A Survey on Retrieval-Augmented Generation: From Naive to Adaptive Approaches with Financial Insights

Yug V. Patel[1], Dr. Vijaykumar Salvia[2,] and Dr. Indrajeet Kumar[3]

[1] M. Tech Student, Department of CSE, Parul University, Vadodara, Gujarat, India

[2] Professor, Department of CSE, Parul University, Vadodara, Gujarat, India

Abstract: This survey examines the evolving landscape of Retrieval-Augmented Generation (RAG) systems, from naive approaches to adaptive and specialized implementations, focusing on financial applications. We explore various RAG architectures including Naive RAG [1], Advanced RAG [1], Modular RAG [2], Adaptive RAG [3], Corrective RAG [4], Self-RAG [5], Hybrid RAG [6], and Graph RAG [24,25]. The paper delves into retrieval methods, including dense [8,9] and sparse [10,11] techniques, and discusses augmentation strategies such as zero-shot [16] and few-shot [17] prompting. We analyze the FinanceBench dataset [20] as a case study, highlighting challenges in answering financial questions and proposing future directions for RAG systems in finance. The survey emphasizes the potential of domain-specific fine-tuning [23] and hybrid retrieval methods [6] to enhance RAG performance in complex financial contexts.

Keywords: AI, FinanceBench, Retrieval Augmented Generation (RAG).

1. INTRODUCTION

In natural language processing, particularly in question answering, RAG (Retrieval Augmented Generation) stands out as a distinctive approach that merges the strengths of both retrieval-based and generation-based techniques. Traditional Q&A systems primarily rely on these two strategies: retrieval-based and generation-based methods. Finding pertinent sections or documents within a huge corpus of text is the foundation of retrieval-based approaches, which select the best response from these retrieved sources. Although these techniques are effective in locating pertinent data, their applicability may be hampered by the corpus's coverage and the quality of the retrieval procedure. On the other hand, generation-based approaches create responses from the ground up depending on the context and the input question. Although these techniques can produce various contextually relevant solutions, they may not always be successful.

2. AN OVERVIEW OF RAG ARCHITECTURE

2.1 Naïve Retrieval Augmented Generation (RAG)

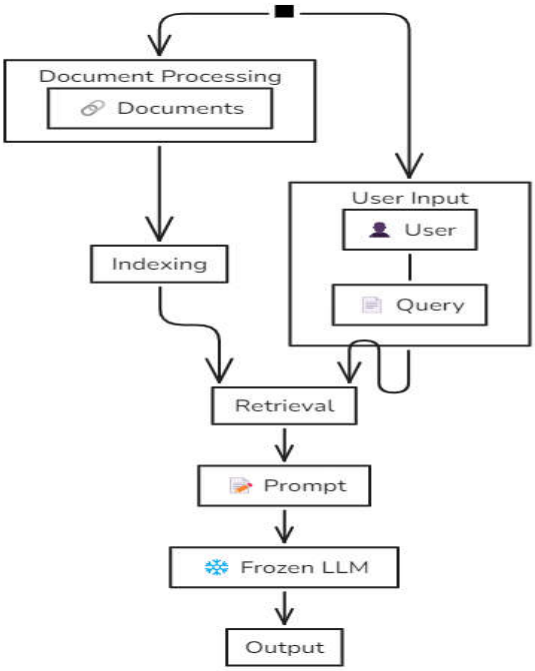

Fig. 1. Architecture of Naive Retrieval-Augmented Generation (RAG) System.

In the paper [1] Naive RAG is described as a basic form of the Retrieval-Augmented Generation framework. It primarily involves a straightforward integration of a retrieval system with a generative model, where the retriever fetches relevant documents or passages based on the input query. These documents are then used by the generative model to enhance the output with accurate and relevant information. Naive RAG is considered foundational and serves as a crucial baseline for evaluating more complex RAG systems. Despite its simplicity, it effectively demonstrates the benefits of incorporating external knowledge into generative models, particularly for knowledge-intensive tasks.

