iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** II **Month of publication:** February 2026

DOI: https://doi.org/10.22214/ijraset.2026.77501

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue II Feb 2026- Available at www.ijraset.com

Enhancing RAG with Agentic AI and Multi-HyDE for Knowledge Retrieval and Hallucination Reduction using Mistral and Qwen2 LLMs

Sumedha Arya

Abstract: ***Financial data is very sensitive and keeps on changing. This makes it important for accurate and reliable information retrieval in financial question-answering systems. These systems must provide precise answers using correct and up-to-date information from sources like financial news, reports, and regulatory filings. Traditional retrieval systems usually search using only one query and one database. This approach is inefficient for finance, because financial documents are complex, detailed, and spread across many years. A single query may miss important information or retrieve incomplete results, results in hallucination. To solve this problem, we used a framework that considers a financial Retrieval Augmented Generation (RAG) system with agentic AI and the Multi-HyDE method. The framework is tested using two different LLMs – MistralAI and Qwen2. Both the models demonstrated the faithfulness score of 1.0 and represents the ability of framework to reduce hallucinations in LLMs.***

Keywords: Hallucination, RAG, AI Agents, Multi-HyDE.

I. INTRODUCTION

Large Language Models (LLMs) such as OpenAI GPT-4 [26], Meta LLaMA [32], and Google PaLM [5] have greatly improved the applications of natural language processing (NLP). These models can understand the context better, with good reasoning ability about for questions, and can generate human-like answers even with few-shot learning. Therefore, they are now used in important areas such as Medical and healthcare [31], law and legal document analysis [14], and financial services [35; 24]. However, LLMs do suffer from hallucination. They may generate incorrect or wrong information [18; 16]. This is especially dangerous in finance, because wrong information can lead to financial losses, damage to reputation, or regulatory problems.

To solve this issue, authors in their research introduced Retrieval-Augmented Generation (RAG) [22; 12]. In RAG, the model first retrieves relevant documents from an external database and then generates answers based on those real documents making the answers more accurate and reliable. Retrieval can be improved using better embeddings [29; 9], hybrid retrieval combining keyword and semantic search [20; 41], and advanced methods such as Hypothetical Document Embeddings (HyDE) [8]. In HyDE, the LLM first generates a possible answer, converts it into an embedding, and then retrieves real documents similar to that generated answer, improving retrieval accuracy.

More recently, Agentic RAG systems have been developed [30; 27; 40; 25]. In these systems, the LLM acts like an intelligent agent or brain that can break complex questions into smaller parts, retrieve information step-by-step, use tools, and verify results before generating the final answer. This approach is very useful in many applications including finance, where questions may require analyzing multiple reports, earnings statements, and financial news [34].

Financial question-answering systems must handle large amounts of complex data such as annual reports, regulatory filings, earnings call transcripts, and market analyses [35; 24]. To improve retrieval accuracy and efficiency, we used Multi-HyDE, that generates multiple hypothetical queries instead of just one, improving retrieval performance without increasing computational cost. It also combines dense and sparse retrieval methods and uses an agentic reasoning system to handle both simple and complex financial queries more accurately and reliably. Using this concept, we used a framework that integrates a financial RAG system with agentic AI and the Multi-HyDE method. The results obtained by testing the framework on two different LLMs – MistralAI and Qwen2, demonstrated the faithfulness score of 1.0. This shows the ability of framework to reduce hallucinations in LLMs.

The paper is further divided into following sections; review of the literature review, research methodology, results analysis, conclusion and references.

