputs across tasks. However, it is computationally intensive due to the large vision transformer and the multimodal fusion process. Additionally, OCR reliance introduces preprocessing complexity, and flattened multimodal fusion may not scale well to documents with highly irregular layouts.

Comparison: Compared to LayoutLMs, UDOP excels in multimodal generation, providing richer task representations. It is similar in architecture to Donut but benefits from OCR-token fusion and a dedicated decoding pipeline. While it lacks the layout-guided decoding precision of DocLLM, its encoder-decoder flexibility supports a wider range of tasks in a unified manner.

4.2.5 DocVLM

DocVLM [29] is a vision-language pretraining framework designed to enhance document understanding by aligning visual, textual, and layout modalities. It is particularly optimized for structured document question answering (DocQA) and form understanding, relying on contrastive pretraining and unified token-level alignment strategies.

Architecture and Input Representation: DocVLM adopts a multimodal encoder-decoder structure. The visual encoder, typically a ViT backbone, processes the rendered document image into patch embeddings. Meanwhile, OCR-extracted tokens are embedded alongside their corresponding bounding box coordinates, forming the textual-layout stream. These tokens are enhanced using "box prompts"—special embeddings that encode spatial layout information—and are fused with visual tokens to build a unified multimodal input representation. This representation is passed through a transformer-based encoder, optionally followed by a decoder for classification or generation tasks.

Training Objectives and Capabilities: DocVLM is trained using a combination of masked language modeling (MLM) for text prediction, contrastive patch-token alignment for cross-modal consistency, and box prompt learning to embed spatial awareness. These objectives enable fine-grained alignment between image regions and layout-aware text spans. During finetuning, the model supports multiple downstream tasks including document QA (DocVQA), infographic reasoning (InfographicVQA), and form classification.

Strengths and Limitations: DocVLM achieves state-of-the-art performance on several vision-language benchmarks. Its strength lies in the precise fusion of spatial and semantic information, enabled by its contrastive learning and coordinate embedding strategies. However, the use of large vision and text encoders makes training and inference computationally expensive. Furthermore, contrastive alignment depends heavily on high-quality OCR results and may degrade with noisy or low-resolution documents.

Comparison: Compared to OCR-free models like Donut, DocVLM integrates layout embeddings more explicitly and achieves stronger visual-text grounding. It improves upon encoder-only models like LayoutLMv2 by leveraging vision-text alignment through contrastive loss. Although less flexible than fully generative models such as DocLLM, DocVLM offers better fine-grained spatial grounding and excels in structured QA tasks.

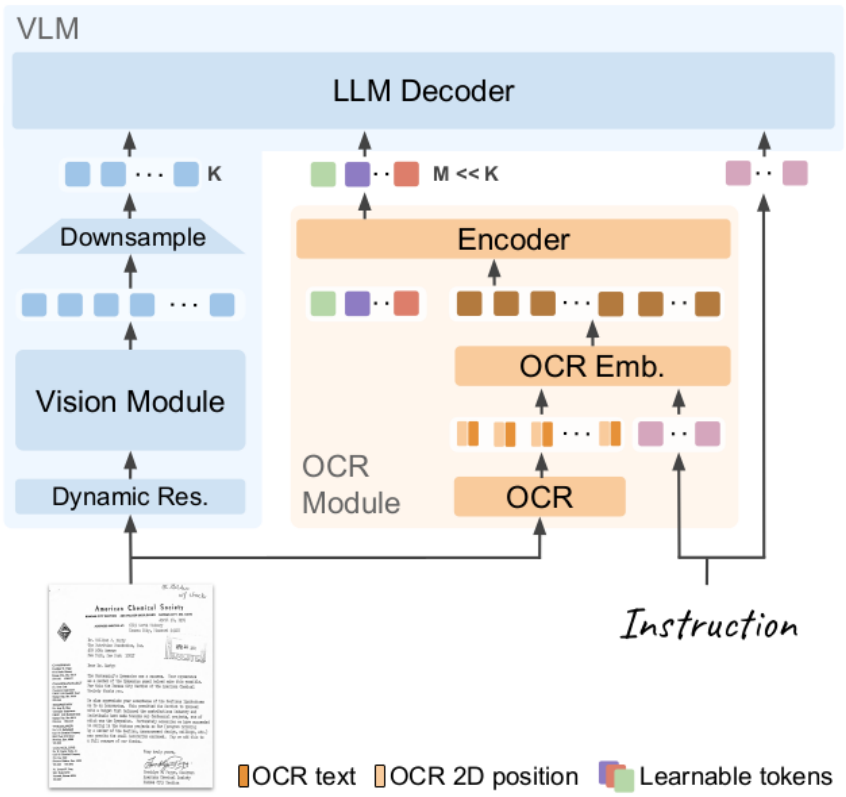


Figure 10: DocVLM architecture [29]: Unified encoding of visual patches and layout-aware text tokens with contrastive alignment for structured document QA.

4.2.6 MoE-LLaVA

MoE-LLaVA [5] is a Mixture-of-Experts (MoE) extension of the LLaVA (Large Language and Vision Assistant) model, designed to handle heterogeneous document understanding tasks across diverse domains such as receipts, charts, diagrams, and forms. It introduces domain-specific expert modules and a dynamic routing mechanism to adaptively select and combine the most relevant visual reasoning skills per input query.

Architecture and Routing Strategy: MoE-LLaVA retains the LLaVA framework's multimodal foundation, combining a CLIP-ViT-based vision encoder with a LLaMA-style decoder. However, instead of relying on a single vision-language pipeline, it incorporates multiple expert modules. Each expert consists of a vision adapter, a projector, and a routing head tailored to a specific document type or visual style. The router computes similarity between the image-question embedding and each expert's domain profile, activating a top-k subset of experts whose outputs are weighted and aggregated. This fused visual embedding is then fed into the language decoder, which generates answers or structured outputs.

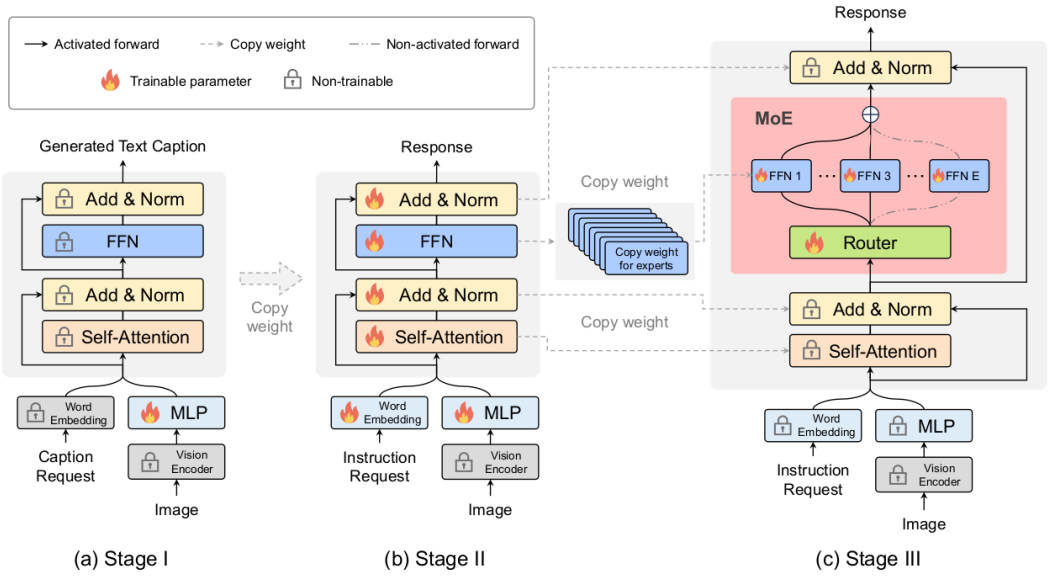


Figure 11: MoE-LLaVA architecture [5]: Mixture-of-Experts vision-language model for document QA with adaptive expert routing.

Training and Supported Tasks: Each expert is pretrained or fine-tuned on a domain-specific dataset, such as ChartQA, DocVQA, or form understanding, and the router is jointly optimized using reinforcement learning or supervised distillation to select experts that minimize task loss. MoE-LLaVA is evaluated on multiple document understanding benchmarks, showing superior performance in document visual question answering, document captioning, and layout-driven reasoning.