2.2 Advanced RAG

Advanced Retrieval-Augmented Generation (Advanced RAG) represents a significant evolution in integrating external knowledge into Large Language Models (LLMs). It enhances the basic RAG framework with more sophisticated retrieval and generation techniques, addressing issues of information accuracy and content relevance [1].

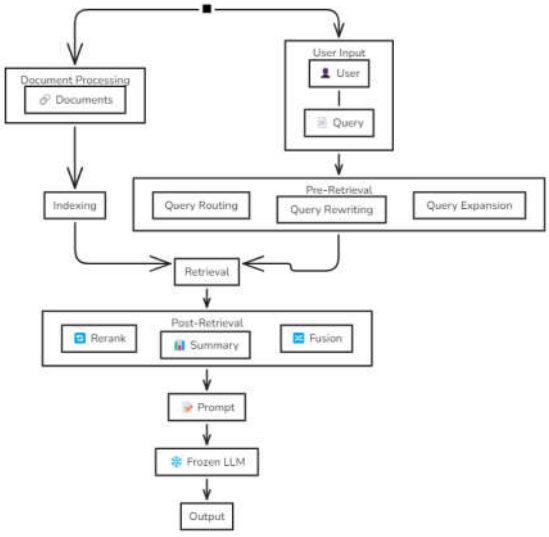

Fig. 2. Enhanced Architecture of Advanced Retrieval-Augmented Generation (RAG) System.

Advanced RAG is particularly effective for tasks requiring high precision and domain-specific knowledge. It employs improved algorithms for retrieving pertinent information from vast databases and generating contextually appropriate, factually accurate responses. This advancement leads to more dynamic and reliable outputs, reducing common LLM problems like hallucination and outdated information. By refining both retrieval and generation processes, Advanced RAG bridges the gap between extensive knowledge repositories and LLM capabilities. This progress has significant implications for various applications, from enhancing question-answering systems to improving automated content generation in specialized fields. Advanced RAG represents a significant breakthrough in natural language processing, leading to the creation of more intelligent and context-sensitive AI systems.

2.3 Modular RAG

Modular RAG systems are engineered to boost the effectiveness of large language models by integrating retrieval mechanisms that actively fetch pertinent information from databases or extensive text collections. This combination greatly improves the precision and relevance of the content produced by these models. The term "modular" in RAG systems highlights their flexible architecture, enabling various system components to be easily reconfigured or swapped out to meet specific tasks or requirements. This modularity ensures that RAG systems are versatile and adaptable, making them suitable for a broad spectrum of applications, ranging from natural language processing tasks to more intricate data handling and content generation across different domains.[1], [2]

2.4 Adaptive RAG

Adaptive-RAG is an innovative QA framework that adjusts dynamically to select the most suitable method based on the complexity of the query, thereby enhancing the performance of retrieval-augmented large language models (LLMs). The system utilizes a classifier, a smaller language model, to assess the complexity of incoming queries. This allows the system to choose between several tactics, from straightforward no-retrieval procedures to more intricate iterative retrieval-augmented options. The framework fluidly adjusts to different query complexities, striving to strike a compromise between accuracy and processing performance. It has been proven to be more accurate and efficient than current techniques on several open-domain quality assurance datasets.[3]

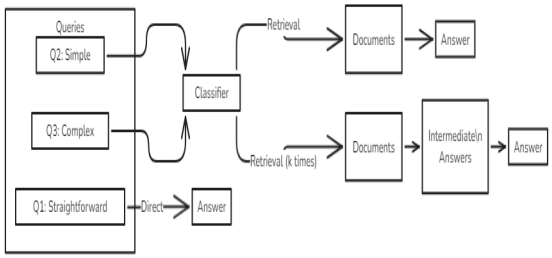

Fig. 3. Workflow of Adaptive Retrieval-Augmented Generation (RAG) Framework.

