UDA: A Benchmark Suite for Retrieval Augmented Generation in Real-world Document Analysis

Yulong Hui
Tsinghua University
huiyl22@mails.tsinghua.edu.cn

Yao Lu
National University of Singapore
luyao@comp.nus.edu.sg

Huanchen Zhang[*]
Tsinghua University
huanchen@tsinghua.edu.cn

Abstract

The use of Retrieval-Augmented Generation (RAG) has improved Large Language Models (LLMs) in collaborating with external data, yet significant challenges exist in real-world scenarios. In areas such as academic literature and finance question answering, data are often found in raw text and tables in HTML or PDF formats, which can be lengthy and highly unstructured. In this paper, we introduce a benchmark suite, namely Unstructured Document Analysis (UDA), that involves 2,965 real-world documents and 29,590 expert-annotated Q&A pairs. We revisit popular LLM- and RAG-based solutions for document analysis and evaluate the design choices and answer qualities across multiple document domains and diverse query types. Our evaluation yields interesting findings and highlights the importance of data parsing and retrieval. We hope our benchmark can shed light and better serve real-world document analysis applications. The benchmark suite and code can be found at https://github.com/qinchuanhui/UDA-Benchmark.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success yet still face limitations [16]. One of the key challenges is to grapple with external knowledge and previously unseen data, which is a common scenario in real-world applications such as enterprise search and data analysis. For example, a company may need to query its proprietary technique documents; a financial expert may need to extract insights from the latest corporate reports; and a research group may need to assimilate cutting-edge academic papers to guide their innovations. To overcome this challenge, retrieval-augmented generation (RAG) that incorporates relevant content from an external data source into the LLM generation procedure, has emerged as a promising approach [30].

As shown in Figure 1, a common RAG workflow involves the following procedures: 1) parse the external data and segment it into chunks; 2) embed the chunks into vectors and create indexes; 3) retrieve the most relevant chunks according to the user query, and 4) assemble the prompt with relevant chunks (i.e., context) as input for the LLM to generate the response. Recently, a multitude of advanced RAG techniques have been proposed with improved retrieval policies [3, 64], context chunk compression [59, 56], and pre-training strategies [49]. Furthermore, as an alternative approach to RAG, recent advances in long-context LLMs [8, 11, 62] have empowered querying directly on lengthy data without chunking and retrieval.

[*] Huanchen Zhang is also affiliated with the Shanghai Qi Zhi Institute. Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

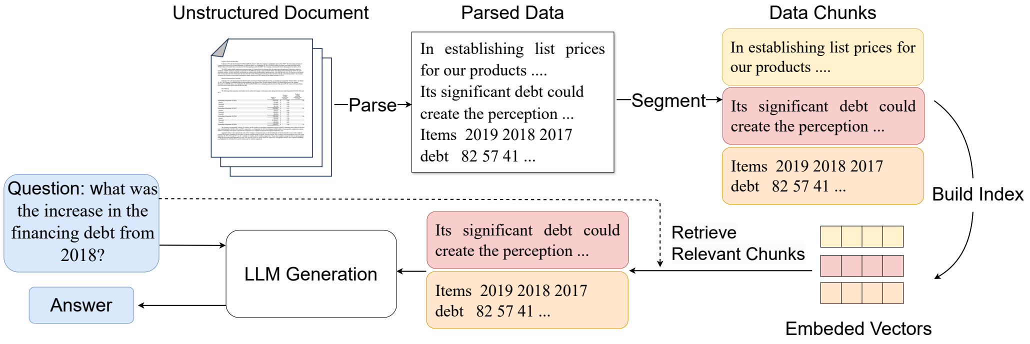

Unstructured Document
Parsed Data
Data Chunks
Parse
In establishing list prices for our products ....
Its significant debt could create the perception ...
Items 2019 2018 2017
debt 82 57 41 ...
Segment
In establishing list prices for our products ....
Its significant debt could create the perception ...
Items 2019 2018 2017
debt 82 57 41 ...
Build Index
Question: what was the increase in the financing debt from 2018?
LLM Generation
Its significant debt could create the perception ...
Items 2019 2018 2017
debt 82 57 41 ...
Retrieve
Relevant Chunks
Answer
Embeded Vectors

Figure 1: An example of basic RAG processing on unstructured documents.

According to a study by Forbes, 95% of business organizations in various domains such as finance and technology need analyzing unstructured texts and tables in raw documents like web pages and PDFs [6]. Analyzing unstructured documents in the world poses the following challenges:

Unstructured inputs. Parsing unstructured documents into regularized text and tables is error-prone. Unlike plain text, unstructured documents often contain intricate layouts and redundant symbols. Prior works [51, 63] have incorporated vision and language models, but their effectiveness is still doubtful. Further, multi-modal data, e.g., tables, require improved indexing and retrieval strategies because classic text embeddings disregard structural information from these data.

Lengthy documents, such as financial reports spanning hundreds of pages, necessitate effective embedding and retrieval mechanisms.

Query answering strategies. User queries span from extractive queries to complex arithmetic reasoning; each may require a different answering strategy, such as using Chain-of-Thought[57] or external tools like Code Interpreters [65]. We wonder how these design choices can impact the end-to-end query answering quality.

In this paper, we propose a benchmark suite that enables the evaluation of various components of RAG-based unstructured document analysis. Specifically, we leverage the Unstructured Document Analysis (UDA) dataset to cover finance, academia, and world knowledge with a total of 2,965 documents and 29,590 expert-annotated Q&A pairs. UDA enables end-to-end evaluations of diverse Q&As and granular analyses of individual components within the RAG pipeline.

Unlike prior datasets and benchmarks which often assume clean or segmented inputs [12, 35, 15], we exploit the design considerations in end-to-end document analysis. We performed extensive experimental analysis to cover different data extraction policies, retrieval and generation strategies, and a range of LLMs. We also compared RAG-based solutions with those that use LLMs with long context capabilities.

Our analysis leads to interesting findings. First, despite that various computer-vision- or language-based parsing techniques have been proposed, conventional solutions may fail to improve the overall Q&A quality due to irregular edge cases. We also found that smaller retrieval models could perform reasonably well in certain RAG applications, while Chain-of-Thought approaches improve the answer quality in zero-shot numerical document analysis. However, long-context LLMs often fall short in these tasks.

We will actively update the benchmark suite and incorporate more state-of-the-art RAG solutions and LLMs in our benchmark. We hope such efforts will shed light on future research and production of RAG- and LLM-based document analysis.

2 Related Work

Retrieval Augmented Generation Large Language Models (LLMs) demonstrate remarkable abilities but struggle with external knowledge and the latest unseen data. Retrieval Augmented Generation (RAG) addresses these limitations by incorporating external information to enrich LLMs' responses,

Table 1: Summary and comparison of Q&A datasets. *Raw documents*: datasets in the native file format, without extraction or parsing; *Long content*: the provision of unsegmented, un-retrieved long content.

