Enhancing Financial RAG with Agentic AI and Multi-HyDE: A Novel Approach to Knowledge Retrieval and Hallucination Reduction

Akshay Govind Srinivasan[*1], Ryan Jacob George[*1], Jayden Koshy Joe[*1], Hrushikesh Kant[*1], Harshith M R[*1], Sachin Sundar[*1], Sudharshan Suresh[*1], Rahul Vimalkanth[*1], Vijayavallabh[*1]

[1]Indian Institute of Technology Madras

Abstract

Accurate and reliable knowledge retrieval is vital for financial question-answering, where continually updated data sources and complex, high-stakes contexts demand precision. Traditional retrieval systems rely on a single database and retriever, but financial applications require more sophisticated approaches to handle intricate regulatory filings, market analyses, and extensive multi-year reports. We introduce a framework for financial Retrieval Augmented Generation (RAG) that leverages agentic AI and the Multi-HyDE system, an approach that generates multiple, nonequivalent queries to boost the effectiveness and coverage of retrieval from large, structured financial corpora. Our pipeline is optimized for token efficiency and multi-step financial reasoning, and we demonstrate that their combination improves accuracy by 11.2% and reduces hallucinations by 15%. Our method is evaluated on standard financial QA benchmarks, showing that integrating domain-specific retrieval mechanisms such as Multi-HyDE with robust toolsets, including keyword and table-based retrieval, significantly enhances both the accuracy and reliability of answers. This research not only delivers a modular, adaptable retrieval framework for finance but also highlights the importance of structured agent workflows and multi-perspective retrieval for trustworthy deployment of AI in high-stakes financial applications.

1 Introduction

Large Language Models (LLMs) such as GPT-4 (OpenAI et al., 2024), LLaMA (Touvron et al., 2023), and PaLM (Chowdhery et al., 2022) have significantly advanced natural language processing, demonstrating strong capabilities in contextual reasoning and few-shot learning. These models are increasingly applied in high-stakes domains, including healthcare diagnostics (Singhal et al., 2023), legal document analysis (Henderson et al., 2023), and financial services (Wu et al., 2023; Li et al., 2023). Their ability to process and generate domain-specific, human-like responses offers clear potential benefits.

However, a persistent limitation of LLMs is *hallucination* - the generation of factually incorrect or fabricated content presented as truth (Ji et al., 2023; Huang et al., 2023). This limitation poses significant risks in domains where factual accuracy is paramount. In domains such as finance, where decisions must be based on accurate and verifiable data, hallucinations can lead to significant monetary losses, reputational harm, and regulatory violations.

Retrieval-Augmented Generation (RAG) frameworks (Lewis et al., 2020; Guu et al., 2020) address this issue by grounding LLM outputs in external knowledge sources. Conventional RAG pipelines use a retriever to fetch relevant document chunks from a database based on semantic similarity between vector embeddings (Karpukhin et al., 2020; Xiong et al., 2020). Improvements in retrieval have come from better embedding methods (Reimers and Gurevych, 2019; Gao et al., 2021), hybrid dense-sparse strategies, and hierarchical retrieval (Khattab and Zaharia, 2020; Zhang et al., 2022).

One particularly effective method for improving retrieval is Hypothetical Document Embeddings (HyDE) (Gao et al., 2023), where an LLM first generates a synthetic "hypothetical" answer to a query, embeds it, and then retrieves real documents most similar to that synthetic answer. This approach improves alignment between queries and relevant passages, especially in cases where the original query is underspecified or phrased differently than the source content.

Recent work in *Agentic RAG* (Schick et al., 2023; Qin et al., 2023; Yao et al., 2022; Liu et al., 2023) extends the static "retrieve-then-generate" pipeline

[*]These authors contributed equally to this work

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

into a dynamic decision-making process. Here, the LLM acts as an orchestrator, capable of decomposing complex queries, selecting appropriate tools or retrieval strategies, performing multi-hop searches, and verifying intermediate results before generating a final answer. Such systems have shown particular promise in domains requiring multi-step reasoning and evidence verification, making them well-suited for financial question answering, where queries may range from straightforward fact lookups to multi-document analyses (Wang et al., 2025).

Financial QA systems must process vast repositories of unstructured data, including annual reports, regulatory filings, earnings call transcripts, and market analyses (Wu et al., 2023; Li et al., 2023). The retrieval strategy must be both accurate and efficient, as inadequate retrieval can lead to irrelevant or misleading context being passed to the LLM. This is especially problematic for multi-hop queries, where context mismanagement or excessive token usage can degrade performance despite the availability of long-context models. Methods that involve processing information in the data stores into structures like graphs result in increased upfront token costs, albeit with better performance. To address these challenges in financial question answering, we present the following contributions:

- **Multi-HyDE**: A retrieval mechanism that utilizes multi-perspective hypothetical documents bringing an improvement in retrieval accuracy without an increase in token costs over HyDE (Gao et al., 2023)
- A combination of dense and sparse retrieval strategies to maintain performance on vector stores with over 500,000 tokens.
- An Agentic system that is capable of handling both straightforward queries and ones requiring planning, multi-hop retrieval, tool calling and verification.

Details of our system have been discussed in detail in Section 3. Details about the evaluation set up have been discussed in Section 4.