Strengths and Limitations: By leveraging expert specialization, MoE-LLaVA improves domain-specific performance while maintaining scalability across tasks. Its modular design also allows new experts to be added incrementally without retraining the entire model. However, the routing mechanism adds computational overhead, and improper expert selection can reduce performance stability. Furthermore, inference time and memory usage are higher than single-path models due to the parallel execution of multiple experts.

Comparison: MoE-LLaVA surpasses traditional vision-language models like DocVLM in adaptability, particularly for multi-domain document QA. Compared to monolithic models like UDOP or DocLLM, it achieves better task decomposition and modular scalability. Its expert routing strategy is especially beneficial when dealing with highly diverse document types or visually complex queries.

Figure 12 compares models based on parameter count and architectural complexity. Traditional non-LLM models such as LayoutLMv2 and DocFormer are relatively lightweight and efficient, while LLM-based architectures like DocLLM and MoE-LLaVA introduce significantly higher complexity due to their generative capabilities and multimodal inputs. This visualization helps highlight the trade-off between performance and computational cost, reinforcing the need for application-specific model selection.

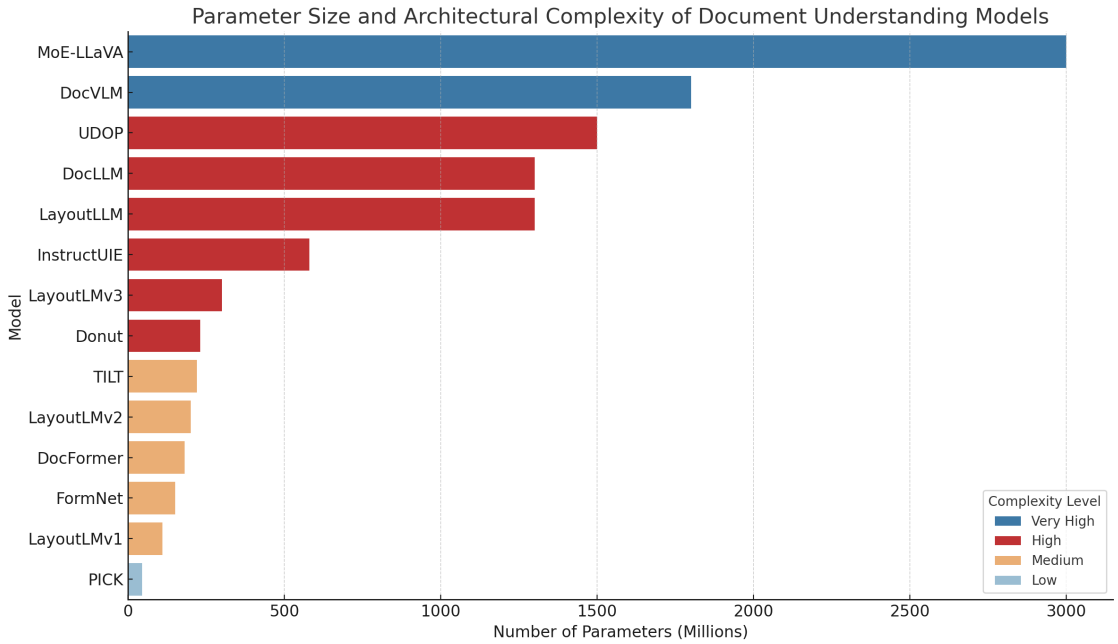


Figure 12: Comparison of document understanding models by parameter count and architectural complexity. LLM-based methods tend to have significantly higher parameter sizes and complexity due to instruction tuning and multimodal fusion.

We conclude this section by summarizing the capabilities and trade-offs of the surveyed models.

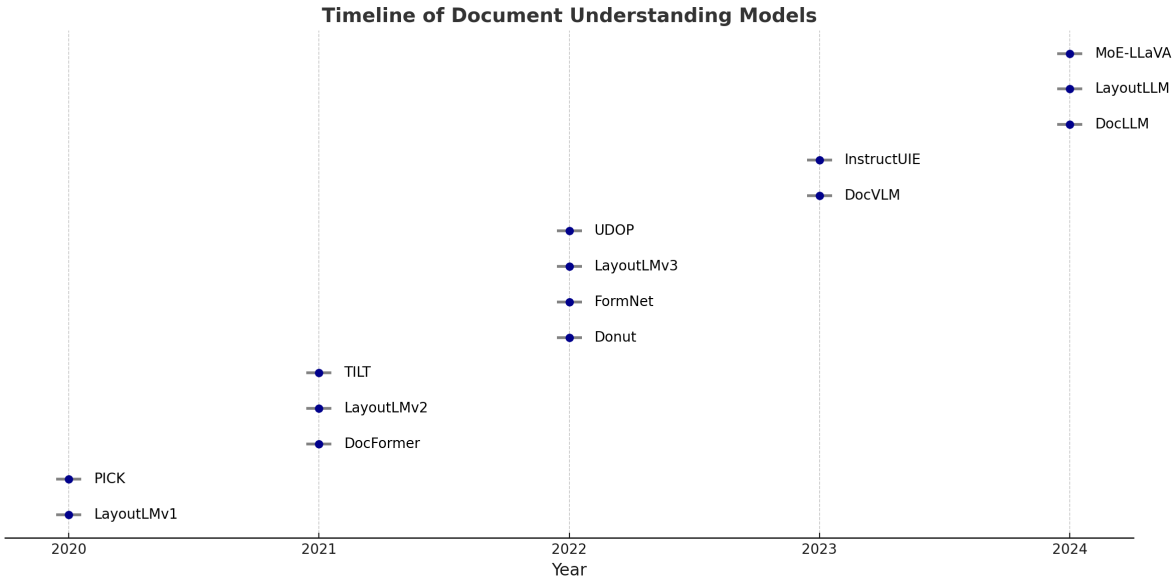


Figure 13: Chronological timeline showing the release of key document understanding models, highlighting the evolution from layout-aware encoders (e.g., LayoutLMv1) to instruction-tuned and multimodal LLMs (e.g., LayoutLLM, MoE-LLaVA). This progression illustrates the shift from static token-level modeling toward flexible, generative architectures.

Figure 13 illustrates the chronological evolution of document understanding models, starting from early layout-aware encoders like LayoutLMv1 and progressing toward multimodal, instruction-tuned LLMs such as

DocLLM and MoE-LLaVA. This timeline highlights key architectural shifts in the field—from token-based BERT-style models to generative and vision-language architectures. It also reflects the increasing complexity and flexibility of models over time, driven by the need for better layout generalization, multimodal grounding, and instruction-following capabilities.

Summary and Comparison

In summary, non-LLM approaches such as LayoutLMv1–v3, DocFormer, FormNet, and TILT have demonstrated strong performance in document understanding by incorporating layout and visual features into specialized transformer or graph-based architectures. These models excel in token-level classification tasks such as NER and KIE extraction, where structured layout encoding plays a critical role. However, they often require OCR preprocessing and are less flexible in adapting to new tasks or output formats.

In contrast, recent LLM-based models—including InstructUIE, LayoutLLM, DocLLM, UDOP, DocVLM, and MoE-LLaVA—rely on instruction tuning and generative modeling to enable greater generalization, adaptability, and multitask capabilities. Some incorporate layout information through rendered layout maps (LayoutLLM), embedded box coordinates (DocLLM), or contrastive alignment with visual encoders (DocVLM). Models like Donut and UDOP further push the boundary by operating OCR-free, leveraging end-to-end vision-to-text pipelines.

The key trend across both categories is a shift toward unified, multimodal models capable of end-to-end understanding with minimal task-specific engineering. While LLMs offer more flexibility and task coverage, non-LLM models still provide competitive performance with lighter architectures, making them suitable for resource-constrained or latency-critical deployments.

Having reviewed representative models across both non-LLM and LLM-based paradigms, we now synthesize key architectural patterns, modality integration strategies, and generative capabilities. Figure 13 illustrates the chronological development of document understanding models, while Table 3 contrasts core design choices across layout fusion, vision usage, and output formats.