2.5 Corrective RAG

Corrective Retrieval-Augmented Generation, or CRAG, has been a notable development in RAG frameworks recently, offering a substantial enhancement over the intrinsic drawbacks of RAG models. To initiate targeted remedial activities, CRAG presents a lightweight retrieval evaluator that rates the relevance of retrieved

documents and provides confidence scores. This reduces the

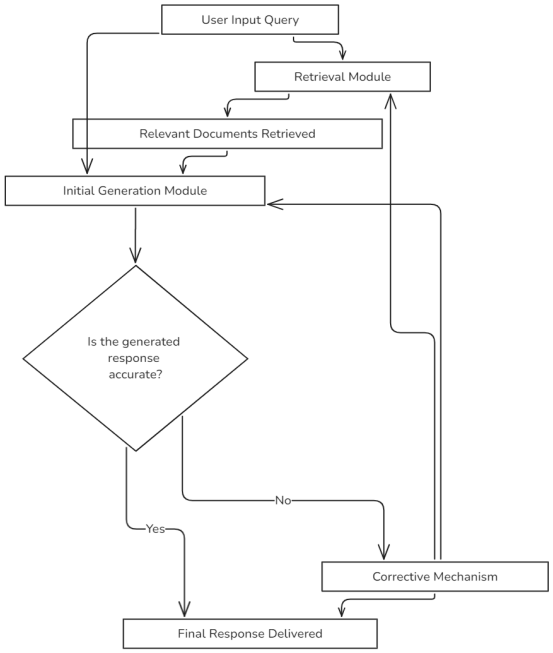

Fig. 4. Process Flow of Corrective Retrieval-Augmented Generation (CRAG) System.

possibility of including erroneous data by guaranteeing that only the most pertinent information is used during the generating process.[4] Furthermore, CRAG supplements static corpora with large-scale web searches to increase the accuracy and diversity of the knowledge base. The decompose-then-recompose algorithm, which eliminates unnecessary information to improve the accuracy and efficiency of knowledge consumption, is a fundamental innovation of CRAG.

2.6 Self-RAG

Self-RAG signifies a major advancement in the domain of large language models (LLMs). By incorporating adaptive retrieval [3] and self-reflection mechanisms, Self-RAG addresses the limitations of traditional retrieval-augmented models. The model's ability to dynamically retrieve relevant information and critically evaluate its own responses enhances its controllability and adaptability to diverse tasks. Experimental results consistently demonstrate Self-RAG's superiority over standard LLMs and other retrieval-augmented models, particularly in terms of factuality and citation accuracy. This innovative approach is poised to have a profound impact on various applications, including open-domain question answering, reasoning, fact verification, and long-form generation [5].

2.7 Hybrid RAG

The Hybrid Retrieval-Augmented Generation (RAG) System represents a notable advancement in optimizing large language models (LLMs) for retrieval-augmented generation tasks. This system integrates several crucial enhancements aimed at boosting accuracy and addressing the common problem of hallucinations in LLMs. Notable innovations include optimized retrieval processes that efficiently leverage text chunks and tables from web sources, the integration of attribute predictors to reduce errors, and the deployment of both a Large Language Model Knowledge Extractor and a Knowledge Graph Extractor. These components collaboratively enhance the model's reasoning abilities and numerical computation accuracy. The effectiveness of this hybrid system was demonstrated during the Meta CRAG KDD Cup 2024, where it secured a third-place finish in Task 1 and achieved first place in five out of seven question types in Task 2, competing against a robust field of over 2,000 participants and 5,500 submissions. This hybrid approach not only elevates model performance but also establishes a new standard for complex reasoning tasks in computational models[6].

2.8 Agentic RAG

The survey of LLM-based Multi-Agents provides a valuable foundation for understanding the broader context of LLMs in multi-agent systems[7]. While the survey does not explicitly address Agentic RAG, it offers insights into the planning, reasoning, and communication capabilities of LLMs, which are essential for integrating RAG systems into complex problem-solving environments. The survey's discussion of LLM profiling and

communication within multi-agent frameworks provides a useful starting point for exploring the potential applications of Agentic RAG in enhancing retrieval and generation processes.

2.9 Graph RAG

Graph Retrieval-Augmented Generation (Graph RAG) emerges as a significant advancement in RAG systems, specifically designed to tackle limitations of LLMs like hallucinations and outdated information. this approach utilizes structured knowledge graphs for precise information retrieval, mitigating issues found in traditional RAG. It works through three key steps: 1) Graph-Based Indexing structures knowledge for efficient retrieval, 2) Graph-Guided Retrieval leverages connections within the graph to find relevant information, and 3) Graph-Enhanced Generation integrates this information for accurate and contextually rich LLM outputs. These surveys highlight the potential of Graph RAG across various domains while outlining challenges and future research directions. Integrating Graph RAG with LLMs promises significant improvements in applicability and accuracy for complex tasks and queries.