II. LITERATURE REVIEW

The performance of Retrieval-Augmented Generation (RAG) systems mainly depends on how well they retrieve relevant documents [22; 12] from the vector database. Early RAG systems used semantic similarity search which are not much efficient in knowledge retrieval due to semantic mismatch. User queries are usually short and simple, while source documents are long and detailed. Because of this mismatch, the system may fail to retrieve the most relevant information [21]. To improve retrieval quality, researchers have focused on three major areas such as pre-retrieval query transformation, hybrid retrieval strategies, and post-retrieval processing. In Pre-retrieval Query Transformation, the query is improved before searching. Based on it, an important technique called Hypothetical Document Embeddings (HyDE) is introduced. Instead of directly using the user query, the system first generates a hypothetical answer using a language model. The embedding of this generated document is then used for retrieval. This changes the process from query-to-document matching to answer-to-answer similarity matching, which improves retrieval performance [8]. Another approach is multi-query generation, where several variations of the user's query are generated to capture different aspects of the information need [21]. While this improves recall, it may sometimes reduce precision if the generated queries are too similar [6]. Recent developments include:

- DMQR-RAG, which generates diverse queries at different information levels [23].
- MUGI, a training-free method that produces multiple pseudo-references to improve both sparse and dense retrieval [42].

However, most of these approaches still generate semantically similar queries. Hybrid retrieval strategies are the combination of both dense and sparse techniques. In dense retrieval, it performs semantic similarity using embeddings while in sparse retrieval, keyword-based matching is performed such as BM25. Dense retrieval captures semantic meaning but may miss exact keywords. Sparse retrieval ensures precise keyword matching but does not fully understand semantic relationships. In financial documents, which are long and structured, they not only rely on vector similarity which may miss important numerical or time-based differences. Hybrid systems are much wider in nature, which improves better coverage and disambiguation. After retrieving documents, some systems apply correction mechanisms. They are called as Post-retrieval Processing. Some of the work based on it are:

- CRAG evaluates the quality of retrieved documents and triggers corrective actions, such as web searches, if needed [38].
- Self-RAG trains models to retrieve information adaptively and self-critique their responses [2].
- MAIN-RAG introduces a multi-agent filtering framework where agents collaboratively score retrieved documents [3].

Although these systems improve reliability, they add computational complexity. Traditional RAG was simple in nature. They follow an easy retrieve-and-generate pipeline. However, complex queries often require multi-step reasoning and dynamic information gathering. This highlights the limitation of traditional RAG. Therefore, to overcome it, the Agentic RAG is developed, where autonomous agents manage reasoning and tool usage. Some systems perform reasoning using finite state machines. For example, models such as StateFlow does workflows using defined states and actions, thereby separating process based on grounding from sub-task solving [36]. This approach has improved the success rates by 13 to 28% as compared to Reasoning and Action and reduced the cost by 3 to 5 percent. Multi-agent systems comprise of various agents working in an environment where they have been assigned with different specialized roles. For example, MAIN-RAG uses multiple agents to filter and score retrieved documents [3]. However, this approach is effective in nature, but multi-agent systems increase system complexity and may introduce failures at more than one point. Financial RAG systems face unique challenges. Some of them are:

- Handling long, multi-year reports (often 100+ pages)
- Disambiguating similar-looking sections
- Managing numerical precision
- Meeting regulatory requirements

Even small numerical errors can have serious consequences. Several domain-specific systems have been proposed, that are specialized financial platforms. Some of them are:

- FinRobot, which uses Financial AI Agents and multi-source foundation models in a four-layer architecture [39]. However, it does not focus specifically on retrieval disambiguation in financial documents.
- FinSage, emphasizes on regulatory compliance using a multi-aspect RAG framework achieving and overall recall of 92.51% and improves accuracy by 24.06% over baselines [34]. However, it relies on standard HyDE rather than a multi-perspective approach. In addition, it uses curated datasets instead of comprehensive benchmark evaluation.

Some approaches use financial knowledge graphs to represent structured relationships. However, this approach is promising in nature, but they require heavy preprocessing and may not adapt well to rapidly changing financial information.

Evaluation in financial RAG is particularly difficult due to the need for high numerical precision. Some of the work based on evaluation in financial RAG are:

- FinanceBench shows that GPT-4-Turbo with retrieval incorrectly answers or refuses 81% of questions [17].
- ConvFinQA highlights challenges in conversational financial reasoning requiring complex calculations [4].

