DocAgent: An Agentic Framework for Multi-Modal Long-Context Document Understanding

Li Sun[1†], Liu He[2], Shuyue Jia[1], Yangfan He[3], Chenyu You[4†],
[1]Boston University, [2]Purdue University, [3]University of Minnesota, [4]Stony Brook University
lisun@bu.edu, chenyu.you@stonybrook.edu

Abstract

Recent advances in large language models (LLMs) have demonstrated significant promise in document understanding and question-answering. Despite the progress, existing approaches can only process short documents due to limited context length or fail to fully leverage multi-modal information. In this work, we introduce DocAgent, a multi-agent framework for long-context document understanding that imitates the human reading practice. Specifically, we first extract a structured, tree-formatted outline from documents to help agents identify relevant sections efficiently. Further, we develop an interactive reading interface that enables agents to query and retrieve various types of content dynamically. To ensure answer reliability, we introduce a reviewer agent that cross-checks responses using complementary sources and maintains a task-agnostic memory bank to facilitate knowledge sharing across tasks. We evaluate our method on two long-context document understanding benchmarks, where it bridges the gap to human-level performance by surpassing competitive baselines, while maintaining a short context length. Our code is available at https://github.com/lisun-ai/DocAgent.

1 Introduction

Effectively understanding and answering questions about lengthy multi-modal documents is crucial for various real-world applications, including business intelligence, academic research, and legal analysis (Mathew et al., 2021). However, automatic document understanding remains a significant challenge due to three key factors. First, unlike purely unstructured text, documents exhibit a hierarchical structure (Tkaczyk et al., 2015), requiring models to capture contextual relationships across sections, subsections, and embedded elements. Second, documents contain diverse modalities, such as text, tables and images (Ma et al., 2024). Processing these multi-modal components adds complexity, as it demands specialized techniques for extracting, interpreting, and integrating information from different formats. Finally, many documents, such as financial reports and legal agreements, are extensive, often spanning hundreds of pages (Chia et al., 2024). This sheer length poses computational challenges for natural language processing (NLP) models, which struggle with long-range dependencies and memory constraints (Finkel et al., 2005).

In recent years, researchers have explored ways to develop automatic document understanding and question-answering models, which can be broadly categorized into two main approaches. The first relies on Optical Character Recognition (OCR) to extract text from documents, often integrating visual elements such as images or layout structures for answer prediction (Xu et al., 2020, 2021; Huang et al., 2022; Peng et al., 2022). The second approach adopts an end-to-end learning framework that interprets documents holistically, bypassing explicit text extraction via OCR (Kim et al., 2022; Lee et al., 2023).

Despite significant progress, most prior research has mainly focused on short documents spanning only a few pages due to memory and computational constraints. However, real-world documents – such as legal contracts, research papers, and technical reports – are often substantially longer, sometimes extending to hundreds of pages. Moreover, prior work has not fully leveraged structural and multi-modal information (Saad-Falcon et al., 2024). Addressing these limitations is essential for advancing long-context and multi-modal document understanding.

To bridge the gap, we propose DocAgent, an agentic framework for long multi-modal document understanding. LLM agents are AI systems that leverage LLMs to perform complex tasks, often utilizing external tools to plan and execute effi-

†Corresponding authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 17701–17716
November 4-9, 2025

ciently (Yao et al., 2023; Shinn et al., 2024). Our approach is inspired by how humans answer questions based on long documents. When engaging with extensive content, humans do not read the entire document word for word; instead, we follow a structured process–scanning the outline, identifying relevant sections, cross-referencing information across sources, and progressively synthesizing insights. Our framework emulates this strategy. Specifically, we extract a concise hierarchical outline from the document to guide the LLM agent in identifying question-relevant sections. This outline provides structural cues, including section headings, table titles, and image captions. Additionally, we develop an interactive reading interface that allows the agent to query and retrieve multi-modal content. By leveraging the outline and selectively retrieved content instead of the full document, we significantly reduce the required context length. Prior research efforts (Shi et al., 2023) have demonstrated that irrelevant context can distract LLMs and degrade performance. Furthermore, we introduce a reviewer agent that cross-verifies answers using complementary sources, enhancing reliability and minimizing errors. When a correction is necessary, the reviewer updates a task-agnostic memory bank, enabling knowledge sharing across tasks. To validate our model, we conduct a comprehensive comparison against various baselines, including OCR-based approaches (Smith, 2007), Multimodal LLMs (Hurst et al., 2024), and Retrieval-Augmented LLMs (Saad-Falcon et al., 2024; Cho et al., 2024).

Our contributions can be summarized as follows:

- We propose a novel multi-modal agentic framework that leverages a tree-structured outline and retrieval tools to identify and extract relevant document content efficiently.
- We introduce a reviewer agent that cross-verifies and enhances answers by incorporating information from complementary sources.
- We develop a task-agnostic memory bank that enables the agent to learn from prior experience, improving performance across tasks.
- We conduct experiments on two long-context multi-modal document understanding benchmarks and perform ablation studies to validate the effectiveness of our proposed method.

2 Related Work

2.1 Document Question Answering

Efforts to advance document question answering (DQA) have generally followed two primary paths. The first utilizes OCR to extract text from documents, often integrating visual features to enhance answer prediction. For example, LayoutLM series (Xu et al., 2020, 2021; Huang et al., 2022) integrate both OCR-extracted text and visual embeddings for document understanding tasks. The second path bypasses the OCR stage and employs an end-to-end learning paradigm (Kim et al., 2022; Lee et al., 2023). Recent works (Hu et al., 2024; Rasool et al., 2024) leverage the capability of LLM to empower DQA. For example, LayoutLLM (Luo et al., 2024) proposes layout chain-of-thought to guide the LLM to focus on relevant regions. Our work differentiates from these previous studies in two aspects: On methodology, we propose an agentic framework that imitates human reading practice; On evaluation, we conduct experiments on long documents spanning over hundreds of pages, where most previous works only benchmark on short documents with few pages.