2 Related works

2.1 Retrieval Methods

The efficacy of Retrieval-Augmented Generation (RAG) systems fundamentally depends on the quality of their retrieval component (Lewis et al., 2020; Guu et al., 2020). Traditional RAG implementations employ semantic similarity search over vector databases, but this approach often suffers from a semantic mismatch between concise queries and the verbose, context-rich nature of source documents (LangChain, 2023). To address this, recent research has focused on enhancements in three main categories: pre-retrieval query transformations, hybrid retrieval strategies, and post-retrieval processing.

Pre-retrieval Query Transformation Pre-retrieval Query Transformation bridges the semantic gap through sophisticated query manipulation. A seminal advancement is Hypothetical Document Embeddings (HyDE), which uses a language model to generate a "pseudo-document" representing an ideal answer. The embedding of this richer document is then used for retrieval, shifting the paradigm from a query-to-document to a more effective answer-to-answer similarity search (Gao et al., 2023). Parallel to this, multi-query strategies improve recall by generating several variations of a user's query to capture different facets of the information need (LangChain, 2023). However, generating merely similar queries can sometimes degrade precision (Eibich et al., 2024). Recent advances include DMQR-RAG (Diverse Multi-Query Rewriting) (Li et al., 2024), which operates at different information granularity levels, and MUGI (Multi-Text Generation Integration) (Zhang et al., 2024), a training-free approach that generates multiple pseudo-references to enhance both sparse and dense retrieval. While these approaches improve retrieval, they fundamentally rely on query similarity rather than the complementary diversity we propose.

Hybrid Retrieval Strategies Hybrid Retrieval Strategies combine sparse and dense methods to leverage both keyword matching and semantic similarity. Dense retrieval excels at capturing semantic connections but can struggle with exact term matching, while sparse methods like BM25 provide precise keyword matching. In the context of large, structured financial reports, methods relying on vector similarity alone often fail to retrieve all relevant information and struggle to disambiguate semantically similar sections that differ only in critical numerical or temporal details. Our framework explicitly integrates Multi-HyDE with BM25 in a unified pipeline optimized for these documents, improving coverage and disambiguation.

Post-retrieval Processing Post-retrieval Processing has evolved beyond simple re-ranking to incorporate sophisticated correction mechanisms. For instance, CRAG introduces a retrieval evaluator that assesses document quality and triggers corrective actions, like web searches, when quality is insufficient (Yan et al., 2024). Self-RAG trains language models to adaptively retrieve passages and self-critique through generated reflection tokens (Asai et al., 2023). MAIN-RAG proposes a multi-agent filtering framework where agents collaboratively score retrieved documents (Chang et al., 2024). While promising, these systems introduce computational overhead to fix retrieval issues. Our approach therefore also emphasizes improving retrieval quality from the outset to reduce the need for extensive correction.

Our Multi-HyDE generates multiple non-equivalent but contextually related queries. Unlike methods that create semantically similar queries, our approach creates distinct but complementary information needs—for instance, generating separate queries about a company's fraud investigations and its criminal cases that might be answered within the same document context.

2.2 Agentic RAG

The static retrieve and generate workflow of traditional RAG is insufficient for complex queries that require multi-step reasoning and dynamic information gathering. This has spurred the development of Agentic RAG, which embeds autonomous agents into RAG pipelines to create dynamic problem-solving systems.

Finite State Machine Approaches Finite State Machine Approaches structure agentic workflows through formal state management. StateFlow models language model workflows as finite state machines, distinguishing between "process grounding" via states and "sub-task solving" through actions (Wu et al., 2024). This approach has achieved 13-28% higher success rates than ReAct on benchmarks while reducing costs by 3-5×. Our work extends this paradigm. In contrast to prior work applying state management primarily to retrieval and generation, we extend it to govern all tool calls issued by the language model, enabling coherent reasoning across multiple modalities.

Multi-Agent Architectures Multi-Agent Architectures coordinate specialized agents for complex tasks. MAIN-RAG exemplifies this with its multi-agent filtering system (Chang et al., 2024). However, such multi-agent systems can suffer from increased complexity and failure points.

2.3 RAG in Finance

Financial RAG systems face unique challenges due to complexity, precision, and regulation. These include handling 100+ page multi-year reports, disambiguating semantically similar sections, and managing numerical precision where subtle differences have significant implications. Specialized Financial Platforms have emerged to address these challenges.

Specialized Financial Platforms FinRobot provides a four-layer architecture with Financial AI Agents and Multi-source Foundation Models (Yang et al., 2024). While comprehensive, it lacks the specialized retrieval innovations for financial document disambiguation that our Multi-HyDE approach directly addresses.

FinSage focuses on regulatory compliance through a multi-aspect RAG framework, achieving 92.51% recall and a 24.06% accuracy improvement over baselines (Wang et al., 2025). However, FinSage relies on standard HyDE rather than our multi-perspective approach and uses curated questions instead of a comprehensive benchmark evaluation.

Financial Knowledge Graph Integration Financial Knowledge Graph Integration handles complex relationships through structured representations. While promising, knowledge graph approaches require significant upfront processing costs and may not adapt well to rapidly changing financial information. Our approach offers greater flexibility and lower preprocessing overhead while achieving comparable performance through retrieval optimization.