3. Retrieval

In RAG, retrieval involves sourcing relevant information from external databases to enhance the performance of language models. This mechanism is essential for increasing the accuracy and relevance of generated responses, especially in complex tasks such as relation extraction and question answering.

3.1 Dense Retrieval

Dense retrieval involves using dense vector representations of queries and documents, often obtained through neural networks, to find the most relevant documents.

3.1.1 Dual-Encoder Models.

The paper [8] presents an innovative method for passage retrieval utilizing dense representations, which are developed through a dual-encoder model. This approach significantly boosts the efficiency of open-domain question answering systems by shifting from traditional sparse vector space models such as TF-IDF or BM25. The dual-encoder framework functions by separately encoding questions and passages into dense vectors, which are then employed to calculate similarity scores for retrieval tasks. This technique enables a more sophisticated understanding and matching based on semantic similarities rather than just keyword overlap. The effectiveness of this approach is evidenced by its superior performance compared to a robust Lucene-BM25 system, achieving absolute improvements of 9%-19% in top 20 passage retrieval accuracy across various open-domain QA datasets.

3.1.2 Contextualized Embeddings.

In their 2019 paper, "Latent Retrieval for Weakly Supervised Open Domain Question Answering" [9], Lee et al. propose a groundbreaking method for open-domain question answering (QA) that moves away from the conventional reliance on highly supervised evidence and opaque information retrieval (IR) systems. They present a method where both the retriever and reader components are trained together directly from question-answer pairs, eliminating the need for an existing IR system. This technique shifts the focus by treating evidence retrieval from vast sources like Wikipedia as a latent variable, moving away from the conventional need for explicit evidence supervision. To support this learning approach, the authors pre-train the retriever using an Inverse Cloze Task, which effectively captures contextualized embeddings that are pertinent to the questions. This method has been proven to significantly outperform traditional IR systems, such as BM25, particularly in scenarios where users are actively searching for information, with improvements of up to 19 points in exact match scores. This advancement highlights the crucial role of contextualized embeddings in enhancing retrieval accuracy in weakly supervised settings, making it an essential element for systems aiming to advance open-domain QA.

3.2 Sparse Retrieval

3.2.1 BM25 Algorithm.

BM25 is a well-established algorithm in the field of information retrieval, particularly noted for its effectiveness in ranking documents based on their relevance to a given search query. Originating from the probabilistic retrieval framework, BM25 calculates the relevance score of documents by considering term frequency (the number of times a term appears in a document) and inverse document frequency (how common or rare a term is across all documents in the collection). This balance helps to address the issues of term frequency saturation and the varying

document lengths. The effectiveness of BM25 in legal case retrieval is underscored by its performance in the COLIEE 2021 competition, where a straightforward implementation of BM25 achieved second place, demonstrating its robustness and reliability as a baseline method in complex domains such as legal document retrieval [8].

3.2.2 TF-IDF Vectorization.

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a fundamental technique in automatic text retrieval, widely recognized for its importance in the literature. It is a statistical method used to assess the significance of a word in a document relative to a collection of documents (or corpus). The primary concept is to balance the frequency of a term in a specific document (term frequency, TF) with how common or rare the term is across all documents (inverse document frequency, IDF). This balance helps in pinpointing terms that are both relevant and distinctive for a given document.

The TF-IDF methodology comprises two main components:

Term Frequency (TF): This measures the frequency of a term in a document. The underlying assumption is that the more frequently a term appears in a document, the more important it is. However, this frequency is normalized to avoid bias towards longer documents.

Inverse Document Frequency (IDF): This component assesses the importance of a term within the entire corpus. It operates on the assumption that terms appearing in many documents are less informative than those appearing in fewer documents. IDF is mathematically calculated as the logarithm of the total number of documents divided by the number of documents containing the term.