2.2 LLM Agents

LLM agents interact with their environment and take actions to accomplish specific goals. ReAct (Yao et al., 2023) introduces interleaved generation of reasoning trace and actions to help the agent overcome hallucination and reduce error propagation in long chain-of-thought. Reflection (Shinn et al., 2024) proposes to use linguistic feedback to reinforce language agents. Moreover, Wu et al. (2023) consolidate multi-agent workflows with conversations to complete complex tasks. Furthermore, Zhou et al. (2024) saves previous trajectories in memory to provide additional context to the agent. Our method differs from these previous works by introducing a novel reflection module that compares the trajectories between actor and reviewer to derive insights into memory.

2.3 Enhancing LLM Efficiency for Long Contexts

Since the complexity of the transformer grows quadratically with sequence length, researchers have been actively exploring techniques to optimize LLM for extended context size. For example, MARG (D'Arcy et al., 2024) utilizes a multi-agent framework that divides long content into chunks and distributes them to worker agents for processing. However, this fragmentation may lead to challenges in maintaining long-distance dependency, potentially affecting the overall comprehension and consistency. In addition, retrieval augmented generation is introduced to overcome the context length constraint (Xu et al., 2023). For example, PDFTriage (Saad-Falcon et al., 2024) uses tools to fetch information from the document. However, their method is constrained to the language modality and lacks the capability to handle images and charts. Wang et al. (2023) propose to recursively summarize textural contexts to overcome context constraints. Despite the progress, enhancing LLM efficiency for multi-modal long context is still an under-explored area.

3 Method

Our proposed DocAgent consists of four key components: (1) Outline construction module – Generates a structured and concise layout of lengthy documents, serving as a navigational guide for the agent; (2) Actor agent – Utilizes the outline and tool interface to retrieve relevant content and generate an initial answer; (3) Reviewer agent – Verifies and refines the initial answer to improve accuracy and reliability; and (4) Memory module – Stores task-agnostic knowledge through reflection to facilitate knowledge transfer across tasks. The overall architecture of our model is illustrated in Fig. 1. In the following sections, we provide a detailed discussion of each component.

3.1 Outline Construction from Document

To help the LLM agent efficiently navigate the document and locate evidence for question answering, we construct an outline from the document. Specifically, we first parse and extract content using Adobe PDF Extract (Adobe, 2024). We also note the availability of open-source alternatives for document content extraction, such as DocXChain (Yao, 2023) and PyMuPDF (Artifex, 2024). Next, we construct a hierarchical XML tree that represents the document's structure. This involves systematically organizing the document into a nested tree format to provide a clear and structured representation. Specifically, each section serves as a parent node, with its associated headers, subsections, paragraphs, images, and tables arranged as child nodes, forming a well-defined hierarchy. To facilitate precise navigation, each section includes attributes for the starting and ending page numbers, allowing the agent to locate relevant content efficiently. To optimize context length, paragraph content remains hidden, while only the first sentence is provided as an attribute of the paragraph element, offering the agent a contextual hint. Similarly, the visual content of figures is omitted, with only the caption included as an attribute of the image element. Additionally, each section, image, and table is assigned a unique identifier, enabling the agent to retrieve full content when necessary. An example of a constructed document outline is shown in Fig. 1.

3.2 Actor Agent

The actor M_A operates using LLM as its reasoning engine. At each timestep n, the agent receives state observations s_n, a question Q, a set of predefined instructions I_A, and learned guidelines in memory mem in its prompt. Based on these inputs, the actor samples an action a_n from its policy ρ_A, which can either involve tool calls to retrieve additional evidence from the document or directly provide an answer, thereby terminating the loop. Formally,

$$a_n \sim \rho_A(a|s_n, Q, I_A, mem). \tag{1}$$

To facilitate efficient query and retrieval of multi-modal content, we design a document interface with five tools that enable the agent to interact with the document effectively. The descriptions of the tools are presented in Table. 1. When these tools are executed, their outputs are incorporated into the agent's observation state s_n, continuously refining its understanding of the document. $T_A = \{s_0, a_0, \cdots, s_n, a_n\}$ is the agent trajectory.

3.3 Reviewer Agent

Given that documents often present information across multiple modalities with overlapping content (e.g., the same information appearing in both images and associated text) (Hassan et al., 2013), we introduce a reviewer M_R to validate the actor's initial responses. The reviewer cross-references additional evidence from different sources or modalities to ensure the accuracy and completeness of the provided answers. Concretely, at each timestep n, the reviewer processes three inputs: the question Q, the reviewer's instructions I_R, and the actor's trajectory T_A (which includes the initially proposed answer). Based on these inputs, the reviewer samples an action a_n – either invoking additional tools

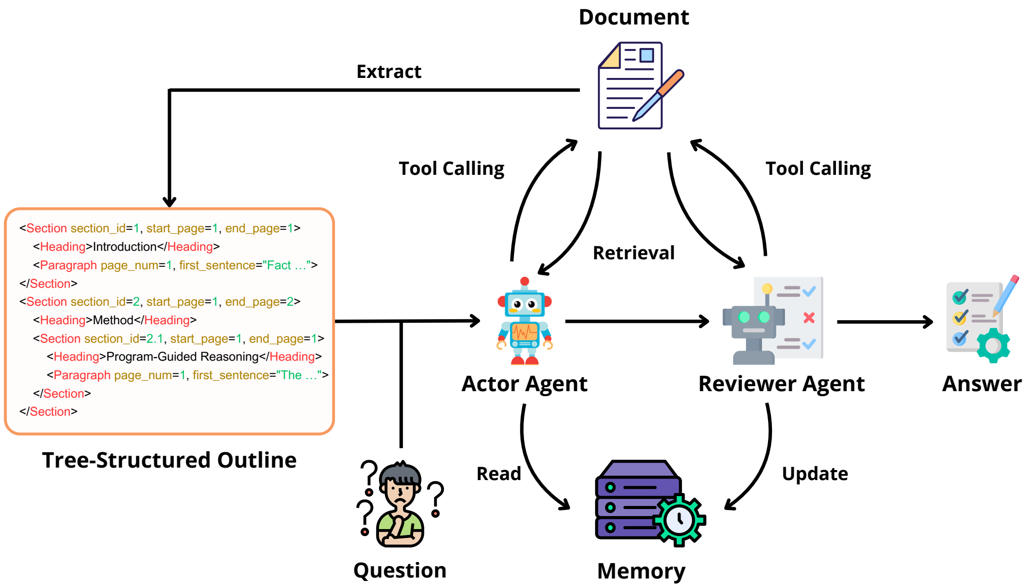


Figure 1: **The overview of our approach.** We first construct the document outline as a compact XML tree structure. An actor agent processes the outline and question as input, leveraging retrieval tools to extract relevant content for answering the question. Next, an evaluator agent reviews the initial response. If a correction is necessary, the actor agent's memory is updated through reflection, facilitating knowledge transfer across tasks.