Model	Architecture	Layout Fusion	Vision Input	Generative
LayoutLMv2	Encoder-only	BBox Embeddings + CNN	Yes	No
LayoutLMv3	Encoder-only	Joint ViT-Text Tokens	Yes	No
DocFormer	Encoder-only	Gated Fusion	Yes	No
FormNet	Encoder-only	Row-Column Attn	No	No
PICK	Graph-based	GAT over Text Boxes	No	No
Donut	Decoder-only	Vision-to-Seq	Yes	Yes
InstructUIE	Enc-Dec (mT5)	None (Text-only)	No	Yes
LayoutLLM	Decoder-only	Layout Map + Prompt	Optional	Yes
DocLLM	Decoder-only	Layout Embeddings	No	Yes
UDOP	Enc-Dec (T5)	ViT + OCR Tokens	Yes	Yes
DocVLM	Enc-Dec	Contrastive Fusion	Yes	Yes
MoE-LLaVA	MoE Dec.-only	Expert Routing	Yes	Yes

Table 3: Comparative summary of document understanding models across architecture, layout fusion, vision input, and generative capabilities.

Having reviewed model architectures in detail, we now synthesize key patterns, trade-offs, and paradigm shifts observed across model families.

5 Trends in Model Design

Over the past five years, document understanding models have evolved rapidly along three major design axes: modality integration (text, layout, vision), generative capacity (discriminative vs. generative), and task conditioning (hard-coded vs. prompt-based). In this section, we categorize models by their dominant architectural paradigm and highlight key design shifts shaping the field.

5.1 OCR-Based vs. OCR-Free Models

Most traditional document understanding systems rely on OCR to convert images into token sequences with bounding box coordinates. Models like LayoutLMv1–v3 [1, 2, 30], DocFormer [31], and FormNet [26] fall into this category, fusing textual embeddings with layout and, optionally, visual tokens. These models are highly effective on clean, structured inputs, particularly forms and receipts.

By contrast, OCR-free models like Donut [16] bypass text extraction entirely. They treat the document as a rendered image and perform sequence generation via a vision encoder and autoregressive decoder. OCR-free approaches avoid OCR noise and simplify pipelines but can struggle with token-level alignment and high-resolution visual reasoning.

5.2 Prompt-Based vs. Task-Specific Models

Traditional models are trained with fixed task heads (e.g., classification, token tagging) and require architecture-level changes for each new task. LayoutLMv2 and FormNet use such specialized decoders.

Instruction-tuned models like InstructUIE [15], LayoutLLM [28], and DocLLM [4] instead frame all document tasks as prompt-response pairs. These models support zero- and few-shot generalization and reduce the need for retraining. Prompt-based models also support dynamic output formats, making them more adaptable for real-world deployment.

5.3 Encoder-Only vs. Generative Architectures

Encoder-only architectures (e.g., LayoutLM, FormNet) perform well for classification and extraction tasks but lack generation capabilities. They are faster and lighter but limited in flexibility.

Decoder-only models like DocLLM and LayoutLLM support open-ended generation, including document QA, structured extraction, and free-form summarization. Encoder-decoder hybrids like UDOP [3] and DocVLM [29] attempt to unify both paradigms, supporting dense multimodal alignment with generative heads.

5.4 Vision-Text Fusion Strategies

Models differ significantly in how they integrate layout and visual information:

- **Late Fusion:** Separate encoders for vision (e.g., CNN or ViT) and text are fused at the transformer stage. Seen in LayoutLMv2.
- **Joint Embedding:** Text and image patches are embedded into a single token sequence, as in LayoutLMv3 and DocFormer.
- **Layout Map Rendering:** LayoutLLM uses grayscale layout maps as a pseudo-visual input to guide text generation.
- **Contrastive Fusion:** DocVLM aligns vision and layout-text tokens via contrastive learning, improving spatial-semantic consistency.
- **Expert Routing:** MoE-LLaVA routes visual encodings through task-specific expert modules.

Each strategy trades off accuracy, interpretability, and computational cost.

5.5

Summary of Design Trends

- Shift from rigid task-specific heads to prompt-driven multi-task learning.
- Movement away from pure OCR dependency toward image-first or hybrid pipelines.
- Emphasis on generative decoding for open-ended tasks like document QA.

- Increasing reliance on instruction tuning to support task generalization.
- Exploration of novel fusion techniques for layout and vision grounding.

6 Comparative Benchmark Analysis

This section presents a comparative analysis of LLM-based and non-LLM document understanding models across four key tasks: Key Information Extraction (KIE), named entity recognition (NER), document classification, and document question answering (DocQA). We summarize performance using standard benchmarks and highlight critical trends in generalization, robustness, and adaptability.

Table 5 consolidates representative results, compiled from original publications and official model repositories.

Caveats in Benchmark Comparisons. Although the analysis strives for fairness, it is important to recognize that reported results are drawn from heterogeneous sources. Differences in OCR engines, prompt design, input tokenization, and model configurations (e.g., base vs. large) influence outcomes. Where relevant, such discrepancies are annotated via footnotes or markers. Numerical comparisons should thus be viewed as directional rather than absolute.

Model Paradigm Overview

We categorize models into four dominant classes: encoder-only models (e.g., LayoutLM [2], DocFormer [31]); encoder-decoder models (e.g., TILT [32], UDOP [3]); decoder-only LLMs (e.g., DocLLM [4], LayoutLLM [28]); and fully multimodal LLMs (e.g., DocVLM [29], MoE-LLaVA [5]). Table 4 provides a summary of task coverage across these models.

Model	KIE	NER	Cls.	DocQA
LayoutLMv2 [2]	✓	✓	✓	✓*
Donut [16]	✓	✓	✓	✓
DocLLM [4]	✓	✓	✓	✓
LayoutLLM [28]	✓	✓	✓	✓
MoE-LLaVA [5]	✓	✓	✓	✓
InstructUIE [15]	✓	✓	✓	✓*
DocVLM [29]	✓	✓	✓	✓

Table 4: Task coverage across representative document understanding models.
Note: $*$ indicates limited or indirect support. LayoutLMv2 lacks native QA capabilities; InstructUIE is unimodal.

Table 4 illustrates that recent LLM-based models, including DocLLM, LayoutLLM, and MoE-LLaVA, deliver comprehensive task support across KIE, NER, classification, and DocQA. In contrast, older unimodal architectures exhibit narrower applicability, especially for QA.

6.1 Key Information Extraction (KIE)

Evaluated on FUNSD [6], SROIE [9], and CORD [7], layout-aware LLMs like DocLLM [4] and LayoutLLM [28] demonstrate clear superiority over encoder-based models. DocLLM achieves 91.2 F1 on FUNSD, outperforming LayoutLMv2 (84.2) and Donut (87.6). Donut's OCR-free design is especially effective on layout-heavy datasets like CORD. Text-only models like InstructUIE perform competitively in simple scenarios but struggle under layout complexity.

6.2 Named Entity Recognition (NER)

Token-level F1 scores on FUNSD, SROIE, and XFUND [8] reveal that layout-aware LLMs consistently excel. DocLLM and LayoutLLM both surpass 98 F1 on SROIE and maintain over 90 F1 on FUNSD. On multilingual XFUND, LLMs generalize well, whereas encoder models such as LayoutLMv2 exhibit sharp performance drops in zero-shot settings.

6.3 Document Classification

Performance on RVL-CDIP [10] and Tobacco3482 [11] further underscores the efficacy of LLMs. DocLLM and UDOP [3] reach 97.5% accuracy on RVL-CDIP, exceeding LayoutLMv2. Domain adaptation with limited tuning remains a strong suit of LLMs, although traditional models still hold ground in high-resource domains.

6.4 Document Question Answering (DocQA)

In vision-intensive benchmarks like DocVQA [12] and InfographicVQA [13], models such as MoE-LLaVA and DocVLM lead by a 4–5 point margin, validating the benefit of multimodal pretraining. LayoutLLM and UDOP remain competitive when supplied with rendered layout maps, though they trail behind in visually dense contexts. For DocVQA we use Average Normalized Levenshtein Similarity (ANLS); ANLS applies a 0.5 thresholded Levenshtein penalty to handle OCR noise.