These findings suggest that some systems may overestimate performance due to weak evaluation strategies.

III.RESEARCH METHODOLOGY

In this section, we detailed about the methodology used in our research explaining, how a RAG system can be built and tested for answering financial news questions. The goal is to make such system answer questions accurately, using real news data instead of guessing. For this purpose, we used two different large language models, such as: Mistral-7B-Instruct-v0.3 and Qwen2-7B-Instruct. Both models followed the same overall framework but had slight differences in prompt design and optimization.

The first phase focused on data collection and preparation. The dataset is sourced from Kaggle, containing around 49,637 financial news entries from 2003 to 2020. Each record had a date and combined news text representing headlines and snippets. In data preprocessing, unnecessary columns were removed, and the news text was converted into a clean list of documents. No synthetic data was added, ensuring that the system only relied on real financial news, maintaining authenticity and reliability.

Next, the environment and models were set up. Important libraries for embeddings, vector search, and language model handling were installed. GPU acceleration was used for faster processing, and memory-saving techniques like float16 precision were applied. The two LLMs, Mistral and Qwen were loaded using Hugging Face Transformers. A text-generation pipeline was created with low temperature values to ensure deterministic or less random outputs. For document embeddings, the model all-MiniLM-L6-v2 was used to convert text into 384-dimensional vectors, enabling semantic similarity search.

The third phase involved hybrid indexing for retrieval. Two types of search methods were combined. First, dense retrieval was implemented using FAISS, where all documents were converted into embeddings and stored in a similarity index. Second, sparse retrieval was implemented using BM25, which focuses on keyword matching. Dense retrieval captures meaning, while sparse retrieval captures exact keywords. Combining both improves the chances of retrieving the most relevant financial news, especially when questions include specific financial terms like “profit before tax.”

To further improve retrieval quality, a Multi-HyDE, Hypothetical Document Embeddings, mechanism was introduced. Instead of searching using only the original query, the system first generated multiple rephrased versions of the query to add diversity in it. Then, for each version, it generated a hypothetical answer paragraph using the LLM. These hypothetical answers were embedded and used to search for similar real documents from the vector database. The retrieved results from both dense and sparse search were combined and ranked. This method reduces mismatch between the user’s request and the document content, which is especially helpful for time-sensitive financial queries.

After retrieval, the system uses an agentic RAG process, which acts like an intelligent agent with tools. Two tools were defined: one for retrieving documents and another for performing simple calculations. The agent followed a step-by-step reasoning loop: it could think, call a tool, observe results, and then produce a final answer. This structured reasoning reduces hallucination because the model must rely on retrieved evidence before answering. Slight differences in prompt instructions were used between Mistral and Qwen to control verbosity and structure.

For evaluation, a faithfulness check of the system was implemented to ensure its factual claims based on the generated answer. Then, each claim was checked to ensure the correctness of the retrieved documents. A score was calculated based on how many claims were supported. For example, if all claims were supported, the faithfulness score was 1.000. This ensures that answers are grounded in actual news data rather than fabricated information.

The proposed algorithm for the financial RAG system is given as:

A. Algorithm Financial_RAG_System(Query, Dataset_Path, Model_Name, Max_Steps=6, N_Variants=4, K1=10, K2=6)

1) Input

- Query: String (user question, e.g., financial news query)
- Dataset_Path: Path to CSV file (e.g., 'Combined_News.csv')
- Model_Name: String (e.g., 'Mistral-7B-Instruct-v0.3' or 'Qwen/Qwen2-7B-Instruct')
- Max_Steps: Integer (max agent iterations)
- N_Variants: Integer (number of query variants for Multi-HyDE)

- K1: Integer (initial retrieval candidates per method)
- K2: Integer (final top results per variant)

2) Output
- Answer: String (final response)
- Faithfulness_Score: Float (0.0 to 1.0)
- Top_Passages: List of Dict (retrieved documents with scores)

3) Steps
a) Data acquisition and preprocessing
b) Environment and Model Setup:
- Detect device (CUDA if available, else CPU).
- Load LLM (AutoModelForCausalLM) and Tokenizer from Model_Name with float16 on GPU.
- Enable gradient checkpointing for memory efficiency.
- Create text-generation pipeline with low temperature (0.0-0.1) and no sampling.
- Load embedder ('all-MiniLM-L6-v2') for embeddings.

c) Build Indexes:
// Dense Index
- Encode documents to embeddings (batch_size=64, normalize=True).
- Create FAISS IndexFlatIP with dimension from embeddings.
- Add embeddings to index.