Table 1: Overview of available tools.

Tool name	Parameter	Description
search	keyword	Find and extract all contents where the search term appears
get_section_content	section_id	Get the full-text content of a section
get_image	image_id	Get the visual content of an image
get_page_images	start_page_num, end_page_num	Extract scanned page images from a specified range of pages
get_table_image	table_id	Obtain the graphical depiction of a table or chart

for further verification or terminating the process by providing a final answer. Formally,

$$a_n \sim \rho_R(a|s_n, Q, I_R, T_A). \quad (2)$$

3.4 Memory

During problem-solving, humans maintain detailed access to recent information while simultaneously drawing upon condensed experiences from long-term memory (Baddeley and Hitch, 1974). To enable the agent to learn from past experiences, we incorporate reflection with memory. Specifically, when the actor's initial answer differs from the reviewer's final answer, we input the actor's and reviewer's trajectories T_A and T_R, the question Q, and the instruction set I_S into a reflection module M_S which is based on an LLM. This module updates the memory by generating verbal guidelines that help the actor agent improve its performance in future tasks. Formally,

$$mem' \sim \rho_S(m|Q, I_S, T_A, T_R, mem). \quad (3)$$

Our motivation stems from the hypothesis that since the reviewer observes additional content about the document, it will perform better than the actor alone. Therefore, if a correction is made by the reviewer, we can extract useful guidance by examining the trajectories to help the actor perform better. Unlike LATS (Zhou et al., 2024), which saves lengthy trajectories in memory, and Reflection (Shinn et al., 2024), which employs a queue-based memory bank where older reflections are discarded once capacity is reached, our design maintains a cumulative memory as a string variable that is continuously updated over time. This approach allows the agent to preserve insights from long history. The complete `DocAgent` algorithm is shown in Algorithm 1.

3.5 Implementation Details

The temperature for LLM inference is set to 0. We utilize GPT-4o (Hurst et al., 2024), Claude 3.5 Sonnet (Anthropic, 2024), and Gemini 2.0 Flash (Anil

Algorithm 1 DocAgent Framework

Initialize Actor M_A, Reviewer M_R, Reflection Module M_S, Memory mem;
Set $s_n = []$
for $t = 0$ to max rounds **do**
  Actor M_A samples action a_t
  if a_t is tool calling **then**
    Retrieve content with tool to update s_n
  else
    Obtain actor's response R_A
    break
  end if
end for
Set $s_n = []$
for $t = 0$ to max rounds **do**
  Reviewer M_R samples action a_t
  if a_t is tool calling **then**
    Retrieve content with tool to update s_n
  else
    Obtain reviewer's response R_R
    break
  end if
end for
if R_A != R_R **then**
  Update mem with Reflection M_S
end if
return R_R

et al., 2025) as the base models through API in our experiments, though it remains compatible with most existing MLLMs. To prevent excessive iterations, we impose a maximum limit of 10 tool-calling rounds. The memory is initialized as an empty string. The actor, reviewer, and reflection module are coordinated by using a shared message thread. To minimize context length, we do not employ chain-of-thought prompting or in-context examples. The prompts used in our framework are provided in the Appendix. A.1. Table. 14 presents an example of `DocAgent`'s workflow.

4 Experiments

We evaluate the performance of our proposed `DocAgent` on two multi-modal long-context document question answering benchmarks. We compare the performance of our method with various baseline methods. In addition, we evaluate the retrieval performance of our approach and analyze patterns in tool utilization. Finally, we conduct ablation studies to validate the effectiveness of our proposed modules.

4.1 Datasets

We conduct experiments on two multi-modal long-context document understanding and question-answering benchmarks, including MMLongBench-Doc (Ma et al., 2024) and DocBench (Zou et al., 2025). The MMLongBench-Doc dataset comprises 135 extensive PDF documents, averaging 47.5 pages each and spanning 7 distinct document categories. The dataset comprises 1,082 expert-annotated questions, from which answers can be derived from multiple sources, including textual content, images, and charts. In addition, the evidence for answers can span across pages. The DocBench dataset contains 229 long documents along with 1,102 questions. The documents come from five different domains, including Academia, Finance, Government, Laws and News.

4.2 Evaluation of Question Answering

We evaluate our `DocAgent` on the two document question-answering benchmarks introduced above. For MMLongBench-Doc, we compare our method against three competitive baseline methods: (1) LLM with OCR-extracted textural content. Following the practice of Ma et al. (2024), we use Tesseract OCR (Smith, 2007) to extract text from each page, then concatenate and feed them to LLM. However, it lacks the ability to process information from charts and images. In addition, texts exceeding the context length limit are truncated; (2) Multi-modal LLM with page images, we scan each page of the document with 144 DPI. The page images are used as input to MLLM. For the Claude model, due to the limitation on the number of uploaded images, we follow the practice in Ma et al. (2024) to concatenate and reduce the image number to 20; (3) Chain-of-Agent (Zhang et al., 2024), an agent-based system where worker agents sequentially process text segments, and a manager agent integrates their outputs into a final response; (4) M3DocRAG (Cho et al., 2024), a multi-modal RAG framework that retrieves relevant pages for question answering; (5) PDFTriage (Saad-Falcon et al., 2024), which adopts a retrieval-augmented generation approach, also relies on content drawn from Adobe PDF Extract. We use the official implementation, and upgrade its base model from GPT-3.5 used in the original paper to GPT-4o for fair comparison. In terms of evaluation, we follow the official approach that first utilizes an LLM-based

Table 2: Evaluation of various models on MMLongBench-Doc. We report the generalized accuracy of seven types of document domains. The overall accuracy and F1 scores are reported in the rightmost columns. The best performance is highlighted in bold. - means not reported.