Benchmark Summary Table

Model	KIE (FUNSD)	NER (SROIE)	Cls. (RVL)	DocQA (DocVQA; ANLS↑)
LayoutLMv2 [2]	84.2	96.2	95.3	70.2
Donut [16]	87.6	97.0	95.7	73.2
DocLLM [4]	**91.2**	**98.3**	**97.5**	73.4
LayoutLLM [28]	90.5	98.1	97.2	72.8
MoE-LLaVA [5]	–	–	96.8	**75.5**

Table 5: Benchmark accuracy and F1-scores across representative tasks and datasets: key information extraction (FUNSD [6]), named entity recognition (SROIE [9]), document classification (RVL-CDIP [10]), and document QA (DocVQA [12]). DocQA reported as ANLS where applicable; other columns are F1/accuracy—as reported by source papers. *Results are reported as-is from source papers.*

To avoid metric mixing, zero-shot generic LLM baselines (LLaMA/Vicuna) are reported separately in Table 6 (ANLS), and DocLLM vs. LLaMA2+OCR comparisons appear in Table 7. Values are quoted verbatim from the source papers.

Model / Prompting	FUNSD (QA)	CORD (QA)	SROIE (QA)	DocVQA
LLaMA2-7B (Plain Text)	40.78	4.39	15.86	61.34
LLaMA2-7B-chat (Plain Text)	48.20	47.70	**68.97**	64.99
Vicuna-7B (Plain Text)	49.79	44.67	67.49	61.39
Vicuna-1.5-7B (Plain Text)	48.06	51.40	68.20	**66.99**
LLaMA2-7B (layout-text)	51.40	28.04	34.96	37.32
LLaMA2-7B-chat (layout-text)	58.34	50.93	51.15	56.55
Vicuna-7B (layout-text)	42.73	46.59	45.43	37.21
Vicuna-1.5-7B (layout-text)	**59.63**	**56.13**	66.20	56.81

Table 6: Zero-shot LLM baselines reported by LayoutLLM [28]. Metric is ANLS ↑. "QA-for-VIE" converts VIE benchmarks to QA form (hence ANLS).

All values from LayoutLLM Table 1; zero-shot prompting; ANLS not directly comparable to F1/accuracy in Table 5.

Table 8: Comparison of commercial vs. open LLMs on SDU benchmarks. Values are F1 ↑ for FUNSD and CORD, and ANLS ↑ for DocVQA. Commercial LLM metrics are qualitative estimates due to lack of public benchmarks.

Model	FUNSD	CORD	DocVQA
LayoutLMv3	92.1	96.0	88.5
UDOP (V+T+L)	91.6	92.0	90.2
LayoutLLM-7B	80.0	89.8	87.0
DocVLM	–	–	91.2
DocLLM-7B (SDDS)	51.8	83.2	65.7
InstructUIE	–	85.0	–
LLaMA-2 + OCR	17.8	–	–
GPT-4 (API)	∼55–60	∼90–93	∼92–94
Claude 3 / Gemini 1.5	N/A	N/A	N/A

Note. See Section 7.1 for task-wise trends; estimates (∼) reflect qualitative vendor reports.

Layout-text sometimes helps VIE but can hurt DocVQA because of prompt length/format sensitivity. *DocLLM vs. LLaMA2+OCR (mixed KIE F1 and DocVQA ANLS) as reported by DocLLM is shown in Table 7.*

Model (setting)	FUNSD (F1)	CORD (F1)	SROIE (F1)	DocVQA (ANLS)
LLaMA2+OCR (ZS)	17.8	13.8	56.4	47.4
DocLLM-1B (SDDS)	48.2	66.9	91.0	61.4
DocLLM-7B (SDDS)	51.8	67.4	91.9	69.5

Table 7: DocLLM vs. LLaMA2+OCR (as reported in DocLLM [4]). KIE datasets use F1 ↑; DocVQA uses ANLS ↑. Settings (ZS/SDDS) follow the paper and are not directly comparable to zero-shot LayoutLLM.

Table 5 synthesizes performance across tasks, reaffirming the superiority of instruction-tuned and layout-aware LLMs. Notably, MoE-LLaVA excels in DocQA, reflecting the power of modular expert routing in complex multimodal tasks.

Discussion. Across all three benchmarks, open-source LLMs have closed much of the historical gap with fine-tuned encoder models. LayoutLMv3 and UDOP remain strong supervised baselines, achieving above 90 F1 on FUNSD and CORD and over 90 ANLS on DocVQA. Among zero-shot methods, LayoutLLM 7B substantially improves over text-only LLaMA-2 by more than 60 F1 points on FUNSD, highlighting the impact of layout-aware instruction tuning. DocLLM demonstrates that incorporating geometric structure yields consistent gains, though performance remains below supervised models. Commercial systems such as GPT-4 reportedly surpass open models on document reasoning tasks, but the absence of reproducible benchmark data limits direct comparison. Overall, the results indicate that open, layout-aware multimodal LLMs now approach commercial-grade performance while maintaining transparency and research accessibility.

6.5 Benchmarking Protocol

This survey aggregates published results without reproducing experiments. To ensure clarity and consistency, we follow these principles:

- **Source Integrity**: All scores are cited from primary papers or official repositories.
- **No Post-hoc Adjustments**: Metrics are reported unaltered. We apply no normalization or re-scaling.
- **Best Variant Selection**: When multiple variants exist (e.g., base vs. large), we report the highest-performing one.

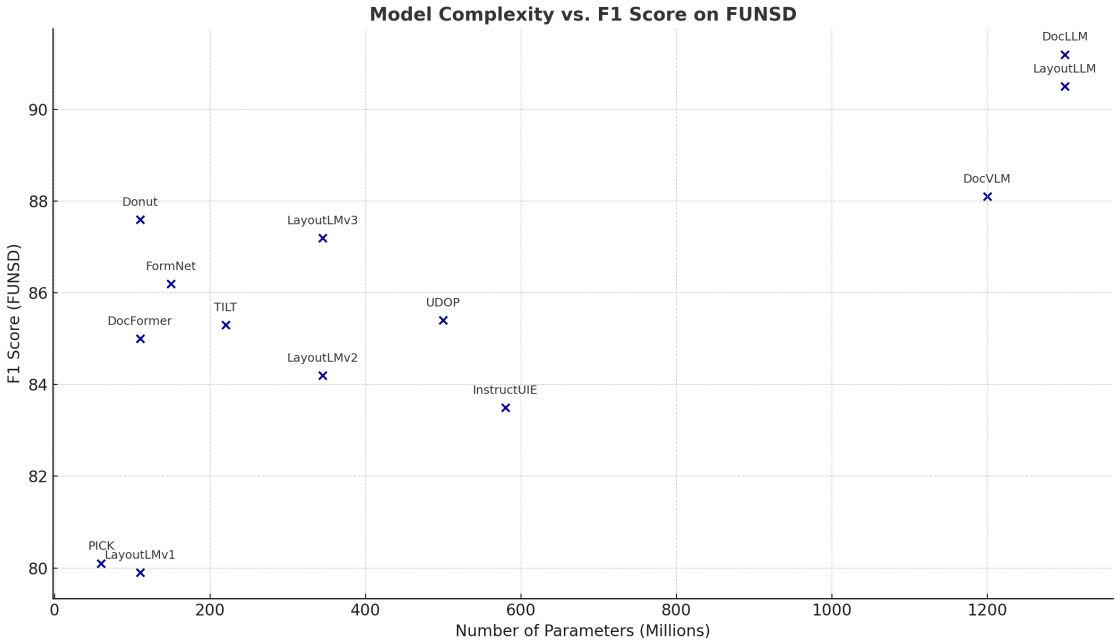


Figure 14: Model complexity vs. F1 on FUNSD. Layout-aware and multimodal LLMs show strong performance gains, though with higher parameter counts.