Dataset	Text + tables	Raw documents	Long content	Diverse Questions	Sources
HybridQA [10]	✓			✓	Wikipedia
WiKiTableQuestion [39]				✓	Wikipedia
TriviaQA [24]	✓	✓	✓	✓	Wikipedia
VisualMRC [52]	✓			✓	Wikipedia
FinQA [12]	✓				Finance
TAT-DQA [66]	✓	✓		✓	Finance
Qasper [13]	✓		✓	✓	Papers
PDF-VQA [14]	✓	✓	✓		Medicine
DocVQA [35]	✓	✓		✓	**Multiple**
NarrativeQA [26]			✓	✓	**Multiple**
UDA (Ours)	✓	✓	✓	✓	**Multiple**

yielding more precise and credible outputs [44]. Furthermore, several innovative techniques have been developed to refine the procedure beyond the basic RAG approach [16]. For instance, Self-RAG [3] and FLARE [23] determine the retrieved content actively according to the generation results. LLMLingua [22, 21], RECOMP [59], and FILCO [56] focus on filtering and condensing the context input to enhance the information efficiency. Additionally, specialized pre-training and fine-tuning techniques can also optimize the process [31, 48, 49]. Despite these advancements, current approaches often overlook the complexities in real-world unstructured document analysis, such as unstructured data and table schemas, extensive document lengths, and diverse analytical queries.

Prior Benchmarks for RAG offer tools for assessing various dimensions of RAG. RGB [9] benchmarks RAG on robustness and negative rejection. CRUD-RAG [34] introduces a Chinese news dataset for multifaceted evaluation, including text continuation, question answering, and error correction. ALCE [15] evaluates the performance of generating cited responses. In contrast, our benchmark emphasizes the tasks of document comprehension and analysis.

Prior Benchmarks for Q&A often inadequately represent real-world scenarios. For example, mainstream datasets like TriviaQA [24], HotPotQA [61], SQuAD [42], and NaturalQuestions [29] predominantly utilize the Wikipedia sources that have limited application scope and potentially overlap with LLM's internal knowledge. Moreover, datasets such as QuALITY [38] and NarrativeQA [26] are pure plain text, while WikiTableQuestions [39] and SQA [18] are pure tabular data. They fail to capture the complexity of real-world analytical documents. Additionally, datasets like FinQA [12] and VisualMRC [52] present well-structured or segmented content directly, thus sidestepping the intricacies of parsing and retrieval. Table 1 summarizes the features of existing datasets and highlights the uniqueness of our UDA benchmark in real-world document analysis.

3 Dataset: UDA

In this section, we outline the composition and construction of UDA. Each data item within UDA is logically structured as a triplet (D, q, a), where D represents a complete unstructured document, q denotes a question raised from the document, and a signifies the ground truth answer (refer to the data example in Appendix A). To mirror the authenticity of real-world applications, the documents are retained in their original file formats without parsing or segmentation.

Dataset Composition. Our UDA dataset includes six sub-datasets across three pivotal domains: finance, academia, and knowledge bases, reflecting typical use cases in document analysis. As delineated in Table 2, the dataset spans table-based and text-based (or hybrid) QA formats in each domain to ensure that the evaluation covers different data patterns. Moreover, UDA contains 2,965 documents with a wide range of content length and 29,590 expert-annotated Q&A pairs that vary from extractive queries to arithmetic reasoning (see examples in Table 3). These features profoundly embody the breadth and depth of practical real-world applications.

Table 2: An overview of sub-datasets in UDA and their statistics

Domain	Sub Dataset	Doc Format	Doc Num	Q&A Num	Avg #Words	Avg #Pages	Tot Size	Q&A Types
Finance	FinHybrid	PDF	788	8190	76.6k	147.8	2.61 GB	Arithmetic
	TatHybrid	PDF	170	14703	77.5k	148.5	0.58 GB	Extractive, counting, arithmetic
Academic Paper	PaperTab	PDF	307	393	6.1k	11.0	0.22 GB	Extractive, yes/no, free-form
	PaperText	PDF	1087	2804	5.9k	10.6	0.87 GB	Extractive, yes/no, free-form
World Knowledge	FetaTab	PDF & HTML	878	1023	6.0k	14.9	0.92 GB	Free-form
	NqText	PDF & HTML	645	2477	6.1k	14.9	0.68 GB	Extractive

Table 3: Examples of different Q&A types

Q&A Types	Example Question	Example Answer
Extractive	Who has the longest win streak in MMA?	Anderson Silva
Yes/No	Are experiments performed with any other pair of languages?	No
Free-form	How did Hayden Panettiere fare at the 2012 and 2013 Golden Globes?	Hayden Panettiere received two nominations for the Golden Globe Award, Best Supporting Actress Series, Miniseries or Television Film, for her work on Nashville in 2012 and 2013.
Counting	How many regions have revenues of more than $20,000 thousand?	2
Arithmetic	What was the percentage increase in cash dividends from 2015 to 2016?	$(0.29 - 0.25) \div 0.25 * 100\% = 16\%$

Label Collection. We first collect the Q&A labels from the open-released datasets (i.e., source datasets), which are all annotated by human participants. Specifically, our **FinHybrid** is based on the financial numerical reasoning dataset FINQA [12], which is constructed based on the public earnings reports of S&P 500 companies. **TatHybrid** is derived from TAT-DQA [66], whose Q&A pairs are accompanied by the document snapshot of 1 to 3 pages from public financial annual reports. Both **PaperTab** and **PaperText** are based on Qasper [13], a reading comprehension dataset based on NLP research papers. **FetaTab** is built upon FetaQA [36], a question-answering dataset for tables from Wikipedia pages. **NqText** is derived from the widely used Q&A dataset, Natural-Questions [28], which uses the Wikipedia pages as context. Its questions are collected from the Google Search engine, and the answers are human-annotated.

Dataset Construction. We conduct a series of essential constructing actions after collecting the Q&A labels from the source datasets. The integrity of original documents is crucial for the fidelity of document analysis. However, most of the source datasets only offer well-parsed and segmented partial content without the complete document. To address this problem, we perform a comprehensive source-document identification process, including retrieval, verification, and cleaning. (1) Retrieval: the original documents are sourced from certain platforms, such as Wikipedia [58] and arXiv [2]. We conduct retrieval on these platforms using the metadata from the source datasets, such as titles, timestamps, and document types. (2) Verification: we verify the accuracy of the retrieved document by cross-referencing the content fragments in existing datasets, collecting only exact matches. Specifically, we employ the pypdf [41] library for automatic data parsing and comparison, supplemented by manual inspection. (3) Cleaning: we remove the documents that are inaccessible or not available, such as damaged files and withdrawn papers.

Then we embark on a rigorous matching and reorganization effort, enhancing the data quality and forming complete triplet data pairs, i.e., document-question-answer. This involves the following transformations: (1) data cleaning by removing the Q&A pairs or documents lacking essential answers; (2) standardizing diverse data formats into consistently structured tables and JSON files, addressing the heterogeneity of presentation across different sub-datasets; (3) categorizing the queries in Qasper into table-centric and text-centric, thus forming the PaperTab and PaperText subsets for different application patterns; (4) converting HTML-token-based data type into natural language, forming our user-friendly NqText dataset (5) manually annotating some tables within Qasper to serve as references for evaluating parsing strategies.

Table 4: The structure of the sampled datasets used in the following evaluation.