Evaluation Challenges Evaluation challenges in finance are complicated by the need for numerical precision. FinanceBench reveals that GPT-4-Turbo with retrieval systems incorrectly answers or refuses 81% of its questions (Islam et al., 2023). ConvFinQA highlights challenges in conversational queries requiring extensive calculations (Chen et al., 2022). These issues suggest that many existing systems may report inflated performance due to flawed evaluation methodologies. Our emphasis on human evaluation provides more accurate assessments for high-stakes appli-

cations. Our framework's modular design and reliability-focused architecture directly address enterprise deployment concerns often overlooked in academic research, demonstrating that retrieval optimization may provide greater returns than developing domain-specific language models alone.

In summary, existing RAG systems face key challenges including retrieval issues with semantic ambiguity in complex financial texts, limited capacity for multi-step reasoning and calculations, and inefficiencies due to complex architectures and flawed evaluations. Our framework addresses these by using Multi-HyDE with hybrid BM25 to improve retrieval accuracy and disambiguation, integrating an agentic tool usage system governed by unified state management for advanced reasoning, and reducing overhead by avoiding heavy knowledge graphs while relying on human evaluation for realistic performance assessment. This approach enhances retrieval reliability, reasoning capabilities, and system efficiency for financial RAG applications.

3 Methodology

To address the challenges outlined in Sections 1 and 2, we propose a retrieval-augmented generation (RAG) pipeline with the following key components:

- **Multi-HyDE:** A multi-hypothesis document expansion module that generates several hypothetical documents based on diverse variants of the input query. These documents are then used to retrieve semantically relevant content from the vector store.
- **Keyword-based Retrieval:** An auxiliary keyword-based retriever (e.g., BM25) designed to enhance retrieval performance for structured data such as tables, as well as for semantically similar documents (e.g., annual reports across different years).
- **Agentic Pipeline:** A multi-stage reasoning and retrieval process comprising:
  1. *Query Clarification:* The system first seeks to clarify the user's question, either through direct interaction with the user or by leveraging web search.
  2. *Initial Retrieval:* The clarified query is used to perform retrieval from the vector store using the components described above.
  3. *Iterative Refinement:* If the retrieved content is unsatisfactory, the system formulates a retrieval plan. This includes the ability to perform multi-hop retrievals, invoke external tools, and decompose the query into sub-queries.
  4. *Final Response:* Once the retrieved evidence is deemed sufficient, the system synthesizes and delivers the final answer to the user.

This integrated design allows the pipeline to combine the semantic strengths of vector-based retrieval with the precision of keyword-based methods, while also enabling dynamic reasoning for complex, multi-step information needs.

3.1 Multiple Hypothetical Dynamic Embeddings (Multi-HyDE)

For our main retrieval tool, we employ a combination of multi-query based retrieval (Eibich et al., 2024) and HyDE (Gao et al., 2023), which we call *Multi-HyDE*, along with BM25 based retrieval for tables and a re-ranker.

HyDE Gao et al. (2023) employ a generator g to create multiple hypothetical documents from a query q and retrieves real documents d_i from the dataset $\mathcal{D}$ that are similar to the hypothetical ones. N documents are sampled from g. An embedding model f is used to generate "hypothetical document embeddings" $\hat{v}$ for a query q as depicted in Equation 1.

$$\hat{v} = \frac{1}{N} \sum_{\hat{d}_i \sim g(q)} f(\hat{d}_i) \tag{1}$$

Multi-HyDE Multi-query approaches usually generate similar queries to the user's, but this has been shown to reduce retrieval precision (Eibich et al., 2024). Our approach instead uses an LLM g_q to generate queries $[q_1, q_2, ..., q_N]$ that may have answers present in the same context, following which it generates a hypothetical document for each query. These queries may take the form of similar queries, related queries with distinct meanings (such as including a query on fraud by a company A and a query on criminal cases by company A) or it may result in query decomposition. To the best of our knowledge, this particular approach has not been tried before. An embedding model f is used to generate "hypothetical document embeddings"

$\hat{v}_i \in \mathbb{R}^{\hat{d}_{embed}}$, as depicted in Algorithm 1. Our retriever h retrieves k_1 documents from $\mathcal{D}$, and we further use a reranker to select the top k_2 documents.

Algorithm 1 Multi-HyDE Retrieval

Require: query q, database $\mathcal{D}$, query and document generators g_q, g, embedding model f, retriever h, reranker r, hyperparameters N, k_1, k_2

$[q_1, \ldots, q_N] \leftarrow g_q(q)$
for each q_i in $[q_1, \ldots, q_N]$ **do**
    $\hat{v}_i \leftarrow f(g(q_i))$
    $S_i \leftarrow h(\hat{v}_i)$
end for
$d_{total} \leftarrow concat(S_1, S_2, ..., S_N)$
$d_{final} \leftarrow r(d_{total})$
return d_{final}

3.2 Agentic RAG

To address both simple and complex multi-hop queries, we employ an agentic system (Figure 1) equipped with several tools, including `edgar_tool`, `Alpha Vantage Exchange Rate`, `web_search`, and a Python calculator, as well as a retriever based on Multi-HyDE. Additional tools are listed in Appendix D.

The query processing begins with direct retrieval using Multi-HyDE, ensuring the system remains grounded in explicitly-included sources. Retrieved documents are then passed to the LLM Agent for reasoning and synthesis. If these documents are insufficient to fully answer the query, the LLM dynamically invokes available tools.