The TF-IDF score is obtained by multiplying the TF and IDF values for each term, resulting in a weight that reflects the term's significance in the document while diminishing the weight of commonly used terms that are less informative.

This approach is particularly notable for its simplicity and effectiveness. Despite being developed decades ago, TF-IDF remains a cornerstone technique in text mining and information retrieval. One might question its performance in modern real-world scenarios, especially with the emergence of advanced models like word embeddings and transformers. Nonetheless, TF-IDF's interpretability and ease of implementation make it a preferred method for many applications, including search engines and document clustering.

Compared to other term-weighting methods, TF-IDF offers a balance between computational efficiency and retrieval performance. While more sophisticated models might provide enhancements in certain contexts, TF-IDF's robustness and simplicity ensure its continued relevance. The insights gained from TF-IDF vectorization have paved the way for more advanced techniques, offering a baseline against which newer methods can be evaluated [9].

3.3 Hybrid Retrieval

Hybrid retrieval, as discussed in the context of RAG, represents a sophisticated blend of techniques aimed at enhancing the performance of LLMs and AI-generated content systems. This method combines conventional retrieval techniques with advanced, adaptive retrieval strategies to create a robust framework for extracting and integrating information. The core strength of hybrid retrieval is its capability to not only gather relevant data but also to dynamically refine the retrieval process according to the context or specific requirements of the task [1].

What we find particularly intriguing about this approach is its potential to significantly reduce the common pitfalls associated with large language models, such as the generation of outdated or irrelevant content. By integrating a hybrid retrieval mechanism, these models can dynamically access the most current and relevant information, potentially leading to more accurate and contextually appropriate outputs. One might wonder how this method would perform in real-world scenarios where the demand for up-to-the-minute information is critical, such as in news generation or financial forecasting.

Moreover, hybrid retrieval could be seen as a bridge between static knowledge bases and the fluid, ever-changing nature of real-world data. It seems that this approach not only supports the generation of more credible and reliable content but also enhances the learning capabilities of the model by exposing it to a broader array of data sources. The results suggest that hybrid retrieval could pave the way for more sophisticated, adaptable, and efficient AI systems, which could be a game-changer in fields requiring high levels of accuracy and timeliness in information retrieval and utilization[10].

3.4 Retriever Finetuning

Retriever fine-tuning in RAG is pivotal for enhancing the performance of AI-generated content systems by optimizing the retrieval component. This process involves refining the retriever's parameters to more effectively select relevant information from a dataset, which then informs the generation process. The significance of this method lies in its capacity to dynamically update and adapt to new information, thereby maintaining the relevance and accuracy of the generated content over time.

The methodology typically involves training the retriever on a specific task or dataset to improve its ability to identify and retrieve the most pertinent data. This step is crucial because the quality of the retrieved input greatly impacts the output of the generator. By fine-tuning the retriever, the system can produce more accurate and contextually appropriate responses. Furthermore, this approach represents an improvement over traditional methods that might rely on static or less adaptive retrieval mechanisms [10].

4. AUGMENTATION

It refers to combining the user query with the retrieved context within a single template. This template also includes the instruction, known as a prompt. So ultimately, it's called a prompt template[11]. The retrieved context is simply the information we get from the retrieval process based on the user's question. the technique of defining a prompt template is called prompt engineering[12], [13]. there are many ways of defining prompt templates.

4.1 Zero-short prompting

This method leverages the extensive pre-training data of LLMs to apply them to new tasks without requiring specific training data for each task. Zero-shot prompting relies on a single prompt, meticulously crafted to describe the task at hand, to guide the LLM in generating responses. The paper [14] emphasizes the importance of designing effective prompts that clearly communicate the desired task to the model, thereby maximizing performance without the need for additional labelled training examples.

The primary benefit of zero-shot prompting is its capacity to quickly adapt large language models (LLMs) to new tasks, especially in situations where obtaining labeled data is impractical or impossible. However, creating prompts that are both clear and accurately representative of the task can be challenging. The effectiveness of the model's performance is heavily influenced by the quality and specificity of the input prompt. Despite these challenges, zero-shot prompting has a wide range of potential applications, including answering domain-specific questions and generating creative content.