	Sum	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
# Docs	1201	100	150	307	194	300	150
# Q&A pairs	2503	451	450	393	480	350	379

4 Benchmarks and Evaluations

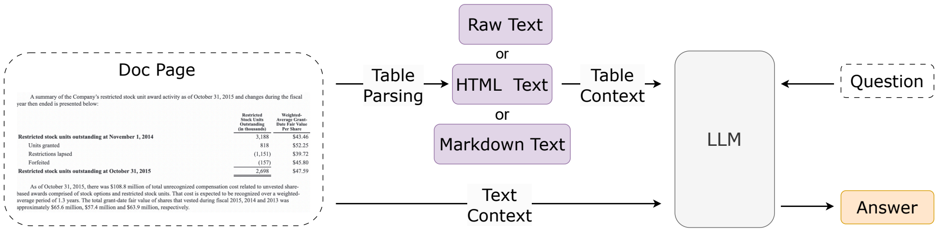
We provide a systematic benchmark of various modules in a typical RAG workflow, as well as an end-to-end LLM-based evaluation. The focused items of our benchmark include:

- The effectiveness of various table-parsing approaches, including raw-text extraction, computer vision (CV)-based, CV-LLM-based and advanced multi-modal parsing (Section 4.1).
- The performance of different indexing and retrieval strategies, spanning sparse retrieval, classic dense embedding, and advanced retrieval model (Section 4.2).
- The influence of precise retrieval on the quality of LLM interpretation (Section 4.2).
- The effectiveness of long-context LLMs compared to typical RAGs (Section 4.3).
- Comparison of different Q&A strategies, such as Chain-of-Thought reasoning and the integration of external code execution (Section 4.4).
- End-to-end comparisons of various LLMs across diverse applications (Section 4.5).

Metrics To evaluate the quality of LLM-generated answers, we apply widely accepted span-level F1-score [42] in PaperTab, PaperText, FetaTab, and NqText datasets, where ground-truth answers are in natural language and the source datasets also utilize this metric. We treat the prediction and ground truth as bags of words and calculate the F1-score to measure their overlap. In financial analysis, the assessment becomes more intricate due to numerical values. For the TatHybrid dataset, we adopt the numeracy-focused F1-score, introduced by Zhu, et al. [67], which considers the scale and the plus-minus of numerical values. In the FinHybrid dataset, where answers are always numerical or binary, we rely on the Exact-Match metric but allow for a numerical tolerance of 1%, accounting for rounding discrepancies. Furthermore, we also incorporate the LLM-based method for a more comprehensive evaluation (more details in Appendix B.5). To assess the effectiveness of retrieval strategies, we identify the factual evidence in retrieved chunks using the relative length of the Longest Common Subsequence (LCS) [40] instead of the exact match, because extracted PDF data chunks often include extraneous symbols that can hinder exact matches while still containing crucial evidence.

Experiment setups In our experiments, we evaluate the performance of various decomposed RAG components, mainly utilizing two representative LLMs: 1) GPT-4 [1], exemplifying the large-scale powerful model, proposed by OpenAI; 2) Llama-3-8B [53], representing the compact yet capable model, proposed by Meta. Furthermore, to ensure a thorough comparative analysis, we also include the end-to-end experiment encompassing a suite of additional LLMs: 3) LLama-3-70B [33], an open-source large-scale model; 4) Qwen-1.5-32B and Qwen-1.5-7B [5], introduced by Alibaba, notable for its 32k token context window; 5) Mixtral-8x7B [20], a Mixture-of-Experts model innovated by MistralAI; 6) Mistral-7B [19], also from MistralAI; 7) CodeLlama-7B and CodeLlama-13B [45], llama models tailored for code generation.

Following prior works in [4, 60, 68], we focus on zero-shot LLM generation, yet add an extra formatting example to align the output with the desired pattern (refer to Appendix B.1). For the GPT-4 model, we leverage the Azure-OpenAI API to access GPT4-Turbo-1106-Preview with the context window of 128k. Other open-source models are obtained from Huggingface, and we always use the instruct-tuned version. The inference is done on 4 NVIDIA-A100 GPUs. To reduce the compute costs, we randomly sample 1201 documents (in PDF format) accompanied by 2503 question-answer pairs to form our evaluation set (detailed in Table 4). We believe it serves as a practical performance indicator for real-world scenarios.



A summary of the Company's restricted stock unit award activity as of October 31, 2015 and changes during the fiscal year then ended is presented below:

	Restricted Stock Units Outstanding (in thousands)	Weighted-Average Grant-Date Fair Value Per Share
Restricted stock units outstanding at November 1, 2014	3,188	$43.46
Units granted	818	$52.25
Restrictions lapsed	(1,151)	$39.72
Forfeited	(157)	$45.80
Restricted stock units outstanding at October 31, 2015	2,698	$47.59

As of October 31, 2015, there was $108.8 million of total unrecognized compensation cost related to unvested share-based awards comprised of stock options and restricted stock units. That cost is expected to be recognized over a weighted-average period of 1.3 years. The total grant-date fair value of shares that vested during fiscal 2015, 2014 and 2013 was approximately $65.6 million, $57.4 million and $63.9 million, respectively.

Figure 2: The procedure of the table parsing experiment

Table 5: Performance scores (EM or F1) of LLMs using varying parsing strategies on table-based Q&A tasks.

Dataset	LLM Name	Well Parsed	GPT-4-Omni	Raw Text	CV	CV + LLM
Tabular FinHybrid (EM)	GPT-4-Turbo	71.9	**72.4**	68.0	61.3	52.4
	Llama-3-8B	**59.5**	56.3	51.6	44.6	40.2
PaperTab (F1)	GPT-4-Turbo	42.8	**44.3**	42.4	38.6	40.7
	Llama-3-8B	35.8	**37.7**	36.5	34.6	32.1

4.1 Evaluating Data Parsing

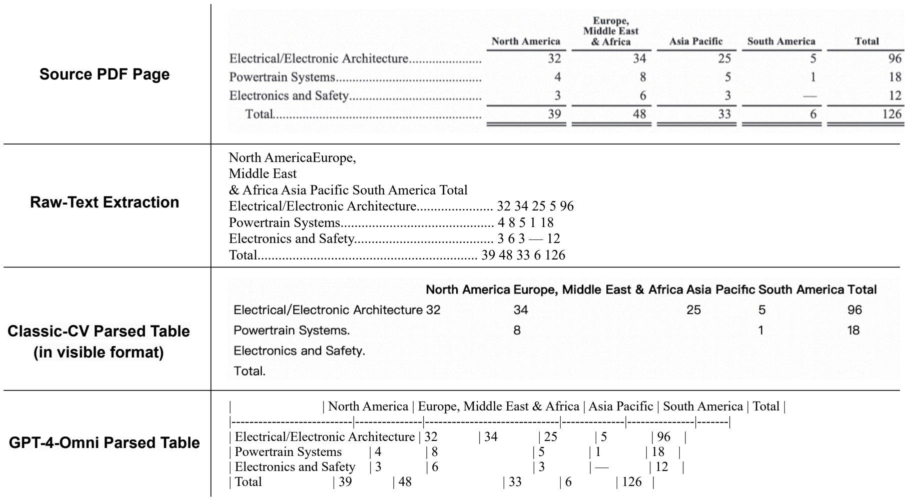
We evaluate various parsing methods to extract tabular information from PDF files and analyze their influence on the downstream tasks, utilizing the table-based questions from PaperTab and FinHybrid (more details in Appendix B.2). As shown in Figure 2, each question is paired with a PDF page that contains the clue tables; doing so prevents inaccurate retrievals. The tabular data are parsed into text and merged with the rest of the text content as the input context to the LLM.

We evaluate several existing approaches of table parsing: (1) **Raw text extraction**, which employs a PDF text extractor [41] to extract all the characters. (2) Classic Computer Vision (**CV**) based approach, which often performs layout detection and OCR extraction at the same time. We follow [55] to use Yolox [17], Tesseract [50] and TableTransformer [51] models together. (3) **CV + LLM** method, which further employs an LLM to transform the outputs of (2) into Markdown tables. (4) For the advanced multi-modal approach, we employ the latest **GPT-4-Omni** [37] to convert image-based document tables into Markdown format. (5) The manually-verified **well-parsed** tables serve as the parsing ground truth.