For improved performance, the LLM not only generates tool calls but also produces intermediate reasoning steps, user-facing responses, decomposed sub-queries, and a structured execution plan, inspired by Hao et al. (2023); Radhakrishnan et al. (2023); Zhou et al. (2023); Wang et al. (2023); Girhepuje et al. (2024). The full prompt is given in Appendix B. Queries are broken down into atomic steps, with each step resolved using the most suitable tool from the current toolset. The LLM evaluates intermediate results at each stage, adapting the plan when necessary to ensure accuracy and grounding.

This design supports highly dynamic workflows: tools can be added or removed on demand, enabling integration of custom data sources, access to live information, and execution of complex sequential reasoning processes. While standard RAG also grounds responses in retrieved documents, it typically relies on a single retrieval step, leaving the model prone to filling gaps with its latent knowledge if the evidence is incomplete. In contrast, Agentic RAG decomposes queries into atomic steps, validates intermediate results, and dynamically invokes additional tools or retrievals as needed. This iterative, evidence-driven process strengthens fidelity to verifiable sources, reduces hallucination, and produces more reliable answers across diverse and complex query types.

Algorithm 2 Agentic RAG System

Require: query q, database $\mathcal{D}$, set of tools T, LLM agent A

function PROCESS_QUERY($q, \mathcal{D}, T, A$)
    $d_{initial} \leftarrow$ Multi-HyDE($q, \mathcal{D}$)
    LLM Agent history $H \leftarrow [q, d_{initial}]$
    loop
        A analyzes H to determine if the query can be answered
        if A determines an answer exists **then**
            Generate final answer from H
            return Final answer
        else
            A generates a sub-query q_{sub} and selects a tool $t \in T$
            $tool\_output \leftarrow t(q_{sub})$
            $H \leftarrow$ concat$(H, tool_output)$ ▷ Add tool's output to the LLM's history
        end if
    end loop
end function

4 Experimental setup

We ran our experiments using subsets of datasets (selection of subset is described in Appendix E) due to limited resources. We employ GPT-4o mini and the Mini-LM reranker for running the pipeline. Additional implementation details are included in Appendix G.

4.1 Evaluation datasets

We use a subset of questions from the FinanceBench (Islam et al., 2023) and ConvFinQA (Chen et al., 2022) datasets. From FinanceBench, we have selected from 150 human-annotated examples provided. These examples include evidence designated as ground truth context, with additional

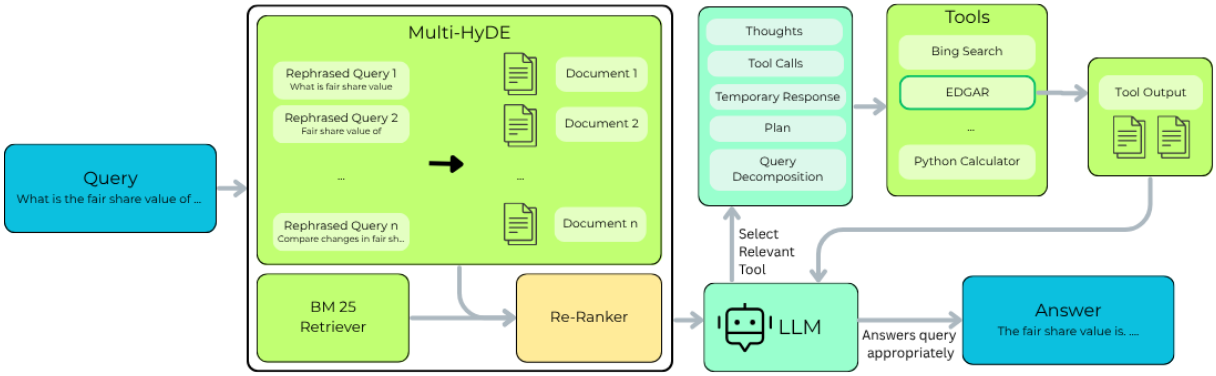

Figure 1: Our proposed agentic framework for financial question-answering.

justification considered when necessary. Appendix E provides details on the subsets used. **Furthermore, we ensure that the entire PDF document is added to the vector store in contrast to the ConvFinQA and FinanceBench datasets, which only pass the evidence pages.** We believe this better simulates real-world scenarios. This restricts us from comparing other retrieval methods where only evidence pages are passed as context. We also include a subset of filings and questions from financial-qa-10K[1] for a comparative study. An example of a question-answer pair is provided in Appendix A.

4.2 Metrics

We evaluate experiments and optimizations using **ROUGE,**[2] **Cosine Similarity**, as well as metrics from RAGAS (Es et al., 2023) and human evaluation. We use RAGAS with GPT-4o mini to calculate **Factual Correctness** (similar to the F1 Score) and **Faithfulness**. During human evaluation, accuracy and reliability are measured. The metrics are defined in Appendix F.1.

5 Experimental results and analyses

Method	Accuracy (%)	Reliability (%)
Multi-HyDE	34.4	37.91
Final Pipeline	**45.6**	**52.91**

Table 1: Human evaluation on subset of ConvFinQA and FinanceBench.

[1] https://huggingface.co/datasets/virattt/financial-qa-10K

[2] Low ROUGE scores in some experiments are attributed to the fact that the ground truth answers in the dataset consisted of only a single number, whereas large models explained their approaches.