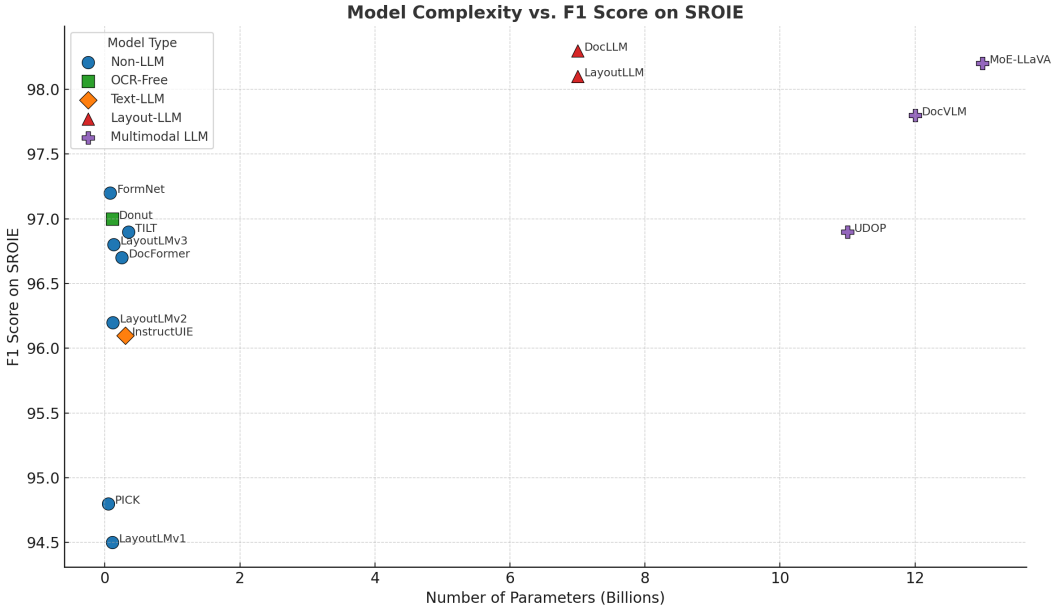


Figure 15: Parameter count vs. F1 on SROIE. Multimodal and layout-aware models deliver the best accuracy, but incur substantial computational cost.

- **Inference Complexity**: We include parameter counts and latency where disclosed, but conduct no new benchmarking.

This methodology enables an authoritative and transparent synthesis of existing research, highlighting the evolving trade-offs in structured document understanding.

7 Reported Performance Trends and Discussion

We analyze the performance of representative models across four major document understanding tasks. A unified visual summary is shown in Figure 16, which presents comparative bar plots for each task: (a) key-value extraction, (b) named entity recognition, (c) classification, and (d) document question answering (DocQA). These plots enable side-by-side comparison of performance across architectures and modalities.

7.1 Performance Trends by Task

We evaluate model performance across four document understanding tasks: key-value extraction, named entity recognition (NER), document classification, and document question answering (DocQA). A unified visual summary is provided in Figure 16, and qualitative outputs are included for representative models.

Key Information Extraction (KIE). As shown in Figure 16a, layout-aware LLMs such as LayoutLLM and DocLLM outperform earlier encoder-based models like LayoutLMv2 and FormNet on datasets like FUNSD and CORD. This improvement is largely due to generative decoding capabilities and stronger spatial context modeling, which enable accurate linking of distant or misaligned key-value pairs. Donut, an OCR-free model, also performs strongly on CORD, indicating that visual sequence generation can be effective in structured layouts.

Named Entity Recognition (NER). Figure 16b shows that decoder-based LLMs, particularly LayoutLLM and DocLLM, achieve state-of-the-art F1 scores on SROIE and XFUND. These models excel at multilingual generalization and token-level precision due to instruction tuning and broader pretraining. Traditional models like LayoutLMv2 and FormNet exhibit performance drop-offs when faced with unseen scripts or complex layouts.

Document Classification. As illustrated in Figure 16c, nearly all models surpass 95% accuracy on RVL-CDIP, suggesting dataset saturation. However, on Tobacco3482—a domain-specific dataset—DocLLM and UDOP exhibit better generalization, likely due to their instruction-following and generative capacities. Encoder-based models plateau or overfit due to narrower task conditioning.

Document QA. Figure 16d highlights the superior performance of vision-language models like MoE-LLaVA and DocVLM on DocVQA and InfographicVQA. Their architecture enables cross-modal reasoning using contrastive learning and adaptive visual experts. These models outperform traditional architectures like LayoutLMv2 by 4–5 F1 points, particularly in visually complex layouts requiring hierarchical understanding.

Limitations of Comparison. The benchmark results reported in this paper are aggregated from original model papers, public repositories, or available evaluations using pretrained checkpoints. While care was taken to ensure consistency, direct comparisons may be affected by discrepancies in experimental setups. Factors such as OCR engine quality, input preprocessing pipelines, prompt phrasing (for instruction-tuned models), model variant (e.g., base vs. large), and token/image resolution can introduce variability. We flag these limitations wherever applicable and encourage readers to interpret trends comparatively rather than as absolute rankings. Table 12 provide additional configuration and prompt details.

7.2 Architectural Implications

Encoders (LayoutLM series) are highly effective for token-level tasks such as named entity recognition and key-value extraction, owing to their integration of textual and spatial embeddings. These models are layout-sensitive and excel when provided with clean, OCR-extracted tokens and precise bounding boxes. However, they require rigid preprocessing pipelines, and their reliance on structured inputs limits their flexibility for end-to-end or generative tasks.

OCR-free models (Donut) offer a simplified and more robust alternative by directly processing rendered document images. This removes the dependency on external OCR systems and alleviates issues related to token misalignment or layout noise. While OCR-based methods rely on text extraction pipelines that can introduce noise or lose structural fidelity, models like Donut [16] process the document as an image and generate outputs directly in a vision-to-sequence fashion. This design enables robustness in visually diverse or low-quality scanned documents, although it requires significant visual pretraining and may lack fine-grained token-level alignment for certain tasks.

7.3 Modality Trade-Offs

LLM-based models exhibit superior task adaptability and require fewer labeled examples, making them ideal for generalization and transfer across diverse document understanding tasks. However, they may underperform in highly structured environments such as forms and tables if input alignment—whether via layout maps or token ordering—is not carefully managed.

Text-only models (InstructUIE[15]) offer a lightweight and deployable solution that is inherently language-agnostic and robust across domains. These models are particularly effective in scenarios where only raw textual input is available. However, their lack of access to layout information or visual features limits their performance in structure-sensitive tasks, such as key-value pair extraction or form-based question answering.

Layout-aware models incorporate spatial information through bounding box embeddings or rendered layout maps, significantly improving their ability to capture the hierarchical and positional structure of documents. These models achieve state-of-the-art results in tasks that require understanding document geometry, but they are sensitive to OCR errors and require well-aligned bounding box inputs for optimal performance.

Image-based models (Donut, MoE-LLaVA) process the document as a visual input, making them inherently resilient to noisy OCR outputs and irregular token layouts. These models excel in vision-language tasks and can handle complex visual structures such as tables, charts, or handwritten forms. However, they demand greater computational resources and longer training cycles, making them less accessible in resource-constrained settings.

7.4 Dataset Bias and Evaluation Challenges

RVL-CDIP and FUNSD are widely used but may no longer differentiate between new models due to ceiling effects. Future work should emphasize more diverse datasets like InfographicVQA, ChartQA, and XFUND for stress-testing layout and vision reasoning.

Evaluation Metrics. Traditional F1 and accuracy do not capture semantic alignment or reasoning ability. For example, in QA tasks, exact match may be too strict. Metrics like ANLS (Average Normalized Levenshtein Similarity) are more reflective of comprehension.