Table 5 reveals that GPT-4-Omni outperforms other approaches, while surprisingly, raw text extraction also yields decent results. We found that the queried tables are relatively simple; the structural markers from the raw text, such as line-breakers and space, are often adequate for LLMs to understand the table. Classic CV methods, if not meticulously tuned, may struggle in handling non-standard table presentations, i.e., edge cases (see an example in Figure 3). Additionally, employing GPT-4-Omni directly for question-answering scores 69.8 and 35.4, lower than sequentially parsing and generating with GPT-4 (i.e., 72.4 and 44.3).

We also observe from the FinHybrid dataset that the GPT-4 model shows a modest 5.7% improvement with well-parsed data over raw-text data, while the much smaller Llama-3-8B offers a significant 15% enhancement, suggesting that compact models with a limited capability of parsing table layouts, may benefit more from enhanced parsing. In the PaperTab dataset, where completely accurate information is less critical, GPT-4-Omni and raw-text parsing could even outperform well-parsed tables by preserving structural cues that highlight important elements for LLM interpretation.

Remark. Evaluations here suggest (1) CV-based parsing methods may require adaptation for edge cases before they can be useful; (2) smaller LLMs may be impacted more by uncleaned input data, when requiring accurate and specific information.



Source PDF Page

	North America	Europe, Middle East & Africa	Asia Pacific	South America	Total
Electrical/Electronic Architecture......................	32	34	25	5	96
Powertrain Systems...	4	8	5	1	18
Electronics and Safety.......................................	3	6	3	—	12
Total...	39	48	33	6	126

Raw-Text Extraction

North AmericaEurope,
Middle East
& Africa Asia Pacific South America Total
Electrical/Electronic Architecture...................... 32 34 25 5 96
Powertrain Systems... 4 8 5 1 18
Electronics and Safety....................................... 3 6 3 — 12
Total... 39 48 33 6 126

Classic-CV Parsed Table (in visible format)

	North America	Europe, Middle East & Africa	Asia Pacific	South America	Total
Electrical/Electronic Architecture	32	34	25	5	96
Powertrain Systems.		8		1	18
Electronics and Safety.					
Total.					

GPT-4-Omni Parsed Table

```
|                          | North America | Europe, Middle East & Africa | Asia Pacific | South America | Total |
|--------------------------|---------------|------------------------------|--------------|---------------|-------|
| Electrical/Electronic Architecture | 32          | 34          | 25          | 5           | 96   |
| Powertrain Systems        | 4           | 8                    | 5           | 1           | 18   |
| Electronics and Safety    | 3           | 6                    | 3           | —           | 12   |
| Total                 | 39          | 48                   | 33          | 6           | 126  |
```

Figure 3: An example of table parsing with different strategies. Raw-text-extraction preserves the informational content with structural markers; CV-based method may struggle with the irregular table presentation; GPT-4-Omni yields the highest accuracy.

Table 6: Relative LCS scores in different retrieval strategies. We evaluate the presence of evidence in most related 1, 5, 10 and 20 chunks.

Model		FinHybrid				PaperTab				PaperText				FetaTab				NqText			
		@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20
Sparse	BM-25	**65.6**	**83.7**	**87.4**	**90.0**	46.0	79.7	90.0	92.3	47.4	80.0	88.0	89.9	68.3	91.9	**95.2**	**96.2**	42.0	69.2	75.8	80.3
Dense Embedding	all-MiniLM-L6	49.1	71.8	78.2	84.0	51.3	81.7	90.4	92.9	45.7	76.7	85.9	89.9	63.6	90.8	94.4	95.3	49.1	71.1	77.3	80.7
	all-mpnet-base	48.7	74.7	81.7	86.7	50.2	82.2	90.8	92.9	40.8	75.3	86.3	89.8	66.3	91.5	94.6	95.5	50.3	73.4	78.8	81.7
	OpenAI	57.2	80.1	85.2	89.3	**55.5**	**85.4**	**91.8**	**93.0**	**52.2**	**83.1**	**89.0**	**90.3**	**69.7**	**92.7**	95.0	95.7	**50.9**	**74.9**	**80.2**	**82.3**
Advanced	Col-BERT	54.4	75.0	80.4	85.0	47.8	79.7	89.2	92.7	48.4	77.1	86.8	89.8	67.8	91.5	94.6	95.4	47.3	70.3	76.5	80.2

4.2 Indexing and Retrieval

We evaluate the performance of 5 different models under the following retrieval paradigms: 1) **BM-25** [7, 54], a lightweight sparse retrieval method without complex neural networks, ranking document segments based on the appearing frequency of query terms. 2) **all-MiniLM-L6** from SentenceTransformer [43], a prevalent dense embedding model, mapping sentences to a 384-dimensional dense vector space. 3) **all-mpnet-base**, another widely utilized embedding model from SentenceTransformer, noted for its larger architecture and improved performance. 4) **text-embedding-3-large model**, the latest embedding model from **OpenAI**, with enhanced capability. These classic dense embedding models process both query and document segments into vectors, and employ cosine similarity measures to retrieve the most relevant segments. 5) **ColBERT** [46], an advanced retrieval model, relying on token-level embedding and fine-grained contextual late interaction.

We use the relative length of the Longest Common Subsequence (LCS) to demonstrate the presence of human-annotated evidence in retrieved chunks (more experimental details in Appendix B.3). As shown in Table 6, OpenAI's text-embedding-3-large model excels in most datasets except for FinHybrid, where the simpler BM-25 approach intriguingly outperforms. This could be attributed to the fact that financial queries often contain more precise details, such as dates or keywords; this aligns well with the direct keyword-matching of BM-25.

We also conduct end-to-end experiments to verify the impact of the retrieval quality. We evaluate the answer quality with top-5 chunks retrieved using OpenAI embeddings or with the human-annotated evidential chunks. As shown in Table 7, providing more relevant context to LLMs improves the answers in most cases, particularly in arithmetic-reasoning tasks such as FinHybrid and TatHybrid. Interestingly, for the FinHybrid dataset, the Llama-3-8B model achieved a score of 51.0 when given accurate context, outperforming GPT-4 with model-retrieved chunks. However, for knowledge-based questions from the NqText and FetaTab, the answer quality remains less affected. This is because complex numerical reasoning demands more precise evidence for accurate arithmetic operations,

Table 7: End-to-end answer scores using retrieved and human-annotated context

LLM Name	Context Type	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
Llama-3-8B	OpenAI Retrieval @5	37.9	22.5	**35.5**	42.3	56.6	**31.7**
	Human-annotated	**51.0**	**35.9**	35.1	**46.4**	**57.5**	31.5
	Improvement	35%	60%	-1%	10%	2%	-1%
GPT-4-Turbo	OpenAI Retrieval @5	45.9	43.5	40.3	45.8	**61.5**	37.4
	Human-annotated	**69.4**	**57.7**	**42.0**	**56.5**	59.5	**39.0**
	Improvement	51%	33%	4%	23%	-3%	4%

Table 8: Performance scores between long-context and RAG mechanism.

LLM Name	Input Type	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
Qwen-1.5-7B	OpenAI Retrieval @5	**21.0**	**26.6**	**31.4**	**39.1**	58.1	**32.4**
	Long Context	3.0	20.9	26.3	33.1	**58.7**	30.2
GPT-4-Turbo	OpenAI Retrieval @5	**43.4**	**46.3**	**43.5**	47.1	61.8	**35.8**
	Long Context	37.4	36.9	43.3	**47.4**	**63.3**	35.4

whereas LLMs can leverage a wide range of narrative information to derive answers to knowledge-based questions.