Performance against other methods We provide a comparison of our pipeline against a representative method for retrieval optimization (HyDE), graph based knowledge organization (LightRAG) (Guo et al., 2024) and post-retrieval corrective measures (CRAG) (Yan et al., 2024). We include scores for Multi-HyDE with access to tools against these baselines.

Our results show improvement across all measures except Cosine Similarity: we achieve significant improvements in Recall, Facutal accuracy and Faithfulness (See table 2, 3) while having the same token costs involved as HyDE (since both generate the same number of hypothetical documents for a given user query) and avoid the upfront costs associated with graph based methods to create the graph.

Our approach supports dynamic vector stores - documents can be added or removed from the vector store without incurring additional costs, graph based approaches where removing information from the graph would incur some costs.

The results show the advantages of Multi-HyDE in the financial domain. We attribute the improved performance to the fact that financial reports across multiple years could have semantically similar content - pairing a dense retrieval method that identifies relevant information from an increased variety of potential sources and a sparse keyword based retriever to identify structured information improves overall performance by being able to handle more cases than any individual method.

Reliability Considerations: In verifying the LLM as a judge procedure utilized by RAGAS, we observe that in numerical examples, the LLM judge might provide incorrect evaluations (see Appendix F.2). Further, cases where the wrong answer is provided confidently has greater chance of ad-

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score*
Multi-HyDE	0.6269	**0.3547**	**0.3849**	**0.8404**	**0.0594**
HyDE	0.7660	0.1154	0.2890	0.8290	0.0498
CRAG	0.7939	0.1556	0.0855	0.2521	0.0443
LightRAG	**0.7999**	0.0000	0.2434	0.4629	0.1632

Table 2: Evaluation Metrics for Different Methods on subset of **ConvFinQA + FinanceBench**.

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score
Multi-HyDE	0.8976	0.8170	0.5205	**0.9352**	0.**4871**
HyDE	0.8883	0.6885	**0.5585**	0.8463	0.3726
CRAG	**0.9347**	**0.8500**	0.4708	0.7774	0.4290
LightRAG	0.7308	0.0000	0.0368	0.4629	0.3412

Table 3: Evaluation Metrics for Different Methods on subset on questions from financial-qa-10K .

verse impact that the system admitting to not having the exact answer. To confirm the performance of our proposed pipeline in light of the above challenges, we conduct a human evaluation of the responses with metrics reliabilty (fraction of confidently given answers which are correct rather than hallucinations) and accuracy (fraction of correct answers). Detailed definitions are provided in F.1.

Ablation study: In Table 4, we show that Multi-HyDE outperforms regular HyDE. We also perform a comparison between 2 rerankers `ms-marco-MiniLM-L-6-v2` (Cross Encoder) and `bge-reranker-v2-m3` (BGE) from huggingface. Though BGE is more performant, it is significantly more resource-intensive and slower. We also show that hybrid retrieval with BM25 clearly outperforms dense retrieval methods for long-document financial data. Tool calling does not improve accuracy, however it provides resiliency when some types of relevant data are not provided.

6 Future work

Agents and fine-tuning Small Language Models finetuned using parameter efficient techniques like LoRA(Hu et al., 2021) to be used as individual agents instead of relying on large closed source models, especially for tasks like query re-writing or hypothetical document generation, particularly to suit the language and format used in financial reports.

Better metrics for financial RAG Currently, LLM-based evaluation often incorrectly evaluates responses, especially when an answer is primarily numeric. Different evaluation systems may help improve this. In addition, a more comprehensive evaluation on complete datasets could be undertaken given more resources.

7 Conclusion

This research presents a novel approach to financial question answering, addressing key challenges in hallucination reduction and accurate information retrieval from complex financial documents. Our framework introduces Multi-HyDE, an extension of Hypothetical Document Embeddings that leverages multiple non-equivalent queries to enhance retrieval effectiveness. When combined with BM25 for tables and appropriate rerankers, Multi-HyDE demonstrates superior performance in capturing relevant information from financial corpora. Additionally, we developed and evaluated an agentic pipeline offering improved performance, capable of handling both simple queries, and ones requiring complex multi-hop retrieval and reasoning.

Our evaluation highlights the importance of specialized retrieval techniques for domain-specific applications and underscores the limitations of current LLM-based assessment metrics in financial contexts. Human evaluation proved crucial for accurately measuring performance, revealing substantial improvements with our ensembled approach. The modular design of our framework facilitates adaptation to other domains requiring precise information extraction. By addressing fundamental challenges in financial RAG systems, our work contributes to building more trustworthy AI systems for high-stakes applications where factual accuracy

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score
1	0.8883	0.6885	0.5585	0.8463	0.3726
2	0.8932	0.7464	0.5539	0.8837	0.3575
3	0.8935	**0.8484**	**0.5868**	0.8768	0.3996
4	0.8976	0.8170	0.5205	**0.9352**	0.**4871**
5	**0.9119**	0.8033	0.5172	0.8298	0.4628
6	0.8935	**0.8484**	0.5867	0.8767	0.3996

Table 4: Effect of BM25, rerankers and tools on recall. (with financial-qa 10k dataset)

1. HyDE
2. Multi-HyDE + Cross Encoder Reranker
3. Multi-HyDE + BM25 + Cross Encoder Reranker
4. Multi-HyDE + BM25 + BGE Reranker
5. Multi-HyDE + BM25 + BGE Reranker without tools
6. Multi-HyDE + BM25 + Cross Encoder Reranker without tools

is paramount. Future research directions include fine-tuning models for financial contexts and developing more nuanced evaluation metrics.