Method	**Document Domain** Academic	Brochure	Financial	Guidebook	Industry	Report	Tutorial	**ACC**	**F1**
Methods Based on Gemini 2.0 Flash (Anil et al., 2025)									
Gemini w/ OCR (Ma et al., 2024)	36.9	30.6	45.4	39.9	44.1	41.9	37.4	39.6	37.2
Gemini w/ Page Images (Ma et al., 2024)	27.9	31.5	31.4	43.3	44.6	44.3	43.2	38.4	36.1
Gemini w/ CoA (Zhang et al., 2024)	28.0	28.6	44.2	34.6	43.2	45.2	33.2	37.2	31.9
Gemini w/ RAG (Cho et al., 2024)	19.3	24.4	7.7	22.9	40.8	24.9	18.8	22.1	11.1
DocAgent (Ours)	**51.0**	**41.2**	**54.5**	**50.0**	**50.7**	**49.8**	**45.8**	**49.3**	**47.0**
Methods Based on GPT-4o (Hurst et al., 2024)									
GPT-4o w/ OCR	-	-	-	-	-	-	-	30.1	30.5
GPT-4o w/ Page Images	35.9	**42.7**	46.5	44.7	**53.3**	43.9	53.0	42.8	44.9
PDFTriage (Saad-Falcon et al., 2024)	40.6	32.8	45.9	40.0	42.8	41.7	31.2	39.6	35.6
DocAgent (Ours)	**44.7**	**42.7**	**63.0**	**52.1**	53.2	**50.7**	**59.9**	**51.8**	**49.1**
Methods Based on Claude 3.5 Sonnet (Anthropic, 2024)									
Claude 3.5 Sonnet w/ OCR	46.4	36.6	**62.8**	43.3	46.4	43.4	37.3	44.9	42.4
Claude 3.5 Sonnet w/ Page Images	26.5	30.9	30.3	38.2	57.0	44.4	49.4	39.0	36.7
DocAgent (Ours)	**60.2**	**49.1**	59.4	**53.4**	**57.8**	**61.3**	**53.4**	**57.3**	**54.1**
Human Baseline (Ma et al., 2024)									
Human Baseline	-	-	-	-	-	-	-	65.8	66.0

Table 3: Results on DocBench across various types and domains. We report the generalized accuracy of five types of document domains, including Academia (Aca.), Finance (Fin.), Government (Gov), Law, and News. The best performance is highlighted in bold.

Methods	**Document Domain**					**Evidence Source**				**Overall Acc**
	Aca.	Fin.	Gov.	Law	News	Text-only	Multimodal	Metadata	Unanswerable	
GPT-4o w/ File Attachment	56.4	56.3	**73.0**	65.5	75.0	85.0	62.7	50.4	17.7	63.1
DocAgent (Ours)	**77.9**	**80.9**	72.3	**80.1**	**87.8**	**90.5**	**85.7**	**61.2**	**68.5**	**79.9**
Human Baseline (Zou et al., 2025)	83.0	82.2	77.8	75.0	86.4	81.4	83.3	77.5	82.2	81.2

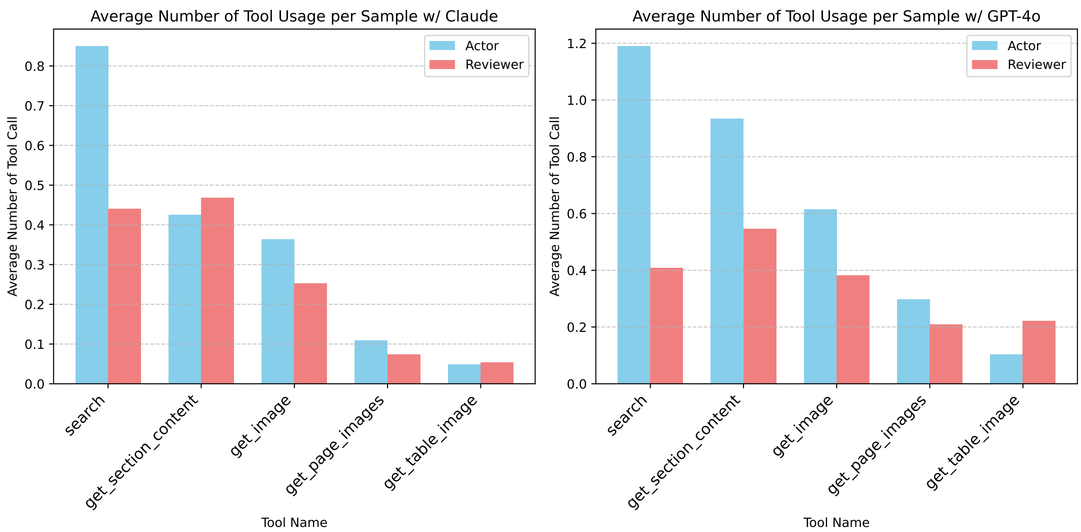


Figure 2: Average number of tool usage per sample on MMLongBench-Doc, categorized by agent role and base model. We observed that the actor primarily relies on the search tool, facilitating efficient retrieval of information relevant to the question. In contrast, the reviewer most frequently utilizes the get_section_content tool, enabling a thorough examination of the full section to assess the answer comprehensively.

answer extractor (GPT-4o in this case) to transform lengthy responses into concise answers. Then, a rule-based score calculator is applied to evaluate the shortened answers. The results are shown in Table. 2. For GPT-4o with OCR or page images, we use results from (Ma et al., 2024). In terms of overall accuracy, our proposed DocAgent outperforms the best-performing baseline by 9% with GPT-4o, and 12.4% with Claude 3.5 Sonnet, closing the gap with human performance. When categorized by different domains, our DocAgent outperforms baseline methods most of the time. The most significant improvement occurs in financial reports (16.5%) when based on GPT-4o, and in research reports (16.9%) when based on Claude 3.5 Sonnet. Both categories feature lengthy documents with multi-modal content like tables and charts.

For DocBench, we compare our method with GPT-4o with File Attachment, which is a proprietary system from OpenAI. The baseline results are reported in Zou et al. (2025). We follow the official evaluation process, which includes judging criteria within the instruction prompt, and then classify the results using GPT-4. The results are presented in Table. 3. We found that our DocAgent outperforms the baseline method by 15.2% in terms of overall accuracy, and is only 1.3% away from human performance. We also report results categorized by document domains and evidence sources. Our DocAgent outperforms baseline most of the time. The document category we observe the most improvement is financial report (24.6%), which contains extensive multi-modal content with tables and charts.