Remark. We found that the model scaling law may not hold true in retrieval scenarios. The retrieval quality does matter, particularly in arithmetic tasks, but the incremental benefit of including additional chunks diminishes. Some use cases (e.g., queries that involve exact entity matching) may prefer some specific data embedding or indexing mechanisms.

4.3 RAG vs. Long Context

We compare RAG-based methods with long-context LLMs, utilizing GPT-4-Turbo with a 128k context window and Qwen-1.5-7B with a 32k context window. Due to the high cost of long context inference, we conduct this experiment on a subset of 600 documents (more details in Appendix B.4). The results are demonstrated in Table 8.

Remark. We notice that for free-form or knowledge-based tasks (i.e., paper-based and wiki-based Q&A), RAG and long-context solutions demonstrate comparable capability. Conversely, in tasks with more numerical reasoning (i.e., financial Q&A), long-context LLMs fail to match the performance of RAG. In such use cases, the excess of verbose content might hinder long-context LLMs in pinpointing facts and performing numerical reasoning effectively (see an example in Table 10). Additionally, RAG is likely to surpass the long-context mechanism more in the smaller LLM, given its constrained capacity to handle large volumes of data.

4.4 Evaluating Chain-of-Thought and Code Interpreters

In real-world analytical queries, reasoning capabilities can be essential, yet LLMs face limitations in this area [25]. To overcome these constraints, advanced methods such as Chain-of-Thought (CoT) [57] and Code-Interpreter (CI) [65] have been introduced. CoT prompts LLMs to generate a series of intermediate reasoning steps, whereas CI lets the LLM produce executable codes and then invoke an external executor to derive the answer. In this section, we evaluate the efficacy of basic generation, the CoT approach, and CI methods using the numerical reasoning dataset FinHybrid. We use the top-5 chunks retrieved with OpenAI's embedding as the context and benchmark GPT-4-Turbo, Llama-3-8B, and the code-tailored CodeLlama models. The CoT approaches are implemented with step-wise instructive prompts, while the basic CI method asks LLMs to produce Python codes if necessary (more details in Appendix B.1).

Remark. As illustrated in Table 9, the Chain-of-Thought (CoT) approach outperforms others across all model configurations (see an example in Table 10). The Llama-3-8B model shows improvements when incorporating codes, yet CoT methods still prove superior. CodeLlama models, however, struggle with generating viable code in this scenario with lengthy context and ambiguous code-gen instructions. GPT-4-turbo exhibits a native ability to produce step-by-step explanations, leading to

Table 9: Exact-match scores of different generation strategies on the FinHybrid dataset

LLM Name	Base	CoT	Code
Llama-3-8B	21.3	**37.9**	26.4
CodeLlama-7B	5.5	**7.3**	5.5
CodeLlama-13B	10.6	**11.3**	9.1
GPT-4-Turbo	**45.9**	**45.9**	32.2

Table 10: Case study of long-context and generating strategy. RAG outperforms the long-context through more accurate evidence retrieval, and CoT's superiority is attributed to the integration of explicit reasoning steps.

Question	LLM Name	Strategy	LLM's Response
What percentage of contractual obligations is due to maturities of long-term debt?	Qwen-1.5-7B	long-context	33% ✗
		RAG	690 million out of 1416 million, 49% ✓
What percent of the share-based compensation expense was related to stock options?	Llama-3-8B	basic-gen	100% ✗
		code	94.4% (no code is generated, direct response) ✗
		CoT	To find the percentage, we can divide the portion related to stock options by the total share-based compensation expense ($7 million / $36 million) x 100 = 19.4% ✓

Table 11: End-to-end performance scores of different LLMs

LLM Name	Avg	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
GPT-4-turbo	**45.7**	**45.9**	**43.5**	**40.3**	**45.8**	61.5	37.4
Llama-3-70B	42.5	43.5	30.9	38.7	44.4	63.3	33.9
GPT-3.5	40.9	36.6	33.9	35.5	42.1	**64.1**	33.1
Qwen-1.5-32B	38.9	31.3	27.9	31.6	43.1	58.4	**41.3**
Llama-3-8B	37.7	37.9	22.5	35.5	42.3	56.6	31.7
Mixtral-8x7B-v0.1	34.5	28.4	22.5	35.0	38.1	54.1	29.1
Qwen-1.5-7B	33.8	17.0	22.6	28.0	37.7	58.6	38.9
Mistral-7B-v0.2	26.6	18.2	15.9	20.9	22.7	54.3	27.3
Llama-3-8B-NoRAG	18.0	3.6	6.0	13.0	16.2	47.7	21.7
GPT-4-turbo-NoRAG	17.6	0.4	3.0	15.3	18.9	48.6	19.6

equivalent performance between basic generation and CoT methods. It is worth noting that while code interpreter has been demonstrated capable of handling numerical and tabular data [27, 68], its efficacy in document analysis is hindered by the unstructured tables and lengthy context, if just using the basic code strategy.

4.5 End-to-End Evaluations

Assembling the insights derived from the above analyses, we construct an end-to-end RAG pipeline. Specifically, we parse unstructured data from PDFs leveraging the raw-text extraction approach, and employ OpenAI's text-embedding-3-large model to index and retrieve relevant data chunks. In the generation phase, we incorporate the Chain-of-Thought approach to handle arithmetic-intensive tasks. Based on this end-to-end pipeline, we evaluate 8 LLMs spanning various model sizes and architectures.

Remark. Table 11 presents the results, where GPT-4 leads in overall performance. GPT-3.5 and Qwen-1.5-32B excel in FetaTab and NqText, respectively. The overall end-to-end scores demonstrate the challenges of our benchmark and also indicate considerable room for improvement in real-world document analysis for both RAG and LLMs.

Additionally, Table 12 presents the results when the LLMs respond to the questions with versus without RAG support. A significant decline in accuracy is observed when RAG is absent, underscoring

Table 12: Performance scores with and without RAG

LLM and Strategy	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
GPT-4	**45.9**	**43.5**	**40.3**	**45.8**	**61.5**	**37.4**
GPT-4-NoRAG	0.4	3.0	15.3	18.9	48.6	19.6
Llama-3-8B	**37.9**	**22.5**	**35.5**	**42.3**	**56.6**	**31.7**
Llama-3-8B-NoRAG	3.6	6.0	13.0	16.2	47.7	21.7

the LLMs' deficiency of related internal knowledge and their reliance on external documents. This reinforces the effectiveness of our benchmark for the evaluation and exploration of RAG approaches.

5 Conclusion

In this paper, we propose a novel benchmark to assess Retrieval Augmented Generation (RAG) methodologies in real-world document analysis scenarios. Our benchmark features diverse question types and encompasses thousands of unstructured documents with expert labels from financial and other domains. Meanwhile, we discuss interesting findings from our evaluations, covering data parsing, information retrieval, long context mechanism, generating strategies and end-to-end performance. We believe our benchmark will advance future research and production in unstructured document analysis.

6 Limitations

Despite the valuable contributions of this study, we acknowledge its limitations: (1) While the efficacy of parsing strategies was evaluated through the downstream Q&A performance, a direct comparison of the parsed content was not undertaken. This stems from the absence of well-defined standards for direct quality assessment in the scenario of document understanding. (2) This study did not extend to the in-depth analyses of noise sensitivity and hallucination. We will delve into these topics and conduct detailed analyses in our future work.