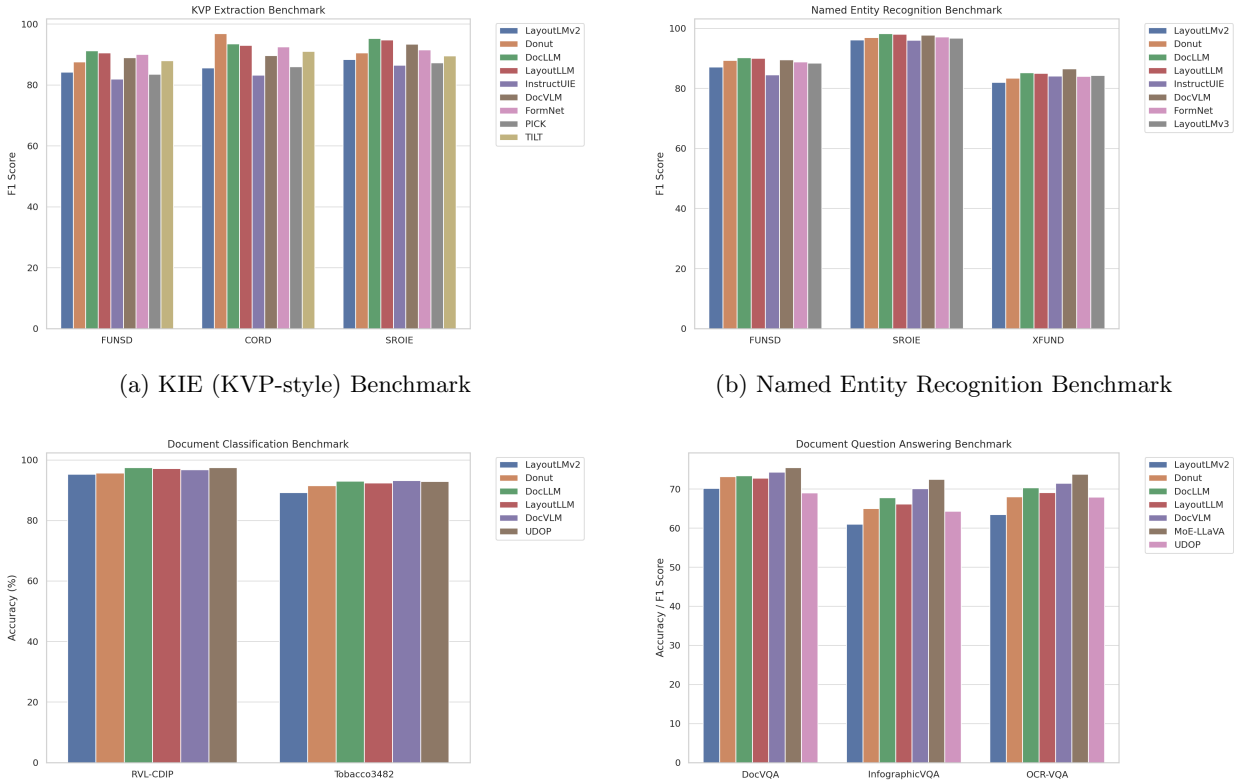
7.5 Scalability and Efficiency

Decoder-based LLMs are powerful but computationally expensive. While DocLLM shows superior accuracy, its inference latency and memory usage are significantly higher than LayoutLM or InstructUIE. MoE architectures mitigate some cost via routing but introduce complexity in training.

Summary

Large language model (LLM)-based approaches have reached or surpassed the performance of traditional non-LLM baselines across most structured document understanding tasks. Among these, layout-aware LLMs demonstrate the most effective balance between understanding document structure and maintaining high task performance. Despite these advances, non-LLM methods still offer practical value, particularly in scenarios with strict latency requirements or limited computational resources, where lightweight architectures remain competitive.

We visualize the performance of state-of-the-art models across four core tasks in Figure 16, which presents bar plots comparing F1 or accuracy scores on each dataset. This multi-panel figure helps highlight trends in task specialization, generalization, and the trade-off between model complexity and performance.



(a) KIE (KVP-style) Benchmark

(b) Named Entity Recognition Benchmark

(c) Document Classification Benchmark

(d) Document Question Answering Benchmark

Figure 16: Performance of document understanding models across four tasks: (a) key information extraction (KIE; F1), (b) named entity recognition (F1), (c) document classification (accuracy), and (d) document QA (accuracy/F1). *Note: Results are taken from original papers and may vary due to differences in prompt phrasing, token limits, OCR quality, or evaluation protocols.*

Figure 16 provides a unified comparison of model performance across four core document understanding tasks—key-value extraction, named entity recognition (NER), document classification, and document question answering (DocQA). In subfigure 16a, layout-aware LLMs like DocLLM and LayoutLLM consistently outperform earlier encoder-only models on KIE datasets such as FUNSD and CORD, demonstrating the effectiveness of spatially-informed generative modeling. Subfigure 16b reveals similarly strong trends in NER, where LayoutLLM and DocLLM surpass traditional models on benchmarks like SROIE and XFUND, with notable advantages in multilingual generalization. For document classification (subfigure 16c), most models achieve saturated performance on RVL-CDIP, but instruction-tuned models such as UDOP and DocLLM maintain superior accuracy under low-resource conditions like those in Tobacco3482. Finally, subfigure 16d highlights the dominance of multimodal LLMs—including MoE-LLaVA and DocVLM—on DocVQA, benefiting from richer visual-text alignment and adaptive routing mechanisms. Together, these

results emphasize the growing gap in performance between LLM-based systems and earlier transformer or graph-based architectures, especially in tasks that require multimodal reasoning or generalization across layouts and domains.

7.6 Failure Modes and Trade-offs

While many models show strong performance across standard benchmarks, practical deployment surfaces several limitations:

Hallucinations. Decoder-only models like Donut and DocLLM may generate plausible but incorrect outputs, particularly under ambiguous layouts or noisy OCR.

Prompt Sensitivity. Instruction-tuned models such as InstructUIE and LayoutLLM exhibit high sensitivity to prompt phrasing. For instance, on a sample invoice, LayoutLLM returns the correct value when prompted with "What is the invoice total?" but fails when asked "Extract the total amount." Such discrepancies highlight the brittleness of instruction-following models in production settings.

To mitigate prompt instability, several strategies can be applied:

- **Prompt Paraphrasing:** Generating lexical variants of the instruction helps test and stabilize model outputs.
- **Prompt Ensembles:** Aggregating results from multiple prompt formulations using voting or confidence scores.
- **Retrieval-Augmented Prompts:** Including few-shot examples or similar samples retrieved from a database improves grounding.
- **Self-Consistency Decoding:** Sampling multiple generations and choosing the most consistent answer improves reliability.

Generalization. Traditional models like LayoutLMv2 perform poorly on unseen layouts or languages without extensive fine-tuning.

Efficiency vs. Accuracy. Models like FormNet offer efficient inference but lack multimodal grounding. In contrast, MoE-LLaVA achieves high accuracy but incurs high inference cost.

Table 9 summarizes these trade-offs.

Model	Failure Mode	Use Case Strength	Efficiency
Donut	May hallucinate under noise	OCR-free, layout-agnostic	Medium
LayoutLMv2	Rigid input requirements	High accuracy on forms	High
DocLLM	High compute demand	General-purpose generation	Low
InstructUIE	Prompt phrasing sensitive	Lightweight, multilingual	Very High
MoE-LLaVA	Expert misrouting	Domain transfer	Low

Table 9: Failure modes, strengths, and efficiency trade-offs for selected models. *Descriptions based on paper-reported behaviors and qualitative observations; efficiency is reported qualitatively.*

8 Trade-offs Across Model Families

The landscape of document understanding models reflects a spectrum of design choices, each offering different strengths and limitations depending on task type, document structure, and deployment constraints. This section provides a comparative analysis of trade-offs across model families.

8.1 Layout Bias vs. Instruction Tuning

Traditional layout-aware models such as LayoutLMv2 and FormNet encode spatial structure via bounding box embeddings or inductive biases (e.g., row-column attention). These models perform well on structured forms where geometry reflects semantics. However, they lack adaptability to new tasks or domains without retraining.

In contrast, instruction-tuned LLMs such as DocLLM and InstructUIE operate via task prompts, offering greater flexibility and zero-shot capabilities. While these models support multitask scenarios and open-ended outputs, they may suffer from prompt sensitivity and underperform in rigidly structured layouts without layout grounding.

8.2 Encoder-Only vs. Generative Architectures

Encoder-only models (e.g., LayoutLMv2, FormNet) are efficient and interpretable, producing token-level predictions suited for extraction tasks. However, they require separate decoders per task and cannot support free-form output formats.

Generative models (e.g., DocLLM, Donut) unify tasks under a single architecture via sequence generation. They support more natural interaction paradigms (e.g., QA, summarization) but are computationally expensive and more prone to hallucinations, especially in ambiguous layouts.

8.3 OCR-Based vs. OCR-Free Pipelines

OCR-based models depend on text extraction quality. While layout-aware encoders rely on bounding boxes and tokenized inputs, they are sensitive to OCR noise and token alignment issues.

OCR-free models like Donut and MoE-LLaVA remove this dependency by directly encoding rendered document images. This makes them robust to noisy scans and token misalignment, but limits their ability to output token-aligned labels and increases training complexity.

8.4 Vision-Language Fusion Strategies

Fusion strategies impact both expressiveness and computational cost:

- **CNN/ResNet backbones** (LayoutLMv2) offer fast fusion but limited spatial alignment.
- **Joint token embedding** (LayoutLMv3, DocFormer) enables seamless integration at the token level.
- **Rendered layout maps** (LayoutLLM) provide lightweight layout grounding.
- **Contrastive alignment** (DocVLM) improves visual-semantic consistency.
- **Expert routing** (MoE-LLaVA) optimizes domain-specific performance at inference cost.