4.2 Few-short prompting

The few-shot prompt technique represents a compelling approach to leveraging large pre-trained language models (PLMs) for tasks in low-resource languages. This technique involves providing the PLM with a minimal number of examples (few-shot) of a specific task in a target language, which serves as a context for the model to learn and perform the task on new, unseen data [15]. The beauty of this method lies in its simplicity and efficiency, particularly when resources are scarce or when training data is limited.

What we found particularly intriguing about this approach is its potential to democratize access to advanced NLP technologies across diverse linguistic landscapes. Using only a few examples, the model can effectively adapt to new languages and tasks, marking a significant shift from traditional methods that demand extensive data and computational resources.

5. GENERATION

The generation part in RAG involves generating the answer or response based on a given prompt template with the help of LLM [16]. There are two types of LLMs available in the industry, one is closed source [18], and the other one is open source [18]. For using closed source LLMs, we have to pay the amount to the respective organization on the other hand opensource LLMs are completely free, we can use them in our local computer system or by using some inference engine [17], however in the inference engine case, the user may have to pay some cost associated with inference.

6. Literature Review on FinanceBench Dataset

6.1 Overview Of FinanceBench Dataset

The FinanceBench dataset serves as a thorough benchmark for assessing the performance of large language models (LLMs) in financial question answering (QA) [18]. It comprises 10,231 questions about 40 publicly traded companies, derived from 361 public filings, including 10-Ks, 10-Qs, and earnings reports, covering the period from 2015 to 2023. The questions are categorized into three types: domain-relevant, novel-generated, and metrics-generated, ensuring coverage of various financial analysis scenarios. Each question is paired with an answer, evidence, and relevant metadata. This benchmark is crucial for assessing LLMs in real-world financial contexts, especially in tasks involving numerical reasoning, information extraction, and logical deductions. The dataset highlights the limitations of current LLMs in accurately handling complex financial QA, as even state-of-the-art models struggle with tasks requiring in-depth financial knowledge and reasoning.

Key Findings for the FinanceBench

1. Document Chunking for RAG in Financial Reports
   Antonio Jimeno Yepes et al. (2024) in the paper [19] introduced a novel document chunking approach specifically designed for financial documents in RAG systems. Traditional chunking methods often disregard document structures, leading to suboptimal retrieval. The authors proposed a method that chunks documents based on their structural elements, such as titles and tables, which enhances the relevance and accuracy of the retrieved content. This method showed a significant improvement in processing financial documents, which often contain complex layouts and dense information.

2. Improving RAG Retrieval for Financial Documents
   Spurthi Setty et al. (2024) explored the limitations of traditional RAG pipelines in financial document retrieval and proposed enhancements like query expansion and re-ranking algorithms. Their study research [20], emphasized that standard retrieval methods often fail due to the complex nature of financial texts. By integrating more sophisticated chunking and re-ranking methods, their approach improved the accuracy and relevance of the information retrieved, thereby reducing the hallucination problem in LLMs.

3. Impact of Domain-Specific Fine-Tuning on RAG Systems
   Zooey Nguyen et al. (2024) in the paper [21] examined the effects of fine-tuning both embedding models and LLMs on the FinanceBench dataset. The study demonstrated that fine-tuning significantly improves RAG performance, especially when combined with iterative reasoning frameworks like the OODA loop. Their findings suggest that while generic RAG models struggle with domain-specific queries, fine-tuned models can achieve near-human-expert accuracy.

6.2 Challenges and Limitations

Complex Document Structures**:** Financial documents often contain complex structures, such as tables, footnotes, and multi-part sections. This complexity poses a significant challenge for AI systems, particularly in accurately chunking and retrieving relevant information.

Need for Domain Expertise**:** The dataset's questions require a deep understanding of financial concepts, which can be difficult for generic AI models. This limitation has driven research into domain-specific fine-tuning and specialized retrieval strategies.

Scalability Issues**:** As the dataset is based on publicly available documents, scaling it to cover more companies or additional years of filings would require significant manual effort, particularly in crafting new questions and verifying answers.