References

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] ArXiv. Arxiv platform, 2024.

[3] Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations* (2024).

[4] Baek, J., Aji, A. F., and Saffari, A. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136* (2023).

[5] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[6] Baviskar, D., Ahirrao, S., Potdar, V., and Kotecha, K. Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access 9* (2021), 72894–72936.

[7] Brown, D. Rank-bm25: A two line search engine, 2022.

[8] Cai, Z., Cao, M., Chen, H., Chen, K., Chen, K., Chen, X., Chen, X., Chen, Z., Chen, Z., Chu, P., et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297* (2024).

[9] Chen, J., Lin, H., Han, X., and Sun, L. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 17754–17762.

[10] Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., and Wang, W. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347* (2020).

[11] Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., and Jia, J. LongloRA: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations* (2024).

[12] Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.-H., Routledge, B., et al. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122* (2021).

[13] Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N. A., and Gardner, M. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011* (2021).

[14] Ding, Y., Luo, S., Chung, H., and Han, S. C. Vqa: A new dataset for real-world vqa on pdf documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2023), Springer, pp. 585–601.

[15] Gao, T., Yen, H., Yu, J., and Chen, D. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (Singapore, Dec. 2023), H. Bouamor, J. Pino, and K. Bali, Eds., Association for Computational Linguistics, pp. 6465–6488.

[16] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[17] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430* (2021).

[18] Iyyer, M., Yih, W.-t., and Chang, M.-W. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017), pp. 1821–1831.

[19] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825* (2023).

[20] JIANG, A. Q., SABLAYROLLES, A., ROUX, A., MENSCH, A., SAVARY, B., BAMFORD, C., CHAPLOT, D. S., CASAS, D. D. L., HANNA, E. B., BRESSAND, F., ET AL. Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).

[21] JIANG, H., WU, Q., LIN, C.-Y., YANG, Y., AND QIU, L. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (Singapore, Dec. 2023), H. Bouamor, J. Pino, and K. Bali, Eds., Association for Computational Linguistics, pp. 13358–13376.

[22] JIANG, H., WU, Q., LUO, X., LI, D., LIN, C.-Y., YANG, Y., AND QIU, L. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839* (2023).

[23] JIANG, Z., XU, F., GAO, L., SUN, Z., LIU, Q., DWIVEDI-YU, J., YANG, Y., CALLAN, J., AND NEUBIG, G. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (Singapore, Dec. 2023), Association for Computational Linguistics, pp. 7969–7992.

[24] JOSHI, M., CHOI, E., WELD, D., AND ZETTLEMOYER, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vancouver, Canada, July 2017), R. Barzilay and M.-Y. Kan, Eds., Association for Computational Linguistics, pp. 1601–1611.

[25] KADDOUR, J., HARRIS, J., MOZES, M., BRADLEY, H., RAILEANU, R., AND MCHARDY, R. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* (2023).

[26] KOČISKÝ, T., SCHWARZ, J., BLUNSOM, P., DYER, C., HERMANN, K. M., MELIS, G., AND GREFENSTETTE, E. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics 6* (2018), 317–328.

[27] KONG, K., ZHANG, J., SHEN, Z., SRINIVASAN, B., LEI, C., FALOUTSOS, C., RANGWALA, H., AND KARYPIS, G. Opentab: Advancing large language models as open-domain table reasoners. In *The Twelfth International Conference on Learning Representations* (2024).

[28] KWIATKOWSKI, T., PALOMAKI, J., REDFIELD, O., COLLINS, M., PARIKH, A., ALBERTI, C., EPSTEIN, D., POLOSUKHIN, I., DEVLIN, J., LEE, K., ET AL. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics 7* (2019), 453–466.

[29] KWIATKOWSKI, T., PALOMAKI, J., REDFIELD, O., COLLINS, M., PARIKH, A., ALBERTI, C., EPSTEIN, D., POLOSUKHIN, I., DEVLIN, J., LEE, K., TOUTANOVA, K., JONES, L., KELCEY, M., CHANG, M.-W., DAI, A. M., USZKOREIT, J., LE, Q., AND PETROV, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics 7* (2019), 452–466.

[30] LEWIS, P., PEREZ, E., PIKTUS, A., PETRONI, F., KARPUKHIN, V., GOYAL, N., KÜTTLER, H., LEWIS, M., YIH, W.-T., ROCKTÄSCHEL, T., ET AL. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems 33* (2020), 9459–9474.

[31] LI, X., LIU, Z., XIONG, C., YU, S., GU, Y., LIU, Z., AND YU, G. Structure-aware language model pretraining improves dense retrieval on structured data. In *Findings of the Association for Computational Linguistics: ACL 2023* (Toronto, Canada, 2023), A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Association for Computational Linguistics, pp. 11560–11574.

[32] LIU, Y., YANG, T., HUANG, S., ZHANG, Z., HUANG, H., WEI, F., DENG, W., SUN, F., AND ZHANG, Q. Calibrating LLM-based evaluator. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (Torino, Italia, May 2024), ELRA and ICCL, pp. 2638–2656.

[33] LLAMA, M. Meta-llama-3, 2024.

[34] LYU, Y., LI, Z., NIU, S., XIONG, F., TANG, B., WANG, W., WU, H., LIU, H., XU, T., AND CHEN, E. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *arXiv preprint arXiv:2401.17043* (2024).

[35] Mathew, M., Karatzas, D., and Jawahar, C. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (2021), pp. 2200–2209.

[36] Nan, L., Hsieh, C., Mao, Z., Lin, X. V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., et al. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics 10* (2022), 35–49.

[37] OpenAI. Gpt-4o, 2024.

[38] Pang, R. Y., Parrish, A., Joshi, N., Nangia, N., Phang, J., Chen, A., Padmakumar, V., Ma, J., Thompson, J., He, H., and Bowman, S. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Seattle, United States, July 2022), M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Association for Computational Linguistics, pp. 5336–5358.

[39] Pasupat, P., and Liang, P. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305* (2015).

[40] Paterson, M., and Dančík, V. Longest common subsequences. In *International symposium on mathematical foundations of computer science* (1994), Springer, pp. 127–142.

[41] PyPDF. Py-pdf: A pure-python pdf library capable of splitting, merging, cropping, and transforming the pages of pdf files, 2024.

[42] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Austin, Texas, Nov. 2016), Association for Computational Linguistics, pp. 2383–2392.

[43] Reimers, N., and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[44] Ren, R., Wang, Y., Qu, Y., Zhao, W. X., Liu, J., Tian, H., Wu, H., Wen, J.-R., and Wang, H. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019* (2023).

[45] Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950* (2023).

[46] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Seattle, United States, July 2022), Association for Computational Linguistics, pp. 3715–3734.

[47] Sarmah, B., Mehta, D., Pasquali, S., and Zhu, T. Towards reducing hallucination in extracting information from financial reports using large language models. In *Proceedings of the Third International Conference on AI-ML Systems* (2023), pp. 1–5.

[48] Shi, T., Li, L., Lin, Z., Yang, T., Quan, X., and Wang, Q. Dual-feedback knowledge retrieval for task-oriented dialogue systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (Singapore, Dec. 2023), Association for Computational Linguistics, pp. 6566–6580.

[49] Shi, W., Min, S., Lomeli, M., Zhou, C., Li, M., Lin, X. V., Smith, N. A., Zettlemoyer, L., tau Yih, W., and Lewis, M. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations* (2024).