Limitations

Due to resource constraints, our evaluation is conducted on a relatively small dataset, which may limit the generalizability of the results.

Although our approach demonstrates improvements over existing baselines, its practical deployment is still challenged by the presence of hallucinations in more complex and ambiguous datasets. Consequently, the system currently requires human oversight and verification to ensure reliability and factual consistency.

Acknowledgements

The authors thank InterIIT Tech Meet 13.0 and Pathway for proposing the problem statement and facilitating access to task materials and clarifications during the competition.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *Preprint*, arXiv:2310.11511.

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and Na Zou. 2024. MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation. *Preprint*, arXiv:2501.00332.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. *Preprint*, arXiv:2210.03849.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Matouš Eibich, Shivay Nagpal, and Alexander Fred-Ojala. 2024. ARAGOG: Advanced RAG Output Grading. *Preprint*, arXiv:2404.01037.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated Evaluation of Retrieval Augmented Generation. *Preprint*, arXiv:2309.15217.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Sahil Girhepuje, Siva Sankar Sajeev, Purvam Jain, Arya Sikder, Adithya Rama Varma, Ryan George, Akshay Govind Srinivasan, Mahendra Kurup, Ashmit Sinha, and Sudip Mondal. 2024. RE-GAINS & EnChAnT: Intelligent Tool Manipulation Systems For Enhanced Query Responses. *Preprint*, arXiv:2401.15724.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *Preprint*, arXiv:2410.05779.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint*. ArXiv:2002.08909 [cs].

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. *Preprint*, arXiv:2305.14992.

Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Geneviève Fried, Ryan Lowe, and Joelle Pineau. 2023. Foundation models for legal reasoning. *arXiv preprint arXiv:2307.03557*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Yue Huang, Xiaohan Sun, Yao Xiong, Zhicheng Dou, Guoliang Zhang, and Jian Yuan. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. FinanceBench: A New Benchmark for Financial Question Answering. *Preprint*, arXiv:2311.11944.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

LangChain. 2023. Query Transformations.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zhicong Li, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazhen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. 2024. Dmqr-rag: Diverse multi-query rewriting for rag. *Preprint*, arXiv:2411.13154.

Zhuangzhuang Li, Hanyi Wang, Zhengqing Chen, and Xia Chen. 2023. Finbert: A pre-trained financial language representation model for financial text mining. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.

Yujia Liu, Yuwei Xie, Chunyuan Chen, Sylvia Wang, Yuxin Yuan, Yang Liu, Xiang Hu, Songyang Wang, Tianyu Qiao, Lingyu Pan, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yujia Qin, Shengding Deng, Furui Xu, Shiwei Chen, Yankai Lin, Weilin Sun, Meng Bu, Peng Li, Shulin Zhou, Chao Yang, and 1 others. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, and 5 others. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *Preprint*, arXiv:2307.11768.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Venkataraman, Gabriel Maginnis, Arun Nori, and 1 others. 2023. Large language models in medicine. *Nature Medicine*, 29(8):1998–2012.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix,

Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *Preprint*, arXiv:2305.04091.

Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Muzhi Li, Zhuhong Li, Hailin He, Yuchen Hua, Peng Lu, Suyuchen Wang, Yihong Wu, Jerry Huang, Jingrui Tian, Fengran Mo, Yufei Cui, and Ling Zhou. 2025. Finsage: A multi-aspect rag system for financial filings question answering. *Preprint*, arXiv:2504.14493.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Yiran Wu, Tianwei Yue, Shaokun Zhang, Chi Wang, and Qingyun Wu. 2024. StateFlow: Enhancing LLM Task-Solving through State-Driven Workflows. *Preprint*, arXiv:2403.11322.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective Retrieval Augmented Generation. *Preprint*, arXiv:2401.15884.

Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, and Christina Dan Wang. 2024. FinRobot: An Open-Source AI Agent Platform for Financial Applications using Large Language Models. *Preprint*, arXiv:2405.14767.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Barret Zhang, Jonathon Shlens, and Jeff Dean. 2022. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the best practices of query expansion with large language models. *Preprint*, arXiv:2401.06311.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

A Question-answer example


```
{
    "question": "For American Water
    ↪  Works, what was the rate of
    ↪  growth from 2013 to 2014 in the
    ↪  fair value per share"
    "answer" : "",
    "context": "~13.3%. Page 81,
    ↪  Table[The
    weighted-average assumptions used in
    ↪  the
    Monte Carlo simulation and the
    ↪  weighted-average
    grant date fair values of RSUs
    ↪  granted
    for the years ended December 31]

    [45.45 -40.13]/40.13 = 13.3%

    AMERICAN WATER WORKS COMPANY, INC.",
    "ticker": "AWK",
    "filed on": "31 December 2015"
}
```