High computational cost of fine-tuning large language models (LLMs) on domain-specific datasets like FinanceBench. Fine-tuning requires substantial computational resources, including powerful GPUs or TPUs, and can be time-consuming, especially when dealing with large models like GPT-3 or GPT-4. The cost can be

prohibitive for smaller organizations or research teams, limiting their ability to customize models for specific financial tasks.

6.3 Future Directions

Dataset Expansion: Future research could aim to enhance the FinanceBench dataset by incorporating more companies, extending the coverage to additional years of filings, and including a broader array of financial documents. Such expansion would further strengthen the dataset's robustness and applicability to a wider range of financial analysis tasks.

Integration with Other Datasets: Combining FinanceBench with other financial datasets, such as those focusing on market data or financial news, could create a more comprehensive benchmark for testing AI systems in finance.

Advanced RAG Architectures: Research into more advanced RAG architectures, such as modular [2] or graph-based [22], [23] models could lead to significant improvements in handling the complexities of financial documents.

Improve the retriever quality by fine-tuning or adopting hybrid RAG retrieval methods. Fine-tuning the retriever model [21] specifically on financial documents could significantly enhance its ability to identify and retrieve the most relevant information. Additionally, hybrid RAG retrieval techniques [6], which combine traditional retrieval methods with advanced machine learning models, can be adapted to better handle the complexity of financial data. By integrating vector-based retrieval [24] with traditional keyword search or incorporating domain-specific embeddings, hybrid approaches could offer more precise and contextually relevant retrieval, ultimately leading to more accurate and reliable Q&A performance.

7. CONCLUSION

The FinanceBench dataset marks a major advancement in assessing AI systems for financial analysis, providing a thorough benchmark for evaluating Retrieval-Augmented Generation (RAG) models and other AI techniques. Through a combination of realistic financial scenarios, diverse question types, and a focus on key financial statements, it provides an in-depth assessment of how well AI models can process and comprehend complex financial information.

The ongoing research has highlighted several challenges, such as the high computational cost of fine-tuning models and the difficulties in handling complex document structures. However, these challenges also present opportunities for future improvements.

RAG systems have demonstrated significant potential in improving the performance of large language models, especially in knowledge-intensive fields such as finance. By incorporating relevant retrieved documents, RAG systems enable LLMs to produce more precise and contextually relevant answers. The future direction of improving retriever quality through fine-tuning or adopting hybrid RAG retrieval methods could lead to even greater advancements, enabling more precise and reliable retrieval of financial data.

As research advances, integrating sophisticated RAG architectures and enhancing retrieval strategies will be essential for addressing existing limitations and advancing the capabilities of AI systems in financial analysis. These advancements will ensure that AI models can provide more accurate, reliable, and insightful financial information, ultimately aiding analysts, investors, and other stakeholders in making informed decisions.

8. REFERENCES

[1] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey,” Mar. 27, 2024, *arXiv*: arXiv:2312.10997. Accessed: Jul. 01, 2024. [Online]. Available: http://arxiv.org/abs/2312.10997

[2] Y. Gao, Y. Xiong, M. Wang, and H. Wang, “Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks,” Jul. 25, 2024, *arXiv*: arXiv:2407.21059. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2407.21059