[50] Smith, R. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)* (2007), vol. 2, IEEE, pp. 629–633.

[51] Smock, B., Pesala, R., and Abraham, R. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 4634–4642.

[52] TANAKA, R., NISHIDA, K., AND YOSHIDA, S. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 13878–13888.

[53] TOUVRON, H., LAVRIL, T., IZACARD, G., MARTINET, X., LACHAUX, M.-A., LACROIX, T., ROZIÈRE, B., GOYAL, N., HAMBRO, E., AZHAR, F., ET AL. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[54] TROTMAN, A., PUURULA, A., AND BURGESS, B. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium* (2014), pp. 58–65.

[55] UNSTRUCTURED. Open-source pre-processing tools for unstructured data, 2024.

[56] WANG, Z., ARAKI, J., JIANG, Z., PARVEZ, M. R., AND NEUBIG, G. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377* (2023).

[57] WEI, J., WANG, X., SCHUURMANS, D., BOSMA, M., BRIAN ICHTER, XIA, F., CHI, E. H., LE, Q. V., AND ZHOU, D. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems* (2022), A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds.

[58] WIKIPEDIA. Wikipedia: a free online encyclopedia, 2024.

[59] XU, F., SHI, W., AND CHOI, E. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations* (2024).

[60] XU, P., PING, W., WU, X., MCAFEE, L., ZHU, C., LIU, Z., SUBRAMANIAN, S., BAKHTURINA, E., SHOEYBI, M., AND CATANZARO, B. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations* (2024).

[61] YANG, Z., QI, P., ZHANG, S., BENGIO, Y., COHEN, W., SALAKHUTDINOV, R., AND MANNING, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 2369–2380.

[62] YOUNG, A., CHEN, B., LI, C., HUANG, C., ZHANG, G., ZHANG, G., LI, H., ZHU, J., CHEN, J., CHANG, J., ET AL. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652* (2024).

[63] YU, T., WU, C.-S., LIN, X. V., WANG, B., TAN, Y. C., YANG, X., RADEV, D., SOCHER, R., AND XIONG, C. Grappa: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations* (2021).

[64] YU, Z., XIONG, C., YU, S., AND LIU, Z. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of ACL* (2023).

[65] ZHOU, A., WANG, K., LU, Z., SHI, W., LUO, S., QIN, Z., LU, S., JIA, A., SONG, L., ZHAN, M., ET AL. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921* (2023).

[66] ZHU, F., LEI, W., FENG, F., WANG, C., ZHANG, H., AND CHUA, T.-S. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia* (2022), pp. 4857–4866.

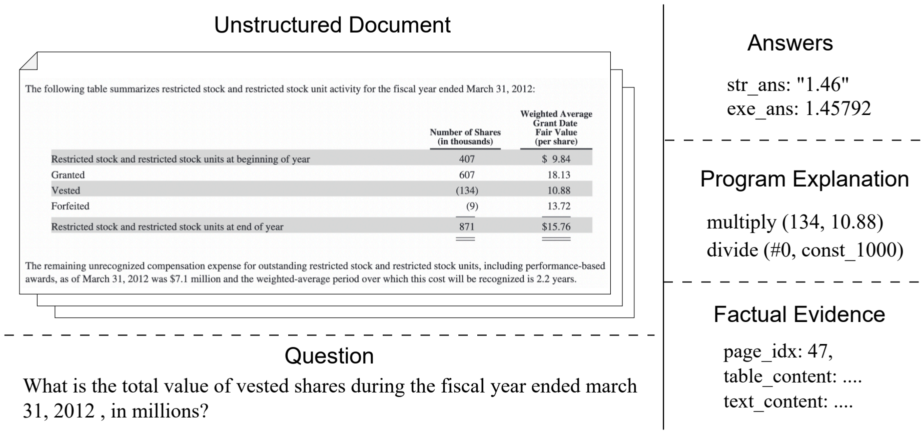
[67] ZHU, F., LEI, W., HUANG, Y., WANG, C., ZHANG, S., LV, J., FENG, F., AND CHUA, T.-S. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, Aug. 2021), Association for Computational Linguistics.

[68] ZHU, F., LIU, Z., FENG, F., WANG, C., LI, M., AND CHUA, T.-S. Tat-llm: A specialized language model for discrete reasoning over tabular and textual data. *arXiv preprint arXiv:2401.13223* (2024).

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In supplemental material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Both in the supplemental material and as a URL in Abstract.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] There's no training process in this paper.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Due to the significant computational cost, the evaluation of LLMs often doesn't need multiple experimental iterations, supported by extensive prior works.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3
   (b) Did you mention the license of the assets? [Yes] In supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Both in the supplemental material and as a URL in Abstract.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] In supplemental material.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] In supplemental material.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Dataset Example



The following table summarizes restricted stock and restricted stock unit activity for the fiscal year ended March 31, 2012:

	Number of Shares (in thousands)	Weighted Average Grant Date Fair Value (per share)
Restricted stock and restricted stock units at beginning of year	407	$ 9.84
Granted	607	18.13
Vested	(134)	10.88
Forfeited	(9)	13.72
Restricted stock and restricted stock units at end of year	871	$15.76

The remaining unrecognized compensation expense for outstanding restricted stock and restricted stock units, including performance-based awards, as of March 31, 2012 was $7.1 million and the weighted-average period over which this cost will be recognized is 2.2 years.

What is the total value of vested shares during the fiscal year ended march 31, 2012 , in millions?

str_ans: "1.46"
exe_ans: 1.45792

multiply (134, 10.88)
divide (#0, const_1000)

page_idx: 47,
table_content: ....
text_content: ....

Figure 4: Data example of FinHybrid in UDA

Figure 4 presents a data item from the FinHybrid dataset, a prototypical representation common to all datasets within UDA. It comprises an original multi-page unstructured document containing tables and text, accompanied by related question-answer pairs. The datasets within UDA also feature additional explanations in different formats, such as the programmatic operation and factual evidence in FinHybrid.

B Experimental Details

B.1 Prompt Templates for LLMs

Table 13 presents the prompt templates for LLM. We use the Chain-of-Thought approach for FinHybrid and TatHybrid datasets and also explore basic generation and Code-Interpreter strategies for FinHybrid, as detailed in Section 4.4. The example instruction of the prompt omits full context, just using a Q&A pair to guide LLM output toward the target answer format.

B.2 Settings of Parsing Experiments

In our parsing experiments, we utilize 341 table-based questions from FinHybrid and 100 questions from PaperTab. FinHybrid provides cleanly extracted tables which we formatted into Markdown text as our well-parsed ground truth. To get the well-parsed content in PaperTab, we first use GPT-4-Omni to convert table images to Markdown, followed by manual corrections for inaccuracies. Since Markdown doesn't support the nested structure, we replicate the root content to represent hierarchical relationships.

B.3 Settings of Indexing and Retrieval Experiments

In our indexing and retrieval experiments, we first extract raw text from documents, then segment it into 3000-character (about 500 words) chunks using the recursive-split method [47], ensuring a 10% overlap to mitigate information loss. Then the index is constructed on these chunks for the retrieval task. For evaluation, human-annotated factual evidence serves as the ground truth for retrieval. We measure evidence presence by calculating the ratio of the Longest Common Subsequence (LCS) length to the full evidence length.