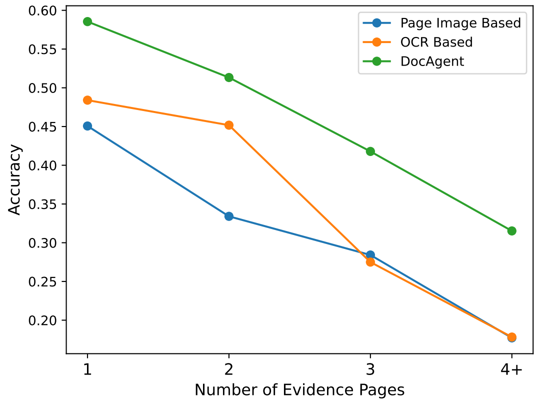


Figure 3: Accuracy on MMLongBench-Doc categorized by the number of evidence pages.

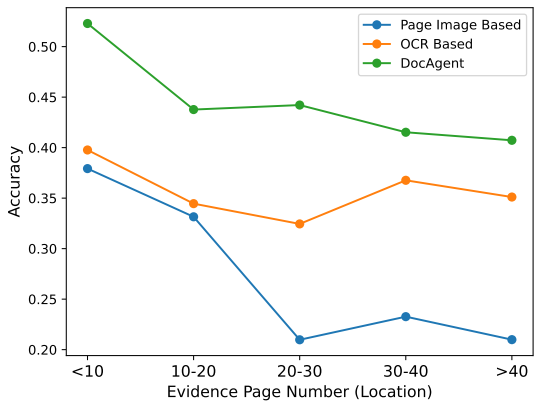


Figure 4: Accuracy on MMLongBench-Doc categorized by page number (location) of evidence.

4.3 Fine-Grained Performance Analysis

Number of Evidence Page. We study the performance of models categorized by the number of pages that contain the evidence that can be used to answer the question. The results are shown in Fig. 3. We use Claude 3.5 Sonnet as the base model for this experiment.While increasing the number of evidence pages led to decreased performance across all models, our DocAgent demonstrates superior robustness compared to the baselines. Not only does it maintain the highest overall performance, but it also exhibits the smallest decline as the evidence volume grows.

Evidence Position. In this section, we study the relationship between the location of evidence and model performance. The results are shown in Fig. 4. Claude 3.5 Sonnet is used as the base model for this experiment. We find the performance of most methods (except for OCR-based) drops when the relevant evidence is located towards the end of the document. Our DocAgent consistently outperforms baseline methods regardless of evidence locations.

4.4 Evaluation of Retrieval Performance

Our DocAgent uses tool calling to gather information needed to answer the question. In this section, we evaluate the performance of information retrieval on the page level. Specifically, we collect the page numbers of retrieved content, including section paragraphs, images and tables. Next, we evaluate the overlap between the page numbers of retrieved elements and the page numbers of evidence in the annotation. We present the results on MMLongBench-Doc in Table. 4. Our results demonstrate that Claude 3.5 Sonnet performs better than GPT-4o when used as the base model, achieving 69.1% on recall and 35.4% on F1 score. In addition, by selectively accessing only 24.2% of document pages on average when powered by Claude, our DocAgent significantly minimizes context window consumption, making it capable of processing and comprehending lengthy documents effectively.

4.5 Analysis of Tool Usage

In this section, we analyze the statistics of tool usage of our DocAgent to gain insight into the behavior of agents. The study is conducted at two levels: across the entire dataset and at individual instances. On the dataset level, we calculate the average number of calls made for each tool. The results are shown in Fig. 2, we found that the search tool and the get_section_content tool are consistently the most frequently used. In addition, the actor primarily relies on the search tool, enabling efficient retrieval of information relevant to the question. In contrast, the reviewer most frequently utilizes the get_section_content tool, allowing for an in-depth examination of the full section to assess the answer thoroughly.

On instance level, in order to determine whether the actor and reviewer utilize complementary types of tools, we analyze the extent of overlap in their tool usage. Specifically, we calculate the Intersection over Union (IoU) of used tool categories between the actor and reviewer for each instance. The averaged results are presented in Table. 5. We found that when driven by Claude 3.5 Sonnet model, the IoU is only 15.3%, which indicates that the reviewer chooses to use different types of tools with the actor to cross-check the evidence.

4.6 Ablation Studies

To assess the impact of the proposed components, we conduct ablation studies by removing the reviewer agent and memory from DocAgent. In addition, we measure the performance divided by evidence sources. The results are shown in Table. 6. Our findings indicate that both components contribute to overall performance. We found that the reviewer agent brings the most performance gain on multi-modal evidence, including layout (6.2%) and chart (3.5%), which highlights its effectiveness in cross-checking content from diverse sources.

4.7 Context Length Analysis

In this section, we study the context window utilization by different methods. Specifically, we count the total number of tokens in the sequence after the model produces an answer. We report the results in Table. 7. We found that DocAgent's context window usage is comparable to the baseline methods. However, when we disable both the reviewer and memory module, DocAgent uses significantly less context, resulting in the most efficient context length consumption among all methods tested, while still achieving higher accuracy than baselines.

Table 7: Results of context length usage (number of tokens in the sequence) on MMLongBench-Doc. The results with the shortest context length are highlighted in bold. - means not reported.

	w/ Claude	w/ GPT-4o
OCR Based	21809	-
Page Images Based	22417	-
PDFTriage	-	20677
DocAgent	22820	20727
DocAgent w/o Memory	22530	19015
DocAgent w/o Reviewer & Memory	**20680**	**17093**

Besides the average value, we also measure the P99 and maximum context length usage. The results with Gemini as the base model are reported in Table 8. While the average context length usage is similar, DocAgent demonstrates clear advantages under high-load scenarios, reducing the P99 context length by 37% and maximum context length by 26%. These results highlight the efficiency and scalability of our approach, especially for processing large or complex documents. In addition, DocAgent is capable of incorporating visual context, which the text-only baseline fails to capture. However, we note that our DocAgent has a higher API cost than text-only baselines, primarily due to the cost introduced by prompt caching.