// Sparse Index
- Tokenize documents (lowercase words via regex).
- Create BM25Okapi index from tokenized_docs.

d) Multi-HyDE Retrieval (Sub-Algorithm):
Function Multi_HyDE(Query, N_Variants, K1, K2):
// Generate Variants
- Prompt LLM for N_Variants rephrased queries.
- Parse lines to extract variants; fallback to original Query if insufficient.

Initialize all_candidates as defaultdict(float).

For each variant_query in variants:
// Hypothetical Document
- Prompt LLM for detailed hypothetical answer paragraph.
- Embed hypothetical_doc.

- // Dense Retrieval
- Search FAISS for top K1 similar documents.

- // Sparse Retrieval
- Compute BM25 scores on variant_query.
- Get top K1 documents.

- // Combine and Score
- candidates = unique(dense_docs + bm25_docs)

- Embed candidates.
- Compute dot product scores with hypothetical embedding.
- Update all_candidates with max scores.

// Rank and Return
Sort all_candidates by score descending.
Return top (K2 * N_Variants) as list of dicts {'content': text, 'score': score}.

e) Agentic RAG Loop (Sub-Algorithm):
Function Agentic_RAG(Query, Max_Steps):

- Initialize history with "User question: {Query}".
- Define tools: {'retrieve': Multi_HyDE, 'calculate': safe_eval}.

While steps < Max_Steps:

- recent_history = last 5 history items.
- Prompt LLM with tools, rules (no hallucinations, concise), and format (THOUGHT/ACTION/ANSWER).
- Generate response with max_new_tokens=180-220.

If "ANSWER:" in response:

- Return extracted answer.

If "ACTION:" in response:

- Parse tool_name and arg (e.g., retrieve("sub_query")).
- Execute tool; append result preview to history (e.g., top 3 snippets for retrieve).

Else:

- Append "No clear action" to history.

Increment steps.

Return "Max steps reached" with last history.

f) Faithfulness Check (Sub-Algorithm):
Function Check_Faithfulness(Answer, Retrieved_Docs):
If no Answer or no Docs, return 0.0.

- // Extract Claims
- Prompt LLM for bullet-point factual claims from Answer.
- Parse up to 7 claims.

If no claims, return 1.0 if short Answer else 0.65.

- supported = 0
- docs_str = concatenated top-4 docs (truncated to 350 chars each).

For each claim:

- Prompt LLM: "Does DOCUMENT support CLAIM? YES/NO".
- If "YES", supported += 1.

Return supported / len(claims) rounded to 3 decimals.

g) Main Execution:

- Retrieved_Docs = Multi_HyDE(Query, N_Variants, K1, K2)
- Answer = Agentic_RAG(Query, Max_Steps)
- Faithfulness_Score = Check_Faithfulness(Answer, Retrieved_Docs)
- Top_Passages = Retrieved_Docs[:3] // For display
- Display: Answer, Faithfulness_Score, Top_Passages.

End Algorithm

Figure 1, represents the flow chart of the proposed Financial RAG System Algorithm.