B.4 Long-context Policy

In our long-context experiments, we employ the Qwen-1.5-7B-32k and GPT4-Turbo-128k models, which can handle extended contexts but are still insufficient for quite long documents, such as financial reports exceeding 100k words (more than 150k tokens). For such cases, the strategy falls

Table 13: Prompt templates for LLMs in each dataset and task.

Dataset	Prompt Template	
PaperTab and PaperText	System	You are a scientific researcher, given a section of an academic paper, please answer the question according to the context of the paper. The final answer output should be in the format of "The answer is: <answer>", and the <answer> should be concise with no explanation.
	User	### Context: ... ### Question: Which Indian languages do they experiment with? ### Response:
	Assistant	The answer is: Hindi, English, Kannada, Telugu, Assamese, Bengali and Malayalam.
	User	### Context: {context} ### Question: {question} ### Response:
FetaTab	System	Given a section of a document, plese answer the question according to the context. The final answer output should be in the format of "The answer is: <answer>", and the <answer> should be a natural sentence.
	User	### Context: ... ### Question: When and in what play did Platt appear at the Music Box Theatre? with? ### Response:
	Assistant	The answer is: In 2016 and 2017, Platt played in Dear Evan Hansen on Broadway at the Music Box Theatre.
	User	### Context: {context} ### Question: {question} ### Response:
NqText	System	Given a section of a document, plese answer the question according to the context. The final answer output should be in the format of "The answer is: <answer>", and the <answer> should be a paragraph from the context or a summarized short phrase.
	User	### Context: ... ### Question: When will tour de france teams be announced? ### Response:
	Assistant	The answer is: 6 January 2018
	User	### Context: {context} ### Question: {question} ### Response:
FinHybrid and TatHybird (Chain-of-Thought)	System	You are a financial analyzer, given a section of a company's annual report, please answer the question according to the report context. Let's do this step by step. The final answer output should be in the format of "The answer is: <answer>", and the <answer> must be simple and short (e.g. just an accurate numerical value or phrases).
	User	### Context: ... ### Question: What is the average price of the products? ### Response:
	Assistant	There are 8 products with a total price value of 1000 so the average value is 125.00. The answer is: 125.00
	User	### Context: {context} ### Question: {question} ### Response:
FinHybrid (Basic Generation)	System	You are a financial analyzer, given a section of a company's annual report, please answer the question according to the report context. The final answer output should be in the format of 'The answer is: <answer>', and the <answer> must be simple and short (e.g. just an accurate numerical value or phrases).
	User	### Context: ... ### Question: What is the average price of the products? ### Response:
	Assistant	The answer is: 125.00
	User	### Context: {context} ### Question: {question} ### Response:
FinHybrid (Code Interpreter)	System	Given a section of a company's annual report and corresponding question, please generate the python codes to calculate the answer. You should firstly extract and list the relevant information from the context, and then write the arithmetical python codes in the following block format: ```python <python codes> ``` If the answer does not require any calculation, you should directly write the answer in the format of "The answer is: <answer>".
	User	### Context: ... ### Question: What is the average price of the products? ### Response:
	Assistant	The price of product 1,2,3,4 is 700, and the price of product 5,6,7,8 is 900. The python code is ```python products = [700, 700, 700, 700, 900, 900, 900, 900] \n total_price = sum(products) \n average_price = total_price / len(products) \n print(average_price)```
	User	### Context: ... ### Question: What is the net income in 2009? ### Response:
	Assistant	The answer is: 1000 million
	User	### Context: {context} ### Question: {question} ### Response:

back to retrieving the top 30 most relevant data segments to serve as the input context. Due to the high cost of long context inference, we conduct this experiment on a subset of 600 documents, each coupled with a corresponding Q&A pair.

Table 14: Performance scores (Exact Match or LLM-Score) of LLMs using varying parsing strategies on table-based Q&A tasks (supplement to Table 5).

Dataset	LLM Name	Well Parsed	GPT-4-Omni	Raw Text	CV	CV + LLM
Tabular FinHybrid (EM)	GPT-4-Turbo	71.9	**72.4**	68.0	61.3	52.4
	Llama-3-8B	**59.5**	56.3	51.6	44.6	40.2
PaperTab (LLM-Score)	GPT-4-Turbo	54.3	**56.3**	54.3	53.5	52.5
	Llama-3-8B	**46.8**	41.3	40.3	36.0	37.0

Table 15: End-to-end answer scores using retrieved and human-annotated context (with the LLM evaluator, supplement to Table 7).

LLM Name	Context Type	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
Llama-3-8B	OpenAI Retrieval @5	37.9	22.5	**39.9**	**43.2**	**69.0**	82.5
	Human-annotated	**51.0**	**35.9**	39.1	41.2	68.0	**85.2**
GPT-4-Turbo	OpenAI Retrieval @5	45.9	43.5	51.8	53.2	**80.4**	81.7
	Human-annotated	**69.4**	**57.7**	**53.0**	**60.3**	75.0	**81.8**

Table 16: Performance scores between long-context and RAG mechanism (with the LLM evaluator, supplement to Table 8).

LLM Name	Input Type	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
Qwen-1.5-7B	OpenAI Retrieval @5	**21.0**	**26.6**	40.0	44.2	**68.0**	73.0
	Long Context	3.0	20.9	**40.8**	**48.0**	**68.0**	**77.3**
GPT-4-Turbo	OpenAI Retrieval @5	**43.4**	**46.3**	**56.5**	57.1	**81.3**	77.8
	Long Context	37.4	36.9	50.0	**57.8**	81.0	**78.8**

Table 17: End-to-end performance scores of different LLMs (with the LLM evaluator, supplement to Table 11).

LLM Name	Avg	FinHybrid	TatHybrid	PaperTab	PaperText	FetaTab	NqText
GPT-4-turbo	**59.4**	**45.9**	**43.5**	**51.8**	**53.2**	**80.4**	81.7
GPT-3.5	55.3	36.6	33.9	46.2	51.1	76.1	**87.9**
Llama-3-70B	52.4	43.5	30.9	44.3	43.3	72.4	80.3
Mixtral-8x7B-v0.1	50.0	28.4	22.5	48.2	50.7	67.9	82.5
Llama-3-8B	49.2	37.9	22.5	39.9	43.2	69.0	82.5
Qwen-1.5-32B	45.8	31.3	27.9	39.7	43.5	66.5	65.6
Mistral-7B-v0.2	45.1	18.2	15.9	41.8	45.3	66.3	83.0
Qwen-1.5-7B	43.6	17.0	22.6	39.5	44.9	66.1	71.7

B.5 LLM Evaluator

We also incorporate the LLM-based method for a more comprehensive evaluation. Following the previous work [32], we implement our LLM-based evaluator with few-shot in-context learning. Given the question, ground-truth answer, and the generated response, the LLM evaluator scores the response from 0 to 4 based on correctness. For the FinHybrid and TatHybrid datasets, where the answers are digits or extractive words, the existing numeric or word-level matching is sufficient for scoring. For the remaining datasets, PaperTab, PaperText, FetaTab, and NqText, with natural-language answers, we apply the LLM evaluator to re-evaluate all of our experiments. The results are illustrated from Table 14 to Table 17, where the scores are normalized to the 100-scale for clarity.

Our findings are further validated by the results from the LLM evaluator. For example, the long-context and RAG mechanism yield comparable results on free-form and knowledge-based tasks, and GPT-4-omni and raw-text parsing methods demonstrate decent performance. Unlike the rule-based evaluation, the LLM evaluator works for varied but similar expressions and produces generally higher absolute scores, while it may decrease the interpretability of how the scores are determined.