B Meta-Plan JSON Instructions

```
{
    "thought": "...",  # Thought process
    ↪  and reasoning of the bot for the
    ↪  current step
    "tool_calls": [{"name": "...",
    ↪  "args": {...}},
    {"name": "...", "args": {...}},
    ↪  ...],  # List of tools to be
    ↪  called along with the
    ↪  appropriate arguments.
    "audio": "...", # Respond
    ↪  comprehensively to the query in
    ↪  a verbose way and output in
    ↪  formatted markdown string
    "plan": "...",
    # The overall plan for calling
    ↪  various tools and answering the
    ↪  query. This needs to be updated
    ↪  dynamically based on the
    ↪  retrieved information from tool
    ↪  calls.
    "queries":
    ↪  [{"query":"...","answer":"..."},
    {"query":"...","answer":"..."}]
```

```
    # The decomposed sub-queries.
    ↪  Intially all the answers are
    ↪  empty strings, as answers from
    ↪  tool calls come in, update them
    ↪  accordingly
}
```

C Retrieval Challenges

In the financial domain, retrieval methods that rely solely on vector similarity often fail to distinguish between passages that are semantically similar but differ in critical numerical details or temporal references. These distinctions, although subtle, are essential for producing accurate and trustworthy responses when answering questions about structured financial reports.

Consider the following example from our evaluation set:

```
"query": "For American Water Works, what
↪  was the growth in allowance for
↪  other funds used during construction
↪  from 2013 to 2014?"

"retrieved_reference_1": "In 2014, we
↪  spent $3.6 million, including $0.8
↪  million funded by research grants...
↪  (discussion on research and
↪  development spending)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_2": "Cash flows
↪  used in investing activities
↪  increased in 2014 compared to 2013
↪  primarily due to an increase in our
↪  capital expenditures... (details on
↪  capital expenditures)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_3": "Amortization
↪  of contributions in aid of
↪  construction was $23,913, $22,363,
↪  and $20,979 for the years ended
↪  December 31, 2014, 2013, and 2012...
↪  (amortization details)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_4": "Such grants
↪  reduce the cost of research and
↪  allow collaboration with leading
↪  national and international
↪  researchers... (discussion on
↪  research grants and collaboration)
↪  [awk_2017_10K.pdf]"

"retrieved_reference_5": "Amortization
↪  of contributions in aid of
↪  construction was $27, $26, and $24
↪  for the years ended December 31,
↪  2016, 2015, and 2014... (further
↪  amortization details)
↪  [awk_2017_10K.pdf]"
```

As shown above, SEC filings from different years (e.g., 2015 vs. 2017) often include passages with similar or even nearly identical phrasing. However, for financial question-answering, distinctions such as the reporting year or specific numerical values are vital for correctness. Standard dense retrieval models tend to conflate these passages due to their semantic resemblance, leading to unreliable results.

To mitigate this issue, we incorporate BM25 alongside dense vector retrieval. This hybrid approach ensures that keyword and phrase-level matches (e.g., exact years, financial figures, or domain-specific terminology) complement semantic similarity, resulting in more precise and contextually appropriate retrievals.

D Tools

Our agentic pipeline has various tools to fetch data from various data sources apart from the retrieved context. The tools are divided into different types based on their use cases give below. Having more than one tools provide redundancy in case one or more tools fail.

1. **Web-search:** The web search tool provides real-time access to the web search queries providing access to news, web pages, and more which might not be there in the retrieved context. **SERP API**, **Bing Web Search**, and **DuckDuckGo Web Search** are the tools used by the agent to obtain the data from a web search.

2. **Financial Data API:** This is a collection of tools that provide real-time as well as histori-

cal data about the prices of stocks, securities, and cryptocurrencies. **Yahoo Finance**, **Alpha Vantage**, **EDGAR Tool**(Electronic Data Gathering, Analysis, and Retrieval system) and **Financial Modelling Prep** tool providing real time financial data from various exchanges.

3. **Mathematical tools:** The **WolframAlpha API** and **Python Calculator** are the tools incorporated to provide the the data processing ability to the agent. WolframAlpha takes in the mathematical questions in natural language and provides us with the answer whereas python calculator can be used to help the agent with more menial calculations.

E Dataset

Owing to the inconsistent evaluation results often observed in LLM-based methods and limited computational resources, we conduct our experiments on a focused subset of the FinanceBench and ConvFinQA datasets. Specifically, we select reports with the highest density of associated questions to ensure the relevance and informativeness of our evaluation. The selected subset comprises SEC 10-K filings from the following companies:

- **American Water Works**: 2015, 2017, 2018
- **AMD**: 2022
- **American Express**: 2022
- **Boeing**: 2022

F Evaluation

F.1 Definitions of metrics

RAGAS defines metrics by comparing the facts in a model's answer to those in the retrieved context or ground truth. The *Faithfulness Score* is RAG-specific, measuring the proportion of claims in the answer that are supported by the retrieved context. *Factual Correctness*, based on the F1 score, can be applied to any model.

Faithfulness:

$$\text{Faithfulness} = \frac{\text{Supported claims}}{\text{Total claims in answer}} \tag{2}$$

Factual Correctness: Let

$$\text{TP} = \#\{\text{claims in answer present in reference}\}$$
$$\text{FP} = \#\{\text{claims in answer not in reference}\}$$
$$\text{FN} = \#\{\text{claims in reference not in answer}\}$$

Then

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

$$\text{F1} = \frac{2 \circ \text{Precision} \circ \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

For human evalution, we define accuracy (A) as the fraction of correct answers[3] and reliability (R) as the fraction of answers where the LLM was confident about its answer being correct. This is related to the hallucination rate (the fraction of incorrect answers, denoted HR) as

$$HR = \frac{A}{R} - A, \text{where} \tag{6}$$

$U =$ Number of times LLM refused to answer

$N =$ total number of answers

In Table 1, we rely on human evaluation to show that the Agentic pipeline performs better than Multi-HyDE with tools.