Table 8: Results of context length usage (number of tokens in the sequence) on MMLongBench-Doc with Gemini as the base model.

Method	Avg	P99	Max
Text-only (OCR)	**21511**	136491	265288
DocAgent (Ours)	21748	**85758**	**195588**

4.8 Latency Analysis

In this section, we conducted a detailed quantitative analysis of both latency and computational cost across various approaches. We use Gemini as the base model in this comparison. As summarized in the table 9, while our DocAgent exhibits higher latency than a single-pass baseline, it substantially outperforms other multi-agent and RAG-based baselines in terms of speed.

Importantly, DocAgent was designed with modularity and deployment flexibility in mind. When the

Table 4: Evaluation of retrieval performance of DocAgent.

Model	Precision	Recall	F1	% Pages Retrieved
DocAgent w/ GPT-4o	25.6%	67.8%	32.3%	55.9%
DocAgent w/ Claude	28.7%	69.1%	35.4%	24.2%

Table 5: Overlap in used tool type between actor and reviewer.

	w/ Claude	w/ GPT-4o
Tool Usage IoU	15.3%	27.9%

Table 6: Results of ablation studies on MMLongBench-Doc. We report the generalized accuracy of five types of evidence sources, including pure text (TXT), layout (LAY), chart (CHA), table (TAB), and image (IMG). We use Claude 3.5 Sonnet as the backbone for this experiment. The best performance is highlighted in bold.

Model	**Evidence Source**					**ACC**
	TXT	LAY	CHA	TAB	IMG	
Ours	**51.9**	**52.3**	**53.1**	**61.3**	**43.8**	**57.3**
Ours w/o Memory	50.5	49.1	52.1	60.7	41.7	56.7
Ours w/o Reviewer & Memory	50.5	42.9	48.6	58.4	38.3	55.0

reviewer and memory agent modules are disabled, the system achieves a 30% reduction in latency, with only a modest 10% relative drop in performance, a tunable trade-off that supports practical usage in latency-sensitive scenarios.

Table 9: Comparison of different methods and their latency

Method	Latency (s)
OCR Based	1.5
Page Images Based	2.2
CoA (Zhang et al., 2024)	43.7
M3DocRAG (Cho et al., 2024)	44.8
Ours	10.6
Ours w/o Reviewer & Memory	6.5

5 Conclusion

In this work, we present DocAgent, a novel multi-agent framework for multi-modal long-context document understanding. When evaluated on challenging long-context document understanding benchmarks, our method not only surpassed competitive baselines in performance but also benefited from a short context length.

Limitations

In this section, we discuss three main limitations of this work. First, our DocAgent may involve several rounds of tool calling, which can lead to increased system latency due to multiple interactions with external tools. In addition, we use existing general-purpose LLMs as the base model in our framework. They may lack certain domain-specific knowledge required for comprehensive document understanding. Finally, we only use English datasets in our experiments, which could lead to bias in the results and limit the generalizability of our method due to cultural and contextual differences in how information is structured in documents across languages. We plan to address these limitations in future work.

Acknowledgments

We gratefully acknowledge Stony Brook Research Computing and Cyberinfrastructure and the Institute for Advanced Computational Science at Stony Brook University for providing the computational resources and support that contributed to the research presented in this publication.

References

Adobe. 2024. Adobe PDF Extract API. Accessed: 2024-11-01.

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, and Gemini Team. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

AI Anthropic. 2024. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3:6.

Artifex. 2024. Pymupdf: A python binding for mupdf.

Alan D. Baddeley and Graham Hitch. 1974. Working memory. volume 8 of *Psychology of Learning and Motivation*, pages 47–89. Academic Press.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*.

Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics*, pages 363–370.

Ehtesham Hassan, Santanu Chaudhury, and Madan Gopal. 2013. Multi-modal information integration for document retrieval. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1200–1204. IEEE.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. 2024. mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. pages 3096–3120.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

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

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

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

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Yuhui Cao, Weichong Yin, Yongfeng Chen, Yin Zhang, et al. 2022. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3744–3756.

Zafaryab Rasool, Stefanus Kurniawan, Sherwin Balugo, Scott Barnett, Rajesh Vasa, Courtney Chesser, Benjamin M Hampstead, Sylvie Belleville, Kon Mouzakis, and Alex Bahar-Fuchs. 2024. Evaluating llms on document-based qa: Exact answer selection and numerical extraction using cogtale dataset. *Natural Language Processing Journal*, page 100083.

Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, Seunghyun Yoon, Ryan A. Rossi, and Franck Dernoncourt. 2024. PDFTriage: Question answering over long, structured documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 153–169, Miami, Florida, US. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Li Sun, Chaitanya Ahuja, Peng Chen, Matt D'Zmura, Kayhan Batmanghelich, and Philip Bontrager. 2025. Multi-modal large language models are effective vision learners. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8617–8626. IEEE.

Li Sun, Florian Luisier, Kayhan Batmanghelich, Dinei Florencio, and Cha Zhang. 2023. From characters to words: Hierarchical pre-trained language model for open-vocabulary language understanding. *arXiv preprint arXiv:2305.14571*.

Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18:317–335.

Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. 2023. Recursively summarizing enables long-term dialogue memory in large language models. *arXiv preprint arXiv:2308.15022*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

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

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

Cong Yao. 2023. Docxchain: A powerful open-source toolchain for document parsing and beyond. *arXiv preprint arXiv:2310.12430*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language agent tree search unifies reasoning, acting, and planning in language models. In *Forty-first International Conference on Machine Learning*.

Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. 2025. Docbench: A benchmark for evaluating llm-based document reading systems. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 359–373.

A Appendix

A.1 Prompts Used By DocAgent

In this section, we present the prompts used for the actor agent, the reviewer agent, and the reflection module in Table. 10.

A.2 Ablation Study for the Update Strategy of the Memory Module

In this section, we benchmark the performance of different update strategies of the memory module. Specifically, in the main paper, we adopt the online update strategy, which updates the memory module once the model has processed and answered one question. In this section, we also benchmark the performance of the offline update strategy, which first processes all samples in the dataset, updates the memory module, and then conducts a second pass over the dataset with the updated memory module. We use Gemini 2.0 Flash as the base model, and report the results in Table 11.