8.5 Model Size vs. Efficiency

Models like DocLLM and MoE-LLaVA deliver state-of-the-art performance but incur high latency and GPU memory usage. In contrast, lightweight models like InstructUIE and FormNet offer efficient inference and smaller footprints, making them suitable for edge or low-latency applications.

Summary

There is no universally optimal model. Trade-offs must be considered across:

- **Accuracy vs. speed**
- **Flexibility vs. structure bias**
- **OCR-dependence vs. vision robustness**
- **Token alignment vs. generative flexibility**

Table 9 and Table 3 complement this discussion by summarizing model-specific behaviors and architectural strategies.

While Section 7 focused on task-specific outcomes and model behaviors, we synthesize cross-cutting design trends and architectural trade-offs in Section 8, providing a broader perspective on model suitability and deployment implications.

8.6 Evaluation Metrics and Limitations

Document understanding tasks are evaluated using a variety of metrics depending on task type—yet many of these do not fully capture semantic correctness or structural fidelity. In this section, we briefly critique the suitability and limitations of common metrics used in the literature.

- **F1-score**: Widely used for key-value extraction and NER. While it measures overlap at the token or entity level, it fails to reflect semantic equivalence when field boundaries shift or synonyms are used (e.g., "Due Date" vs. "Invoice Date").
- **Exact Match (EM)**: Often applied to structured extraction and QA. EM is overly strict—small formatting or tokenization differences (e.g., "$421.56" vs. "421.56 USD") are penalized despite semantic equivalence.
- **ANLS (Average Normalized Levenshtein Similarity)**: Used in DocVQA to provide partial credit based on edit distance. It captures surface similarity but not semantic correctness, and can be inflated by padding or template repetition.
- **BLEU and ROUGE**: Used in generative settings such as document captioning or summarization. These metrics emphasize n-gram overlap, which can penalize syntactic variation and fail to reflect factual accuracy.
- **Structural Accuracy**: Rarely reported, but critical for tasks like table extraction or form parsing. Field-level or schema-level metrics (e.g., hierarchical alignment) are more appropriate but lack standard adoption.

Most generative models are evaluated using metrics originally designed for classification or string overlap. This introduces challenges in assessing models that hallucinate plausible but incorrect outputs, or those that produce correct answers with slightly different wording.

Recommendations. Future work should standardize evaluation pipelines by:

- Incorporating semantic-aware metrics such as BERTScore or referential consistency.
- Using human evaluation or task-specific validators (e.g., schema-checkers for KIE).
- Reporting per-field and per-type accuracy (e.g., date vs. amount fields).
- Visualizing failure cases, not just scores.

We advocate for evaluation protocols that better reflect end-user utility, robustness to ambiguity, and output interpretability in document intelligence systems.

9 Future Directions and Open Challenges

While structured document understanding has made notable strides, the field is far from mature. This section outlines decisive next steps and strategic challenges that, when addressed, will significantly accelerate progress and deployment readiness.

9.1 Unified Multimodal Instruction-Tuned Models

Instruction tuning has proven highly effective in aligning LLMs with downstream tasks, yet current models remain largely trained on unimodal, text-centric corpora. Advancing structured document understanding demands a shift toward large-scale, multimodal instruction datasets that reflect real-world document heterogeneity. Future architectures must go beyond text and integrate layout-aware prompting and visual context natively. Techniques such as chain-of-thought [33] and intermediate-step reasoning hold strong promise for enhancing performance in complex DocQA scenarios.

9.2 Multilingual and Cross-Domain Generalization

Performance disparities across languages and domains are no longer tolerable for globally deployed document AI systems. Models still exhibit significant degradation on non-English documents (e.g., XFUND [8]) and domain-specific content like legal or medical text. The path forward requires cross-lingual pretraining, unified embedding spaces, and domain adaptation protocols that scale robustly across both linguistic and structural variations.

9.3 Commercial vs. Open-Source LLMs for Document AI

The rapid progress of document-centric large language models (LLMs) has been driven by both closed commercial systems (e.g., GPT-4, Claude 3, Gemini 1.5) and open-source counterparts (e.g., LLaMA 3, Mistral, Phi 3, LayoutLLaMA, DocLLM). Although these families share architectural foundations, their accessibility, transparency, and reproducibility profiles differ substantially.

Commercial Models. Proprietary LLMs generally achieve the highest benchmark accuracy and exhibit superior multilingual and multimodal reasoning thanks to large-scale private training corpora and instruction data. However, the opaque nature of their data sources, prompt filters, and safety interventions limits scientific reproducibility. Furthermore, restricted API access prevents fine-grained ablations on layout-aware inputs or document-specific reasoning steps, making rigorous evaluation across SDU benchmarks (FUNSD, CORD, DocVQA) infeasible for the research community.

Open-Source Models. Openly released architectures such as LLaMA 3, Mistral, Phi 3, and specialized variants like LayoutLLM and DocLLM enable transparent experimentation, parameter sharing, and dataset alignment. Their checkpoints can be fine-tuned or quantized for domain-specific tasks—an essential property for enterprise deployment under privacy constraints. Although open models may trail proprietary ones by several percentage points in raw accuracy, they offer decisive advantages in controllability, interpretability, and replicability.

Reproducibility and Evaluation Implications. From a research standpoint, open-source LLMs are indispensable for standardized evaluation and fair comparison of SDU systems. They allow public reproduction of training and inference conditions, facilitate community benchmarks, and support audits of hallucination or bias. Conversely, results derived from closed models should be clearly annotated as *non-reproducible* baselines. A balanced ecosystem—leveraging commercial APIs for performance ceilings and open models for transparent baselines—will be critical to advancing structured document understanding responsibly.

Table 10 summarizes the complementary strengths of commercial and open-source LLMs, emphasizing how accessibility and reproducibility considerations shape their suitability for document AI research and deployment.

9.4 Open-Source LLMs for Document AI

The dominance of closed-source models like GPT-4 and Claude restricts reproducibility and community progress. To democratize document intelligence, the field urgently needs high-performing, openly available multimodal LLMs—such as LayoutLLaMA and Phi-Doc. Moreover, standardized checkpoint releases for key datasets (e.g., FUNSD, DocVQA) will catalyze benchmarking, accelerate downstream adoption, and foster transparent evaluation.

Table 10: Commercial vs. open-source LLMs for document AI.

Aspect	Commercial LLMs	Open-Source LLMs
Access	API-based, closed weights	Public checkpoints and code
Reproducibility	Limited; proprietary data	High; transparent training and eval
Adaptability	Prompt tuning only	Full fine-tuning and PEFT supported
Performance	Typically higher accuracy	Competitive with task tuning
Transparency	Opaque data and methods	Open architecture and logs
Cost	Pay-per-use API pricing	One-time or local compute cost
Privacy	External data handling	On-premise or private deployment
Best Fit	Prototyping, multilingual QA	Research, domain-specific adaptation

9.5 Prompt Robustness and Adaptability

Despite their flexibility, prompt-based systems remain overly sensitive to phrasing and structure. Improving prompt robustness is not optional—it is essential. Research must target prompt rewriting, self-refinement, and consistency enforcement mechanisms. Additionally, automatic retrieval and selection of few-shot examples via semantic clustering or summarization will unlock scalable, adaptive prompt pipelines across diverse document types.

9.6 Efficient Training and Inference

Scaling LLMs for document AI should not come at the cost of efficiency. Sparse Mixture-of-Experts (MoE) models [34, 5] offer compelling efficiency gains without sacrificing quality. Techniques like LoRA [35], BitFit [36], and quantization enable domain-specific adaptation on low-resource setups. For real-world deployment, model compression through pruning and distillation [37] will be essential to achieve fast, memory-efficient inference.

9.7 Benchmark Standardization and Evaluation Gaps

The current evaluation landscape is fragmented and insufficiently diagnostic. The community must converge on unified, multilingual, multi-task benchmark suites with consistent metrics such as F1, ANLS [12], and BLEU. Equally important is the development of failure mode taxonomies—capturing hallucinations, layout misalignment, and reasoning gaps—to ensure deeper insight into model limitations and reliability.

10 Conclusion

This paper delivers a comprehensive survey and benchmark-driven comparison of large language models (LLMs) versus traditional non-LLM models for structured document understanding. We systematically covered four core tasks: key-value pair extraction, named entity recognition, document classification, and document question answering, evaluating model performance across more than ten widely used datasets.