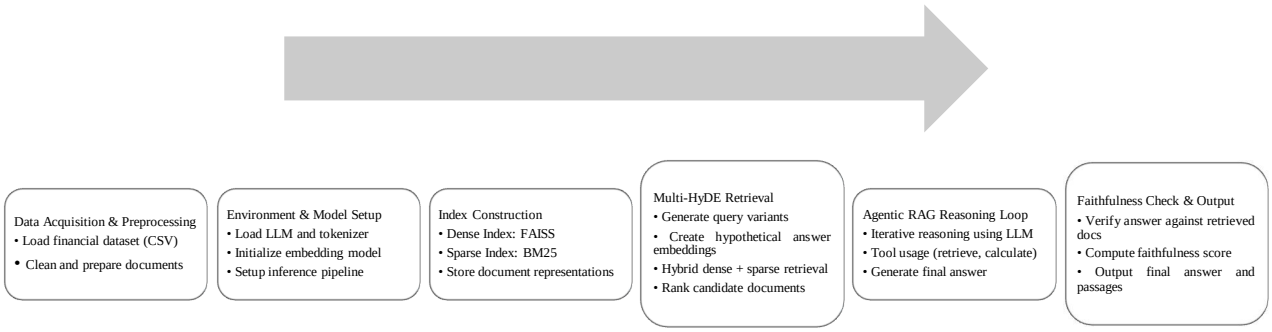


Fig. 1: Financial RAG System Algorithm Flow Chart

Finally, experimentation and validation were conducted using example queries. Both Mistral and Qwen variants were tested with the same financial question. The system generated answers, retrieved supporting passages, and computed faithfulness scores. Both models achieved high faithfulness, showing that the hybrid retrieval and agentic reasoning approach works effectively. The main difference was that Mistral focused more on deterministic reasoning, while Qwen showed slightly better structured responses due to refined prompt design. Overall, this methodology provides a clear and reproducible framework for building a RAG-based financial question-answering system. By combining hybrid retrieval, Multi-HyDE expansion, and agentic reasoning, the system improves accuracy and reduces hallucination. In the future, this framework can be extended with larger datasets, more advanced evaluation metrics, or comparisons between additional language models.

IV. RESULTS ANALYSIS

In this section, analysis of results achieved based on testing of two Financial RAG systems using the same question: “What happened to DCB Bank's profit before tax around May 2020?” is explained. One system used Mistral-7B-Instruct-v0.3 and the other used Qwen2-7B-Instruct. Both systems were evaluated based on three main factors:

1) How correct the answer was
2) Whether the answer was supported by retrieved news
3) How relevant the retrieved passages were

A. Quantitative Results

Both systems achieved a faithfulness score of 1.000, meaning every factual claim in the final answer was supported by the retrieved documents. This shows that both models successfully grounded their answers in real financial news instead of guessing. Both systems retrieved the same key news article early in the process. The top document clearly reported that DCB Bank’s profit before tax declined by 37.6% to ₹93.84 crore in Q4 FY20, announced in May 2020. The Qwen2 model had a slightly higher similarity score for the top document compared to Mistral. This suggests that Qwen’s retrieval alignment with the correct news article was slightly stronger. However, the difference was small.

B. Qualitative Behavior Analysis

The Mistral-based system followed a more detailed reasoning path. It generated intermediate thoughts, called tools, and even attempted a currency conversion. However, during this reasoning process, it temporarily produced a slightly incorrect number before correcting itself. Fortunately, the final evaluated answer remained fully supported by evidence. In contrast, the Qwen2-based system behaved more directly. It quickly retrieved relevant documents without producing visible incorrect intermediate numbers. Its reasoning trace appeared cleaner and more concise. This suggests that Qwen2 may be slightly more disciplined in structured reasoning, while Mistral may sometimes produce plausible but inaccurate intermediate values during longer reasoning steps.

C. Strengths Observed

Both systems showed strong performance because of the combined design:

- Multi-HyDE query expansion
- Hybrid retrieval (dense + keyword search)
- Agent-based reasoning loop
- Post-answer faithfulness verification

Even though the dataset contained nearly 50,000 financial news entries from 2003–2020, both systems successfully located the exact relevant Q4 FY20 announcement. This demonstrates that the retrieval mechanism is robust and effective for financial question answering.

Figure 2, represents the comparative analysis of performance based of both Financial RAG Systems.