We found that the offline update strategy can bring 1.2% additional performance gain. But it also brings 2 times token consumption, as we need to iterate through the dataset twice.

A.3 Evaluation for Robustness to Diverse Structural Noise

Robustness to perturbations is crucial for practical deployment (Sun et al., 2023, 2025). In order to evaluate the robustness of our framework under more realistic and diverse forms of structural noise in the document tree. In response, we conducted new experiments simulating five common types of structural perturbations observed in real-world PDF parsing, including (1) Swaps siblings: Randomly exchanging the order of sibling nodes; (2) Element Relocation: Moving nodes to incorrect parent elements; (3) Element Duplication: Duplicating existing nodes to simulate redundancy; (4) Random Insertion: Adding synthetic elements with random text or tags; (5) Element Removal: Randomly deleting child elements. To conduct this study, we randomly select a subset of nodes from the extracted abstract tree and randomly ap-

Actor:
I've uploaded a document, and below is the outline in XML format: {document_outline} Can you answer the following question based on the content of the document? <question> {question} </question> Follow these steps to answer the question: 1. As a first step, it might be a good idea to explore the document with the provided tools to familiarize yourself with its structure. 2. Locate the source in the document that can be used to answer the question. Then retrieve the full content of the source in the document with tools to examine it in detail. 3. Find the quote from the document that are most relevant to answering the question, and put it within the <quote></quote> tags. If there are no relevant quotes, write "No relevant quotes" instead. 4. When you gather enough information, return the final concise answer within the <final_result></final_result> tags, leave the explanation outside of the <final_result> tags. Important guidelines: - Be aware that the document content is obtained using OCR, so there may be scanning errors or typos. - Before each step, wrap your thought process in <analysis></analysis> tags. This will help ensure a thorough and accurate analysis of the document and question. - Please make your final answer as concise as possible. Please provide responses that only use the information you have been given in the document. If the information is unavailable, irrelevant to the question, or if none of the provided options satisfy the specified condition, you should respond with "Not answerable." {memory}
Reviewer:
Now, please validate the answer using the tools to retrieve the source of information that can be used to answer the question. Only use necessary tools. Return the final concise answer within the <final_result></final_result> tags, leave the explanation outside of the <final_result> tags.
Reflection module:
Please update the reflection listed in the <guideline> tags below on how the agent can perform better next time. Provide the updated guidance within the <updated_guideline></updated_guideline> tags. Be concise and clear, ensuring the updated guideline differs from the original by no more than one sentence. Ensure the content within the <updated_guideline> tags does not exceed 200 words. <guideline> {memory} </guideline>

Table 10: Prompts used in DocAgent. Elements enclosed in curly braces { } represent variables.

Table 11: Evaluation of various models on MMLongBench-Doc. We report the generalized accuracy of seven types of document domains. The overall accuracy and F1 scores are reported in the rightmost columns.

Method	**Document Domain** Academic	Brochure	Financial	Guidebook	Industry	Report	Tutorial	**ACC**	**F1**
Online update	51.0	41.2	54.5	50.0	50.7	49.8	45.8	49.3	47.0
Offline update	48.0	44.1	56.6	50.8	54.1	51.7	48.4	50.5	47.1

ply one of the perturbation types above. The evaluation is performed on 250 question-answering instances from the MMLongBench-Doc dataset using the Gemini model. Results are summarized in Table 12.

Table 12: Evaluation results for robustness to diverse structural noise

Perturbation rate	Accuracy (%)
0% (Baseline)	53.2
10%	52.8
20%	52.3

Encouragingly, DocAgent demonstrates resilience under moderate structural perturbations, with only a marginal performance decline. This indicates that the proposed framework is robust to a range of parser-induced errors, beyond attribute deletion alone. We hypothesize that our agent can still utilize tools such as search to obtain relevant content effectively, even when the outline has been disrupted.

A.4 Effect of Task Order

To assess the impact of task order, we conduct an additional experiment comparing two conditions: (1) the standard order from the official dataset, where tasks sharing the same document are grouped together; and (2) a shuffled order, where task sequences are randomized.

Evaluation on MMLongBench-Doc using Gemini yielded the results in Table 13. Interestingly, we observe a slight improvement when the task order is shuffled. We hypothesize that shuffling may help reduce task-level interference during processing, leading to better overall performance.

Table 13: Evaluation results for effect of task order

Method	Accuracy (%)	F1
Standard task order	49.3	47.0
Shuffled task order	50.6	48.3

A.5 Full Model Name Used

In this section, we include the full names of the APIs used in our experiments. For Gemini 2.0 Flash model, we use `gemini-2.0-flash`. For Claude 3.5 Sonnet, we use `claude-3-5-sonnet-20241022`. For GPT-4o, we use `gpt-4o-2024-08-06` for experiments on MMLongBench-Doc, and `gpt-4o-2024-05-13` for experiments on DocBench for fair comparison with results reported in Zou et al. (2025).

A.6 Case Study

In this section, we examine a specific example that demonstrates how `DocAgent` approaches and resolves a question through its interaction with the document. The example is shown in Table. 14, 15 and 16. In this example, the actor agent initially provides an incorrect answer using only the textual source. The reviewer cross-checks the chart image and provides the correct answer. The memory module is updated accordingly to accumulate experience.