Our analysis clearly demonstrates that instruction-tuned and layout-aware LLMs—such as DocLLM [4], LayoutLLM [28], and MoE-LLaVA [5]—are decisively outperforming layout-specific transformers like LayoutLMv2 [2] and OCR-dependent pipelines such as TILT [32] and FormNet [26]. These LLMs exhibit strong generalization capabilities, robust multitask performance, and effective multilingual adaptability, especially in zero- and few-shot learning scenarios.

Nonetheless, certain trade-offs remain. Non-LLM models continue to offer advantages in latency-sensitive or resource-constrained environments. Additionally, factors such as OCR quality, layout fidelity, and prompt formulation continue to influence real-world performance. No single architecture yet dominates across all domains and use cases.

To support informed deployment choices, Table 11 provides practical model recommendations tailored to common document processing scenarios. These recommendations are further contextualized in Section 8, which highlights the strengths and compromises associated with layout-aware encoders, instruction-tuned decoders, and multimodal LLMs.

Use Case Scenario	**Recommended Model Type**	**Rationale**
Low-latency or mobile deployment	Lightweight encoders	Efficient inference with compact models for edge devices.
Multilingual documents	Text-only LLMs	Language-agnostic, no dependency on layout or visual input.
Visually complex layouts	Vision-based LLMs	Strong spatial grounding from image inputs.
Few-shot generalization	Instruction-tuned LLMs	Prompt-based adaptation with minimal data.
Tabular or form-heavy docs	Layout-biased encoders	Inductive biases for row-column structure.
Multi-domain QA over documents	Multimodal MoE models	Visual experts improve cross-domain coverage.
OCR-challenged scans	OCR-free vision models	Direct image-to-sequence modeling avoids OCR failure.

Table 11: Model selection guide based on deployment scenarios and task requirements.*Performance and efficiency labels are based on reported benchmarks and model documentation.*

Beyond model-specific comparisons, this survey articulates broader insights into emerging design paradigms, evaluation gaps, and architectural trade-offs. We emphasize the ongoing evolution from rigid, OCR-centric pipelines toward adaptive, prompt-driven, and image-first systems—while also surfacing challenges such as prompt variability, hallucinations, and increasing inference costs.

Despite notable progress, the field continues to lack unified benchmarks and standardized evaluation protocols—particularly for generative and multimodal tasks. As open-source layout-aware LLMs continue to mature, we advocate for future efforts that emphasize reproducible pipelines, robust prompting strategies, and scalable lightweight solutions for deployment in constrained environments.

We hope this survey provides valuable guidance to both researchers and practitioners in selecting, evaluating, and advancing document understanding models that effectively balance generalization, efficiency, and structural fidelity.

References

[1] Yiheng Xu et al. Layoutlm: Pre-training of text and layout for document image understanding. *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

[2] Yiheng Xu et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[3] Yuwei Fang et al. Udop: Unifying vision, text, and layout for universal document processing. *arXiv preprint arXiv:2212.02623*, 2022.

[4] Yu Zhang et al. Docllm: Document language models for structured document understanding. *arXiv preprint arXiv:2401.00908*, 2024.

[5] Chen Xu et al. Moe-llava: Mixture of expert vision-language models for document understanding. *arXiv preprint arXiv:2403.15450*, 2024.

[6] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *International Conference on Document Analysis and Recognition Workshops (ICDAR-W)*, 2019.

[7] Seunghyun Park et al. Cord: A consolidated receipt dataset for post-ocr parsing. In *NeurIPS Workshop on Document Intelligence*, 2019.

[8] Yiheng Xu, Tengchao Lv, Lei Cui, Yijuan Huang, Furu Wei, and Ming Zhou. Xfund: A benchmark dataset for multilingual form understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[9] Zenghui Huang et al. Icdar 2019 competition on scanned receipt ocr and information extraction (sroie). In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[10] Adam W. Harley, Alexander Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[11] David D. Lewis et al. Tobacco-3482: A large-scale dataset for document classification. In *Proceedings of the 29th Annual International ACM SIGIR Conference*, 2006.

[12] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[13] Mohsin Afzal et al. Infographicvqa: Infographic visual question answering through graph-based reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Ankush Mishra, Chitta Baral Reddy, and Vivek Natarajan. Ocr-vqa: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[15] Yujie Lu et al. Instructuie: Multi-task instruction tuning for information extraction. *arXiv preprint arXiv:2306.05655*, 2023.

[16] Geewook Kim et al. Donut: Document understanding transformer without ocr. In *European Conference on Computer Vision (ECCV)*, 2022.

[17] Filip Graliński, Tomasz Stanisławek, Ireneusz Gawlik, Maciej Pietras, Przemysław Pawlak, Łukasz Borchmann, Bartosz Topolski, and Michał Wrzeszcz. Kleister: A benchmark for document understanding. *arXiv preprint arXiv:2003.02356*, 2020. URL `https://arxiv.org/abs/2003.02356`.

[18] Yiheng Li, Lei Xu, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020. URL `https://arxiv.org/abs/2006.01038`.

[19] Mina Masry, Prem Natarajan, Silvio Savarese, and Kai-Wei Chang. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. URL `https://arxiv.org/abs/2205.05677`.

[20] Brandon Smock, Nghia Phan, Tathagata Ghosal, Hongfang Yao, Rui Wang, Michael Cafarella, and Andrew Arnold. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4307–4322, 2022. URL `https://aclanthology.org/2022.acl-long.296`.

[21] Minghao Li, Xu Zhong, Leyang Cui, Shaohan Huang, Furu Zhang, Ming Zhou, and Endong Liu. Tablebank: A benchmark dataset for table detection and recognition. *arXiv preprint arXiv:1903.01949*, 2019. URL `https://arxiv.org/abs/1903.01949`.

[22] Benjamin Pfitzmann, Frederik Raue, Jonas Gudladt, Alexander Binder, and Thomas Brox. Doclaynet: A large human-annotated dataset for document-layout analysis. *arXiv preprint arXiv:2303.15452*, 2023.

[23] Geewook Kim, Moonsu Lee, Sungrae Kim, and Heeyoul Kim. Synthdog: Synthetic document generation for pretraining document understanding models. In *European Conference on Computer Vision (ECCV)*, 2022.

[24] Wenwen Yu et al. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, 2020.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[26] Chen-Yu Lee et al. Formnet: Structural encoding beyond sequential modeling in form document information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[27] Taehoon Kim, Seungyoung Lim, and Kee-Eung Kim. Synthdog: Synthetic document generation for pre-training document understanding models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1978–1982, 2021. doi: 10.1145/3404835.3463067.

[28] Jie Luo et al. Layoutllm: Layout-aware instruction tuning for document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[29] Zilong Li et al. Docvlm: A vision-language model for document understanding. *arXiv preprint arXiv:2303.17400*, 2023.

[30] Yupan Huang et al. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*, 2022.

[31] Srikar Appalaraju et al. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[32] Rafał Powalski et al. Tilt: A text-image-layout transformer for document ai. *arXiv preprint arXiv:2103.15601*, 2021.

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. URL `https://arxiv.org/abs/2201.11903`.

[34] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. URL `https://arxiv.org/abs/1701.06538`.

[35] Edward J. Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. URL `https://arxiv.org/abs/2106.09685`.

[36] Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. URL `https://arxiv.org/abs/2106.10199`.

[37] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. URL `https://arxiv.org/abs/1910.01108`.

Appendix A: Model Configuration Summary

Model	Variant	Max Tokens	Image Size	Notes
LayoutLMv2	base	512	224x224	ResNet101 visual backbone
LayoutLMv3	base	512	224x224	ViT-style patch embeddings
DocFormer	base	512	224x224	Gated attention fusion
FormNet	large	–	–	No vision input; row-column bias
Donut	base	–	1280x960	Swin Transformer encoder
InstructUIE	mT5-base	512	–	Text-only; multilingual
DocLLM	LLaMA-7B	2048	–	Layout embedding via discretization
LayoutLLM	LLaMA-7B	2048	224x224	Fused layout map and token input
UDOP	T5-large	512	384x384	OCR-text overlaid on image
DocVLM	ViT-T5-base	512	384x384	Vision-text contrastive learning
MoE-LLaVA	LLaVA + Experts	2048	336x336	Top-k visual expert routing

Table 12: Configuration summary for models surveyed, including variant used, token and image input dimensions, and key architectural notes.*Reported configurations are taken from original papers and official implementations.*