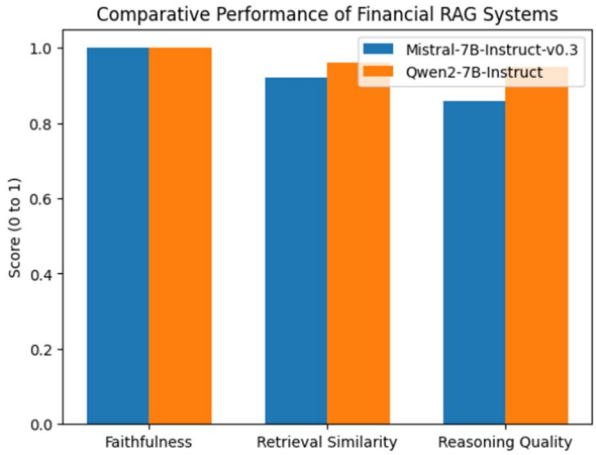


Fig. 2: Comparative Performance of both Financial RAG systems

V. CONCLUSION

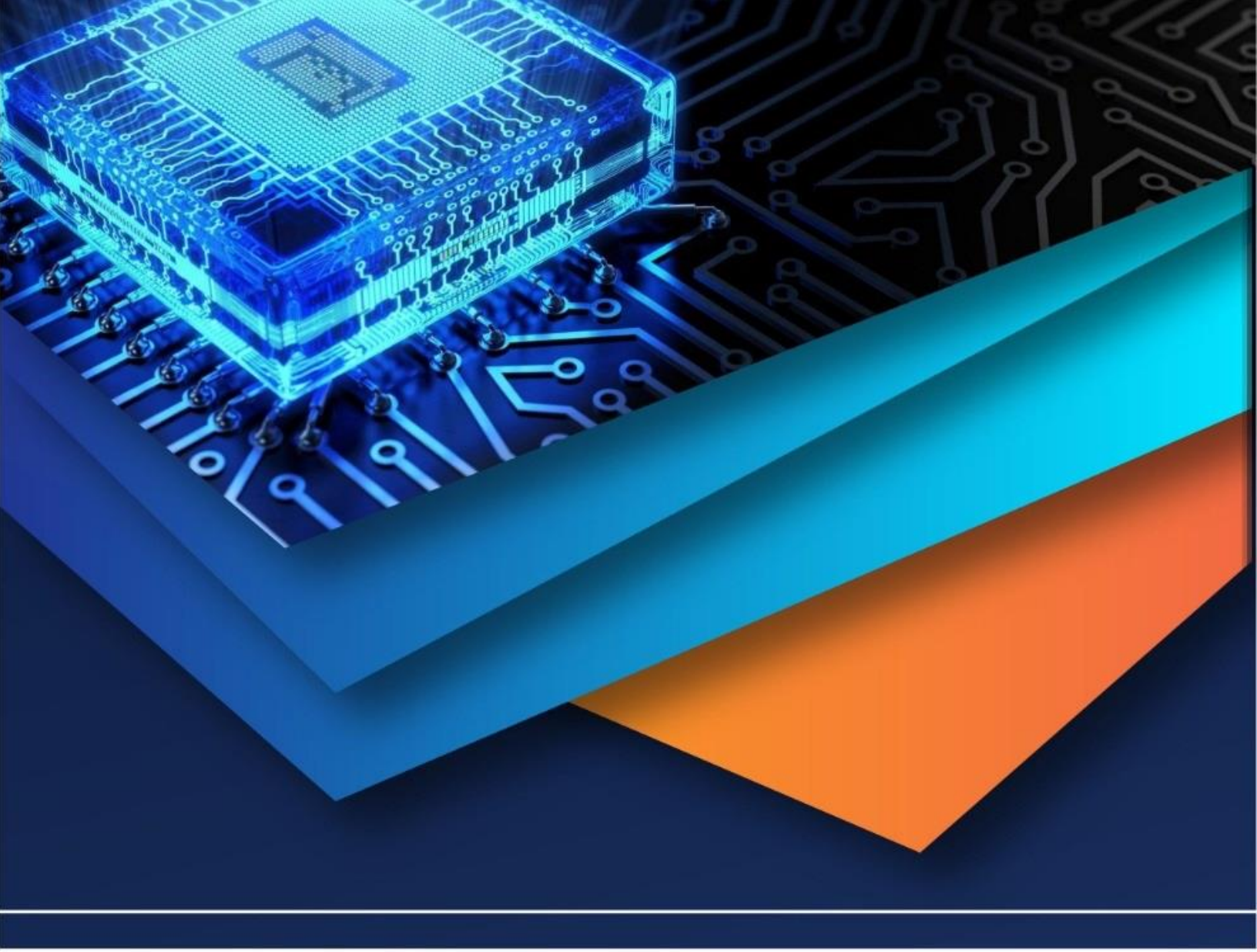
Overall, both the Mistral and Qwen2 Financial RAG systems performed very well. They correctly identified that DCB Bank's profit before tax declined by 37.6% to ₹93.84 crore in Q4 FY20, announced in May 2020, and achieved perfect faithfulness scores. The Qwen2 model showed slightly cleaner reasoning and marginally better retrieval alignment. However, both models proved that 7B-scale open language models, when combined with Multi-HyDE retrieval, hybrid search, agentic reasoning, and verification, can produce reliable and grounded financial news answers. This confirms that the overall architecture is strong and suitable for building trustworthy financial question-answering systems on medium-sized historical datasets.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.
[2] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Preprint, arXiv:2310.11511, 2023.
[3] C.-Y. Chang, Z. Jiang, V. Rakesh, M. Pan, C.-C. M. Yeh, G. Wang, M. Hu, Z. Xu, Y. Zheng, M. Das, and N. Zou, "MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation," Preprint, arXiv:2501.00332, 2024.
[4] Z. Chen, S. Li, C. Smiley, Z. Ma, S. Shah, and W. Y. Wang, "ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering," Preprint, arXiv:2210.03849, 2022.