[3] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," Mar. 28, 2024, *arXiv*: arXiv:2403.14403. Accessed: Jul. 04, 2024. [Online]. Available: http://arxiv.org/abs/2403.14403
[4] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," Feb. 16, 2024, *arXiv*: arXiv:2401.15884. Accessed: Jul. 03, 2024. [Online]. Available: http://arxiv.org/abs/2401.15884
[5] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Oct. 17, 2023, *arXiv*: arXiv:2310.11511. Accessed: Jul. 03, 2024. [Online]. Available: http://arxiv.org/abs/2310.11511
[6] Y. Yuan, C. Liu, J. Yuan, G. Sun, S. Li, and M. Zhang, "A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning," Aug. 09, 2024, *arXiv*: arXiv:2408.05141. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2408.05141
[7] T. Guo *et al.*, "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," Apr. 18, 2024, *arXiv*: arXiv:2402.01680. doi: 10.48550/arXiv.2402.01680.
[8] G. M. Rosa, R. C. Rodrigues, R. Lotufo, and R. Nogueira, "Yes, BM25 is a Strong Baseline for Legal Case Retrieval," Oct. 25, 2021, *arXiv*: arXiv:2105.05686. Accessed: Aug. 25, 2024. [Online]. Available: http://arxiv.org/abs/2105.05686
[9] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.
[10] P. Zhao *et al.*, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," Jun. 21, 2024, *arXiv*: arXiv:2402.19473. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2402.19473
[11] S. Schulhoff *et al.*, "The Prompt Report: A Systematic Survey of Prompting Techniques," Jul. 14, 2024, *arXiv*: arXiv:2406.06608. doi: 10.48550/arXiv.2406.06608.
[12] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," Feb. 05, 2024, *arXiv*: arXiv:2402.07927. doi: 10.48550/arXiv.2402.07927.
[13] V. Venerito, D. Lalwani, S. Del Vescovo, F. Iannone, and L. Gupta, "Prompt engineering: The next big skill in rheumatology research," *International Journal of Rheumatic Diseases*, vol. 27, no. 5, p. e15157, 2024, doi: 10.1111/1756-185X.15157.
[14] Y. Li, "A Practical Survey on Zero-shot Prompt Design for In-context Learning," in *Proceedings of the Conference Recent Advances in Natural Language Processing - Large Language Models for Natural Language Processings*, 2023, pp. 641–647. doi: 10.26615/978-954-452-092-2_069.
[15] C. Toukmaji, "Few-Shot Cross-Lingual Transfer for Prompting Large Language Models in Low-Resource Languages," Mar. 09, 2024, *arXiv*: arXiv:2403.06018. doi: 10.48550/arXiv.2403.06018.
[16] W. X. Zhao *et al.*, "A Survey of Large Language Models," Nov. 24, 2023, *arXiv*: arXiv:2303.18223. doi: 10.48550/arXiv.2303.18223.
[17] Z. Yuan *et al.*, "LLM Inference Unveiled: Survey and Roofline Model Insights," arXiv.org. Accessed: Aug. 26, 2024. [Online]. Available: https://arxiv.org/abs/2402.16363v6
[18] P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, and B. Vidgen, "FinanceBench: A New Benchmark for Financial Question Answering," Nov. 20, 2023, *arXiv*: arXiv:2311.11944. Accessed: Jul. 12, 2024. [Online]. Available: http://arxiv.org/abs/2311.11944
[19] A. J. Yepes, Y. You, J. Milczek, S. Laverde, and R. Li, "Financial Report Chunking for Effective Retrieval Augmented Generation," Mar. 16, 2024, *arXiv*: arXiv:2402.05131. Accessed: Aug. 04, 2024. [Online]. Available: http://arxiv.org/abs/2402.05131
[20] S. Setty, K. Jijo, E. Chung, and N. Vidra, "Improving Retrieval for RAG based Question Answering Models on Financial Documents," Mar. 22, 2024, *arXiv*: arXiv:2404.07221. Accessed: Jul. 01, 2024. [Online]. Available: http://arxiv.org/abs/2404.07221
[21] Z. Nguyen *et al.*, "Enhancing Q&A with Domain-Specific Fine-Tuning and Iterative Reasoning: A Comparative Study," Apr. 19, 2024, *arXiv*: arXiv:2404.11792. Accessed: Aug. 20, 2024. [Online]. Available: http://arxiv.org/abs/2404.11792
[22] B. Peng *et al.*, "Graph Retrieval-Augmented Generation: A Survey," Aug. 15, 2024, *arXiv*: arXiv:2408.08921. doi: 10.48550/arXiv.2408.08921.
[23] T. Procko and O. Ochoa, "Graph Retrieval-Augmented Generation for Large Language Models: A Survey," Jul. 13, 2024, *Rochester, NY*: 4895062. Accessed: Aug. 21, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4895062
[24] V. Karpukhin *et al.*, "Dense Passage Retrieval for Open-Domain Question Answering," Sep. 30, 2020, *arXiv*: arXiv:2004.04906. doi: 10.48550/arXiv.2004.04906.