User:
I've uploaded a document, and below is the outline in XML format:
<Outline>
<Section section_id="1" start_page_num="1" end_page_num=“2”>
<Image image_id="0" page_num="1"/>
<Paragraph page_num="1" first_sentence="NUMBERS, FACTS AND TRENDS SHAPING THE WORLD FOR RELEASE JUNE 8, 2016 "/>
<Image image_id="1" page_num="1"/>
<Image image_id="2" page_num="1"/>
<Paragraph page_num="1" first_sentence="BY Mark Hugo Lopez, Rich Morin and Jens Manuel Krogstad "/>
<Paragraph page_num="1" first_sentence="Mark Hugo Lopez, Director of Hispanic Research Brian Mahl, Communications Assistant 202.419.4372 www.pewresearch.org RECOMMENDED CITATION: Lopez, Mark Hugo, Rich Morin and Jens Manuel Krogstad"/>
· · · **Outline clipped to reduce length**
</Outline>
Can you answer the following question based on the content of the document?
<question>
From this report, which subgroup among Hispanics has gained most confidence from 2008 to 2015?
</question>

Follow these steps to answer the question:
1. As a first step, it might be a good idea to explore the document with the provided tools to familiarize yourself with its structure.
2. Locate the source in the document that can be used to answer the question. Then retrieve the full content of the source in the document with tools to examine it in detail.
3. Find the quote from the document that are most relevant to answering the question, and put it within the <quote></quote> tags. If there are no relevant quotes, write "No relevant quotes" instead.
4. When you gather enough information, return the final concise answer within the <final_result></final_result> tags, leave the explanation outside of the <final_result> tags.

Important guidelines:
- Be aware that the document content is obtained using OCR, so there may be scanning errors or typos.
- Before each step, wrap your thought process in <analysis></analysis> tags. This will help ensure a thorough and accurate analysis of the document and question.
- Please make your final answer as concise as possible. Please provide responses that only use the information you have been given in the document. If the information is unavailable, irrelevant to the question, or if none of the provided options satisfy the specified condition, you should respond with "Not answerable."

Actor:
<analysis>
To answer the question about which subgroup among Hispanics gained the most confidence from 2008 to 2015, I need to locate the relevant section in the document that discusses changes in confidence levels among Hispanic subgroups over this period. According to the outline, Section 9.1.1 titled “Most key Latino demographic subgroups see gains in personal finance ratings since 2008" seems to be the most relevant section. This section likely contains data or analysis on the changes in confidence levels among different Hispanic subgroups from 2008 to 2015. I will retrieve the full content of this section to examine it in detail.
</analysis>
Tool called: get_section_content, parameter: {section_id: 9.1.1}

Tool:
Here is the full text content of Section 9.1.1:
<Section section_id="9.1.1" start_page_num="10" end_page_num="11">
<Heading>Most key Latino demographic subgroups see gains in personal finance ratings since 2008</Heading>
<Paragraph page_num="10">An analysis of 2008 and 2015 survey data finds that Latino perceptions of their economic well-being have increased among most major Latino demographic subgroups. Younger Hispanic adults tended to have bigger gains than older Hispanics. About half (48%) of Latinos ages 18 to 29 report that they were in excellent or good financial shape in 2015, a 27 percentage point increase from 2008 and more than double the share who had a similarly positive view of their Ratings of personal finances improve among most Latino groups % who say their personal financial situation is in excellent or good shape
</Paragraph>
<Image image_id="11" page_num="10"></Image>
· · · **Content clipped to reduce length**
</Section>

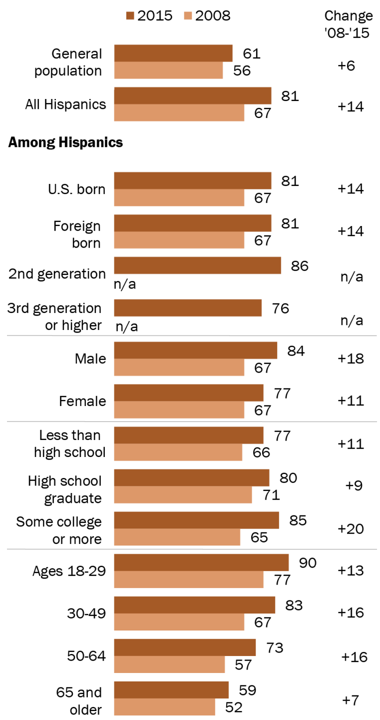
Table 14: Case study of the question answering process of DocAgent (Part 1 of 3).

Actor:
<analysis>
The retrieved content from Section 9.1.1 provides detailed information on the changes in confidence levels among different Hispanic subgroups from 2008 to 2015. According to the section, younger Hispanic adults, specifically those aged 18 to 29, experienced the largest increase in confidence regarding their personal financial situation, with a 27 percentage point increase from 2008 to 2015. This is the most significant gain among the subgroups mentioned. I will now extract the relevant quote and provide the final answer.
</analysis>
<quote>
About half (48%) of Latinos ages 18 to 29 report that they were in excellent or good financial shape in 2015, a 27 percentage point increase from 2008 and more than double the share who had a similarly positive view of their Ratings of personal finances improve among most Latino groups.
</quote>
<final_result>
The subgroup among Hispanics that gained the most confidence from 2008 to 2015 is the younger Hispanic adults aged 18 to 29, with a 27 percentage point increase in confidence.
</final_result>

User:
Now, please validate the answer using the tools to retrieve the source of information that can be used to answer the question. Only use necessary tools. Return the final concise answer within the <final_result></final_result> tags, leave the explanation outside of the <final_result> tags.

Reviewer:
Tool called: get_table_image, parameter: {table_id: 0}

Tool:
Here is the image content for table_id 0:



Reviewer:
<analysis>
The table image confirms that the subgroup "Ages 18-29" experienced a 13 percentage point increase in confidence from 2008 to 2015. However, the subgroup “Some college or more¨shows a 20 percentage point increase, which is higher. Therefore, the subgroup with the most confidence gained is "Some college or more."
</analysis>
<final_result>
The subgroup among Hispanics that gained the most confidence from 2008 to 2015 is those with "Some college or more," with a 20 percentage point increase.
</final_result>

Table 15: Case study of the question answering process of DocAgent (Part 2 of 3).

User: Please update the reflection listed within the <guideline></guideline> tags below that can help you perform better next time. Provide the updated guidance within the <updated_guideline></updated_guideline> tags. Be concise and clear. Ensure the revised guideline deviates from the original by at most one sentence. <guideline></guideline>
Reflection: <updated_guideline> When validating answers, ensure to cross-reference all relevant data sources, such as tables and text, to confirm accuracy and identify any discrepancies. </updated_guideline>
Ground truth: Some college or more

Table 16: Case study of the question answering process of DocAgent (Part 3 of 3). In this example, the actor agent first gives an incorrect answer based solely on the textual source. The reviewer then verifies the chart image and corrects the response. The memory module is updated for future reference. Note in this case, the initial memory is an empty string.