F.2 Examples in which RAGAS fails

In addition to errors in the LLM evaluation used by RAGAS, we note that the evaluation often fails when more precise numbers are given in the generated answer than available in the ground truth. This is partly due to ambiguities in the dataset, but also because, for instance, it often penalizes the model for minor inaccuracies or rounding, and at other times it gives a high score despite an outright refusal to answer.

Listing 1: RAGAS incorrectly gives a factual correctness of 0 though the response is simply more precise than the reference.

```
Question: For American Water
    Works, what was the rate of
    growth from 2013 to 2014 in
    the fair value per share
Response: ```json
{
  "tool_calls": []
}
```
```

[3] In cases when a question requires multiple independent answers, we assign the score as the fraction of correct answers for that question
```

```
Based on the provided text, the
    fair value per share for
    American Water Works was $37
    .40 in 2012, $40.13 in 2013,
    and $45.45 in 2014 (Document
    8). Therefore, the growth
    rate from 2013 to 2014 is:

($45.45 - $40.13) / $40.13 =
    0.1325 or 13.25%



Reference: 13.30%
Semantic Similarity:
    0.5918684811439618
Factual Correctness: 0.0
Faithfulness: 0.6666666666666666
```

Listing 2: RAGAS incorrectly scores the LLM's refusal to answer very highly

```
Question: For American Water
    Works, what was the rate of
    growth from 2013 to 2014 in
    the fair value per share
Response: The provided text does
    not contain information on the
     fair value per share of
    American Water Works for 2013
    and 2014, nor does it provide
    the means to calculate a
    growth rate. Therefore, I
    cannot answer your question
    using the given documents. To
     answer this question,
    additional information is
    needed. I would need to
    access financial statements or
     other reliable sources
    containing this data.

Reference: 13.30%
Semantic Similarity:
    0.5736175582900219
Factual Correctness: 0.91
Faithfulness: 1.0
```

G Additional Implementation Details

PDF Parsing We use a customised PDF parsing system (Fig. 2) using the **Docling** library to extract and structure data from complex documents. It handles text, tables, and images, exporting tables in HTML format. Further, it utilizes recursive chunk splitting as the text chunking strategy for context preservation.

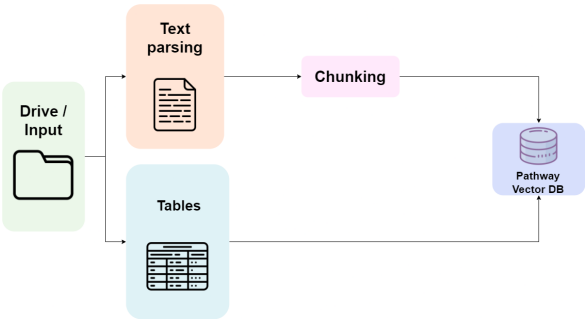

Figure 2: PDF-Parser System: Documents can be added to a Google Drive, which dynamically updates the Vector Store. Text is parsed using our parsing pipeline, chunked recursively before populating the Vector Store.

Retrieval We employ **HNSW** for indexing and in addition use **BM25** for retrieving tables. Our Vector Store is implemented using Pathway [4]. Row and column aggregation is also performed on tables. Keeping modularity in focus, retrieval methods are represented as tools, alongside others like web search and calculator. The **Multi-HyDE** retriever, selects the top $K_1 = 10$ chunks, while the BM25 retriever fetches the top $K_2 = 15$ chunks.

Reranking A re-ranker[5] is employed to pick the top $K = 8$ relevant chunks. This was determined after evaluating performance on various values of K, as shown in Table 5.

Top K Value	Accuracy (%)
1	57.5
2	75.3
8	79.6
10	**80.1**

Table 5: Accuracy for different values of 10 K retrieved documents.

[4] https://github.com/pathwaycom/pathway

[5] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2 or https://huggingface.co/yxzwayne/bge-reranker-v2-m3, specified in our experiments

H Other Ablations

Tables below depict other ablations performed as a part of our experimentation and analysis.

Method	Precision	Recall	Accuracy
Naive-RAG	0.912	0.592	0.616
HyDE	TBD	TBD	TBD
Multi-HyDE	0.932	0.625	0.721

Table 6: Comparison of Naive-RAG, HyDE, and Multi-HyDE on a subset of financial-qa-10K dataset

Method	In-Tokens	Out-Tokens	Time Spent
Naive-RAG	-	-	0.199s
HyDE	133.5	428.2	9.344s
Multi-HyDE	193.6	421.4	9.121s

Table 7: Resource usage comparison on Financial-qa-10K Dataset.

Metric	Method 1	Method 2
Cosine Similarity	**0.5981**	0.5765
Recall	0.2462	**0.2910**
Factual Correctness	0.1738	**0.2291**
Faithfulness	0.8103	**0.8754**
ROUGE score	0.0346	**0.0349**

Table 8: Evaluation metrics comparing different parsers, showing the improvement of Docling over Open Parse (with ConvFinQA and FinBench dataset subsets)

Method 1: Multi-HyDE + Re-ranker + Open Parse (Llama-70b)
Method 2: Multi-HyDE + Re-ranker + Docling Parser (Llama-70b)