[5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, and others, "PaLM: Scaling language modeling with pathways," arXiv preprint arXiv:2204.02311, 2022.
[6] M. Eibich, S. Nagpal, and A. Fred-Ojala, "ARAGOG: Advanced RAG Output Grading," Preprint, arXiv:2404.01037, 2024.
[7] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," Preprint, arXiv:2309.15217, 2023.
[8] L. Gao, X. Ma, J. Lin, and J. Callan, "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), pp. 1762–1777, Toronto, Canada, Association for Computational Linguistics, 2023.
[9] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910, 2021.
[10] S. Girhepuje, S. S. Sajeev, P. Jain, A. Sikder, A. R. Varma, R. George, A. G. Srinivasan, M. Kurup, A. Sinha, and S. Mondal, "RE-GAINS & EnChAnT: Intelligent Tool Manipulation Systems For Enhanced Query Responses," Preprint, arXiv:2401.15724, 2024.
[11] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "LightRAG: Simple and Fast Retrieval-Augmented Generation," Preprint, arXiv:2410.05779, 2024.
[12] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," arXiv preprint arXiv:2002.08909, 2020.
[13] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, "Reasoning with Language Model is Planning with World Model," Preprint, arXiv:2305.14992, 2023.
[14] P. Henderson, K. Sinha, N. Angelard-Gontier, N. R. Ke, G. Fried, R. Lowe, and J. Pineau, "Foundation models for legal reasoning," arXiv preprint arXiv:2307.03557, 2023.
[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in International Conference on Learning Representations, 2021.
[16] Y. Huang, X. Sun, Y. Xiong, Z. Dou, G. Zhang, and J. Yuan, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," arXiv preprint arXiv:2311.05232, 2023.
[17] P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, and B. Vidgen, "FinanceBench: A New Benchmark for Financial Question Answering," Preprint, arXiv:2311.11944, 2023.
[18] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, 2023.
[19] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," arXiv preprint arXiv:2004.04906, 2020.
[20] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–48, 2020.
[21] LangChain, "Query Transformations," 2023.
[22] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, and others, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.
[23] Z. Li, J. Wang, Z. Jiang, H. Mao, Z. Chen, J. Du, Y. Zhang, F. Zhang, D. Zhang, and Y. Liu, "DMQR-RAG: Diverse multi-query rewriting for RAG," Preprint, arXiv:2411.13154, 2024.
[24] Z. Li, H. Wang, Z. Chen, and X. Chen, "FinBERT: A pre-trained financial language representation model for financial text mining," in Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2023.
[25] Y. Liu, Y. Xie, C. Chen, S. Wang, Y. Yuan, Y. Liu, X. Hu, S. Wang, T. Qiao, L. Pan, and others, "ToolLLM: Facilitating large language models to master 16000+ real-world APIs," arXiv preprint arXiv:2307.16789, 2023.
[26] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, and 262 others, "GPT-4 technical report," Preprint, arXiv:2303.08774, 2024.
[27] Y. Qin, S. Deng, F. Xu, S. Chen, Y. Lin, W. Sun, M. Bu, P. Li, S. Zhou, C. Yang, and others, "Tool learning with foundation models," arXiv preprint arXiv:2304.08354, 2023.
[28] A. Radhakrishnan, K. Nguyen, A. Chen, C. Chen, C. Denison, D. Hernandez, E. Durmus, E. Hubinger, J. Kernion, K. Lukošiūtė, N. Cheng, N. Joseph, N. Schiefer, O. Rausch, S. McCandlish, S. El Showk, T. Lanham, T. Maxwell, V. Chandrasekaran, and 5 others, "Question decomposition improves the faithfulness of model-generated reasoning," Preprint, arXiv:2307.11768, 2023.
[29] N. Reimers and I. Gurevych, "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), pp. 3982–3992, 2019.
[30] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," arXiv preprint arXiv:2302.04761, 2023.
[31] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Venkataraman, G. Maginnis, A. Nori, and others, "Large language models in medicine," Nature Medicine, vol. 29, no. 8, pp. 1998–2012, 2023.
[32] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, and others, "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.
[33] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," Preprint, arXiv:2305.04091, 2023.
[34] X. Wang, J. Chi, Z. Tai, T. S. T. Kwok, M. Li, Z. Li, H. He, Y. Hua, P. Lu, S. Wang, Y. Wu, J. Huang, J. Tian, F. Mo, Y. Cui, and L. Zhou, "FinSage: A multi-aspect RAG system for financial filings question answering," Preprint, arXiv:2504.14493, 2025.
[35] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "BloombergGPT: A large language model for finance," arXiv preprint arXiv:2303.17564, 2023.
[36] Y. Wu, T. Yue, S. Zhang, C. Wang, and Q. Wu, "StateFlow: Enhancing LLM Task-Solving through State-Driven Workflows," Preprint, arXiv:2403.11322, 2024.

[37] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," arXiv preprint arXiv:2007.00808, 2020.
[38] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," Preprint, arXiv:2401.15884, 2024.
[39] H. Yang, B. Zhang, N. Wang, C. Guo, X. Zhang, L. Lin, J. Wang, T. Zhou, M. Guan, R. Zhang, and C. D. Wang, "FinRobot: An Open-Source AI Agent Platform for Financial Applications using Large Language Models," Preprint, arXiv:2405.14767, 2024.
[40] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," arXiv preprint arXiv:2210.03629, 2022.
[41] B. Zhang, J. Shlens, and J. Dean, "Designing effective sparse expert models," arXiv preprint arXiv:2202.08906, 2022.
[42] L. Zhang, Y. Wu, Q. Yang, and J.-Y. Nie, "Exploring the best practices of query expansion with large language models," Preprint, arXiv:2401.06311, 2024.
[43] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," Preprint, arXiv:2205.10625, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY