An Empirical Exploration on Enhancing Financial Insight and E-Commerce Strategy with Advanced RAG-Based Agentic Large Language Models

Journal:	*Transactions on Intelligent Systems and Technology*
Manuscript ID	TIST-2024-02-0155
Manuscript Type:	Special Issue on Evaluations of Large Language Models
Date Submitted by the Author:	29-Feb-2024

Complete List of Authors:	Mantha, Kameshwara Pavan Kumar; Indian Institute of Information Technology Design and Manufacturing Kurnool, Computer Science Engineering Nenavath, Srinivas; IIIT DM Kurnool, CSE Muppalaneni, Naresh Babu; Indian Institute of Information Technology Design and Manufacturing Kurnool, Computer Science and Engineering

Keyword:	Reason and Act (ReAct), Chain of Thoughts (CoT), Tree of Thoughts (ToT), Large Language Models (LLM), Graph of Thoughts (GoT)

An Empirical Exploration on Enhancing Financial Insight and E-Commerce Strategy with Advanced RAG-Based Agentic Large Language Models

M. K. Pavan Kumar[1], Nenavath Srinivas Naik[2], and Muppalaneni Naresh babu[3]

[1,2,3]Department of Computer Science and Engineering, Indian Institute of Information Technology, Design and Manufacturing, Kurnool, India

Abstract

In the dynamic realms of finance and e-commerce, the advent of artificial intelligence marks a pivotal shift towards sophisticated analytical and decision-making capabilities. This paper explores the innovative potential of Advanced Retrieval Augmented Generation (RAG) Based Agentic Large Language Models (LLMs), spotlighting their transformative impact on complex challenges inherent to these fields. Our motivation is driven by the compelling need for advanced computational models that can offer precise financial guidance, analyze intricate market data, and interpret visual sales information with unprecedented accuracy. The problem at the heart of this investigation is the existing AI models' inadequacy in financial question answering (FIQA), Massive Multitask Language Understanding (MMLU), and the analysis of exclusive sales datasets, secured through a partnership with antz.ai. Traditional AI approaches often struggle with deep reasoning, forward-looking insight, and the delicate intricacies of user queries, especially in the context of confidential sales information.

This study meticulously analyzes the application of RAG-based Agentic LLMs, proposing a tailored strategy to overcome these limitations. By fine-tuning the RAG methodology for specific financial datasets and incorporating unique sales and customer feedback data, we introduce an innovative method that significantly improves the model's adaptability and insight into the nuanced e-commerce and financial sectors.Our findings demonstrate that RAG-based Agentic LLMs are exceptionally adept at handling a broad spectrum of financial responsibilities, from providing nuanced investment advice to deciphering market trends across varied data points. This research not only showcases the immense promise of RAG-based Agentic LLMs in reshaping financial analysis and e-commerce but also sets the stage for their subsequent enhancement. The future scope of this work involves further fine-tuning and applying RAG methodologies, aiming to achieve even more refined, reliable, and context-aware AI solutions. By advancing towards more sophisticated, dependable, and context-sensitive implementations, this study contributes to the ongoing evolution of financial analysis and e-commerce, highlighting the indispensable role of AI in mastering the complexities of today's financial and commercial landscapes and opening new pathways for future advancements in these critical areas.

Additional Keywords and Phrases: Reason and Act (ReAct), Large Language Models (LLM), Chain of Thoughts (CoT)

1 Introduction

Large Language Models (LLMs) are at the vanguard of artificial intelligence, leveraging their expansive web-acquired knowledge to edge closer to human-like cognitive abilities. The development of AI agents powered by LLMs[4, 5, 6] marks a significant leap forward, endowing machines with the ability to make decisions that resemble human thought, surpassing the capabilities of previous autonomous systems that were designed for narrow, specific tasks within constrained environments.

These agents powered by LLMs eclipse the functionality of their predecessors, which could be likened to advanced calculators, limited to performing predefined operations. They herald a new era of computational entities capable of not only processing information but understanding, inferring, and

acting upon it by drawing from a vast well of knowledge. This evolution defines these AI agents not merely by their programmed tasks but by their ability to assess situations, formulate plans, and execute these plans with a degree of independence and adaptability previously unseen.

In this context, an "agent" is envisioned as an automated system for reasoning and decision-making that interprets user inputs or inquiries, internally formulating and executing responses to produce precise results. Key attributes of such agents encompass their proficiency in breaking down complex inquiries into manageable segments, selecting and applying external tools judiciously, coordinating a series of actions, and maintaining a record of their endeavors in a memory system. This characterization highlights the complex reasoning, memory, and operational capabilities that LLM-powered agents bring to the table.

The breakthroughs associated with LLM-powered agents, as exemplified by projects like AutoGPT[5] and BabyAGI[33], illustrate their potential to tackle complex problems with little need for human intervention. The architecture of these agents, depicted in Figure 1, merges extensive knowledge bases with logical reasoning and task management skills, opening a new frontier in AI research.

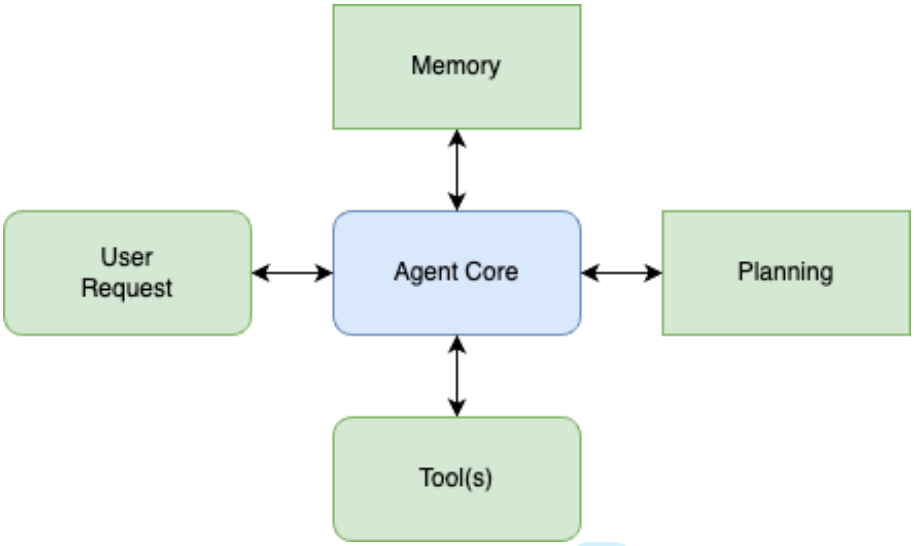

Figure 1: High level Agent Architecture

In the nuanced landscape of LLM-based AI agents, each component plays a sophisticated role that contributes to the agent's overall autonomy and functionality. These components work in concert to simulate a level of understanding and responsiveness that mirrors human interaction within a digital framework.

Memory Module[12]: At the heart of an LLM agent's capability is its memory module, a multi-faceted system that stores and manages the agent's internal logs and user interactions. The memory module comprises both short-term and long-term components. Short-term memory holds the agent's immediate "train of thought," crucial for processing single questions and maintaining the context of ongoing interactions. Long-term memory serves as a logbook of extended interactions, preserving the history of conversations over more extended periods. This memory goes beyond simple retrieval based on semantic similarity; it incorporates a composite score that considers factors such as relevance, importance, and recency, enabling the agent to access the most pertinent information when needed.

Tools[13]: LLM agents employ an arsenal of tools—specialized executable workflows like RAG pipelines for context-aware responses, code interpreters for complex tasks, and APIs for information retrieval or even everyday services. These tools are the means by which agents interact with the digital world, executing tasks and sourcing information to fulfill user requests accurately and efficiently.

Planning Module[16]: The planning module is where the agent's strategy is formulated. It employs techniques like task and question decomposition, as well as reflection or critique, to navigate complex problems. This module enables the agent to plan without feedback, using single-path or multi-path reasoning, and to incorporate feedback from its environment, human users, and its own model to refine its strategies.

Agent Core: The core of an agent is its central command, coordinating the agent's core logic and

behavior. It defines the agent's overarching goals and objectives, catalogues the tools available for task execution, and determines the relevance of memory items based on the user's inquiries. Additionally, the agent core may also contain a persona description, which can guide the agent's preferences in tool usage and infuse its responses with characteristic nuances.

To summarize important contribution of this approach

- Surpasses individual limitations: The synergistic integration of Retrieval-Augmented Generation (RAG) with the fine-tuning of Large Language Models (LLMs) transcends the boundaries of their individual capabilities, presenting a unified approach that adeptly navigates the complexities of data analysis and the nuances of specialized terminology. This collaborative framework not only mitigates the inherent limitations of each technology but also cultivates a more seamless and efficient analytical process.
- Boosts user experience: This harmonious amalgamation significantly enhances the user experience. It goes beyond delivering mere surface-level answers, offering users comprehensive and nuanced analyses meticulously tailored to meet their specific requirements. This level of detail and customization ensures that users are not just recipients of information but are equipped with deep, actionable insights that resonate with their unique contexts.
- Empowers informed decision-making: This integrated approach plays a pivotal role in empowering users towards more informed decision-making. By leveraging the combined strengths of RAG and LLM fine-tuning, it unlocks the full potential of available data, enabling users to base their decisions on a foundation of robust, data-driven insights. This capability is critical in fostering confidence among users, as they navigate through the wealth of information to make choices that are informed, strategic, and aligned with their objectives.

2 Related Work

A substantial amount of research is currently underway to advance Large Language Models (LLMs) as agents. Among these innovative endeavors, Expel[5] stands out as a pioneering learning agent. This agent, rooted in LLM technology, enhances its problem-solving skills by accumulating experiences across various tasks. Distinctively, Expel improves its capabilities not by altering the model's parameters directly but through experiential learning. It has demonstrated superior performance over conventional agents by leveraging learning from training sessions to tackle new evaluation tasks. Moreover, Expel exhibits a remarkable capacity for transferring knowledge across different task domains, offering significant advantages in a range of applications. Throughout its training, Expel has acquired several unforeseen abilities, highlighting the autonomous and human-like learning potential inherent in LLMs.

Another notable contribution is AutoGen[4], an open-source toolkit designed to streamline the creation of conversational agents. This toolkit facilitates effective communication among agents, making it ideally suited for multi-agent collaboration. AutoGen is equipped with a unified conversational interface and automatic response mechanisms, simplifying the coordination between agents. Its versatility is a key advantage, allowing for customization and extension across diverse practical scenarios. AutoGen not only reduces the coding effort required but also enhances the efficiency and performance of agent interactions compared to traditional methods. It supports dynamic, non-linear dialogues among agents, enabling developers to construct more natural and sophisticated multi-agent systems.

The advent of self-teaching language models[7] has dramatically shifted the landscape of artificial intelligence, enabling models to learn autonomously from extensive datasets without explicit supervision. These models now possess the capability to engage in meaningful dialogues and execute tasks well beyond the scope of standard linguistic assessments. This evolution underscores the importance and challenge of accurately evaluating these models' effectiveness. Although traditional benchmarks have been limited to specific tasks, newer benchmarks aim to assess model performance across a broader spectrum of challenges. Nonetheless, many evaluations still fall short of fully examining the models' creative output, conversational endurance, and autonomy in decision-making.

Prior to the emergence of these advanced language models, AI research in linguistics primarily focused on text-based gaming, employing simpler models and learning techniques. The introduction

of more sophisticated language models has ignited a renaissance in AI research, particularly through methodologies that allow models to process sequences of thought before action. Despite these advancements, there remains an acute shortage of diverse and standardized datasets and benchmarks for assessing these models' effectiveness as agents.

Furthermore, as these language models increasingly excel in addressing real-world challenges, there is a burgeoning interest in testing them within environments that necessitate task execution, such as code writing and execution. Initial studies concentrated on the precision of code generation, but recent investigations have broadened to include the models' proficiency in interacting with operating systems and databases over successive conversational exchanges.

3 Anatomy of LLM Agents

The architecture of an AI agent driven by Large Language Models (LLMs), as illustrated in Figure 2, consists of four essential components: Profile, Memory, Planning, and Action. These foundational elements collectively contribute to the agent's framework, enabling it to function as an interactive and intelligent system with the capability for advanced reasoning and making informed decisions. Each component plays a pivotal role in the agent's operational dynamics, working in harmony to facilitate a seamless integration of knowledge processing, strategic planning, and execution. Below is a more detailed exploration of these components

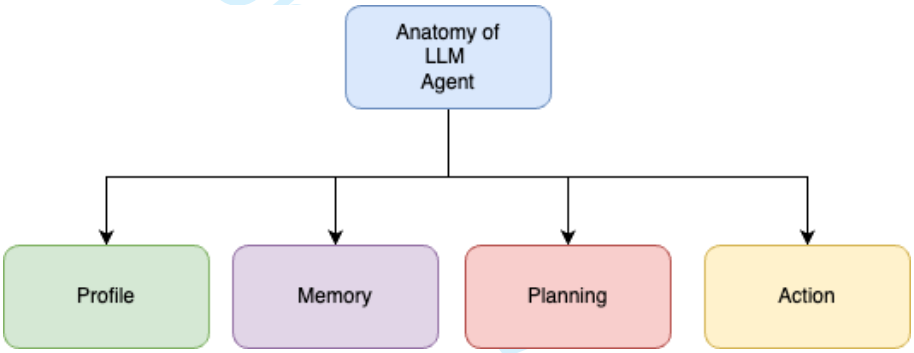

Figure 2: High level Agent Architecture

3.1 Profile

The intricacies of an agent's profile are foundational to its unique identity, weaving together demographic information, personality attributes, and social insights. This comprehensive suite of characteristics is not merely informational but serves as the cornerstone of the agent's interactions with users. It is this rich tapestry of details that informs the agent's approach to communication, influencing both the content and style of its exchanges. The depth and breadth of the profile contents ensure that each agent is distinct, capable of engaging in a manner that is both personalized and reflective of its designed identity.

In parallel, the process through which an agent's profile is generated stands as a critical determinant of its character and interactive capabilities. This process can adopt various forms, from meticulously handcrafted designs that imbue the agent with preconceived traits and tendencies, to more dynamic and fluid methods powered by Large Language Models (LLMs). Automated LLM generation, for example, allows for the derivation of profiles that can evolve and adapt, aligning with the nuances of ongoing interactions and specific datasets. Additionally, aligning the agent's development with targeted datasets can further refine its relevance and effectiveness in specific contexts, ensuring that its persona is both engaging and appropriate for its intended audience.

3.2 Memory

The architecture of an agent's memory plays a pivotal role in its functionality, encompassing both unified and hybrid structures. These configurations facilitate the seamless integration and differentiation

of various types of information, allowing the agent to efficiently organize and access data. The unified structure serves as a comprehensive repository, blending different information streams into a cohesive whole. In contrast, the hybrid structure distinguishes between data types, enabling more specialized processing and retrieval methods tailored to the nature of the information.

Additionally, the versatility of the agent's memory is enhanced by its capacity to encode and retrieve information across a diverse array of formats. From the intricacies of natural languages and structured databases to the straightforwardness of embeddings and lists, the agent's memory system adeptly navigates a wide spectrum of data types and sources. This capability not only deepens the agent's comprehension of varied inputs but also amplifies its interactions with users, facilitating responses that are both rich in context and informed by a comprehensive understanding of the data.

Operational dynamics within the agent's memory, including processes of reading, writing, and reflection, are integral for its continual learning and adaptation. By engaging in memory reading, the agent accesses stored information relevant to current contexts or inquiries. Writing operations allow the agent to assimilate new experiences into its knowledge base, while reflection processes enable the agent to evaluate past actions and outcomes, facilitating the application of learned insights to novel scenarios. These memory operations are essential for the agent's ability to evolve, ensuring that its responses and strategies are informed by an ever-expanding reservoir of knowledge and experience.

3.3 Planning

In the realm of advanced computational agents, the planning mechanisms employed play a crucial role in their operational efficacy and adaptability. These mechanisms can be broadly categorized into two distinct types: planning without external feedback and planning with feedback, each facilitating the agent's decision-making in unique ways.

Planning without external feedback allows the agent to autonomously chart its course of action through either single-path or multi-path reasoning strategies. This autonomous planning capacity enables the agent to navigate through decision-making processes independently, relying on its pre-programmed logic and algorithms. Additionally, the integration of external planning tools can further augment the agent's decision-making capabilities, providing a more nuanced and comprehensive approach to tackling complex problems. This method underscores the agent's ability to operate in environments where immediate feedback is not available, showcasing its self-sufficiency and robustness in decision-making.

Conversely, planning with feedback introduces a dynamic aspect to the decision-making process, wherein the agent actively incorporates input from its surroundings, human interactions, and evaluations of its internal models. This feedback loop enables the agent to continuously refine and adjust its plans based on real-time information and assessments. Such a mechanism ensures that the agent's actions are responsive to the changing environment and tailored to the specific context of each interaction. It highlights the agent's capacity for learning and adaptation, ensuring that its strategies remain aligned with the objectives and expectations of human users and the operational context.

3.4 Action

In the domain of computational agents, the action component is pivotal to their functionality and effectiveness. This component encompasses the objectives that guide the agent's behaviors, the processes through which actions are generated, the operational environment, and the outcomes of these actions.

The agent's objectives, such as accomplishing specific tasks, engaging in exploration, and facilitating communication, serve as the primary drivers of its actions. These goals provide a framework within which the agent prioritizes its activities, ensuring that each action is purposefully directed towards achieving desired outcomes.

The generation of actions is a multifaceted process that relies on the agent's ability to recall previous experiences, adhere to predetermined plans, and, when necessary, adapt its approach in real-time. This ability to improvise is crucial, especially in dynamic environments where unforeseen challenges may arise. By leveraging its memory and planning capabilities, the agent can navigate complex scenarios, demonstrating a level of adaptability and resourcefulness.

The operational environment, or action space, of the agent is defined by the resources available to it, its understanding of itself, and the context within which it operates. This space delineates the

boundaries of possible actions, encompassing the tools at the agent's disposal, its intrinsic capabilities, and the external conditions it encounters. The interplay between these factors determines the range and nature of actions the agent can undertake.

Finally, the impact of the agent's actions is multifaceted, influencing not only the immediate task at hand but also the agent's approach to future challenges. Successful actions can lead to the completion of tasks, the discovery of novel strategies, and alterations in the agent's internal states. These outcomes are integral to the agent's continuous learning process, enabling it to refine its strategies, expand its capabilities, and adapt to new environments and challenges.

4 Topologies of LLM Agents

Large Language Models (LLM) can be powered to create various types of AI agents, each with specific characteristics, profile and capabilities tailored to different applications and tasks. While there isn't a universally standardized classification, here are some commonly recognized types of LLM agents based on their functionality:

- ReAct[7] based agents are a type of AI model that uses a technique known as Reasoning and Acting (ReAct). The core idea of ReAct is to refine the model's responses through iterative self-review. Essentially, the agent attempts to answer a question, then analyzes its own response to identify areas of uncertainty or potential errors. It then revises its answer recursively until it reaches a satisfactory conclusion or a predefined number of iterations is met. This approach is akin to a human double-checking their work for mistakes and making corrections where necessary. ReAct can be particularly useful in tasks where precision is paramount and the cost of error is high.
- CoT[7] based agents employ a method called "Chain of Thought", which is a cognitive approach where the agent simulates a step-by-step reasoning process to reach a conclusion. In this approach, the AI models are designed to mimic human-like reasoning by breaking down complex problems into a series of smaller, more manageable steps, or "thoughts," before arriving at a final answer. This method not only aids in handling multi-step problems but also makes the decision-making process of the AI more interpretable to humans. The transparency of the thought process allows for easier debugging and trust-building, as users can see and understand the logical progression that led to a particular outcome.
- ToT[40] The Tree of Thoughts is an approach where decision-making processes of agents are structured in a hierarchical manner, akin to a branching tree. Each "branch" represents a possible direction of thought, stemming from a central question or problem. This allows the LLM to explore multiple pathways and potential outcomes systematically, much like how a human might approach a complex problem by considering various options and their consequences before arriving at a decision. By implementing a ToT architecture, LLMs can navigate through layers of reasoning, evaluate choices at each juncture, and backtrack when necessary to explore alternative routes. This process can result in a more thorough and nuanced analysis of problems, leading to well-considered and contextually appropriate responses.
- GoT[40]The Graph of Thought offers a non-linear approach to information processing. In a GoT model the agent's knowledge and ideas are represented as nodes within a network, interconnected by edges that signify the relationships and pathways between different concepts. This graph-based structure enables LLMs to traverse through a web of interconnected ideas, allowing for a more fluid and associative style of reasoning. It reflects the multifaceted nature of human cognition, where thoughts are not always sequential but often emerge from a complex network of associations and memories. The GoT framework empowers LLMs to draw upon a broad spectrum of related information, enabling the synthesis of ideas and the generation of creative, informed solutions.

5 Proposed Methodology

For the assessment of the RAG-based model, our study utilized the Financial Question Answering (FIQA) dataset, the Massive Multitasking Language Understanding (MMLU) dataset, and dedicated

proprietary sales dataset. The methodologies employed encompass RAG (Retrieval Augmented Generation) based agentic LLM, depicted in Fig.3 for the extraction of contextual data, along with RAGAs (Retrieval Augmented Generation Assessment)[7], llama-index evaluation strategies, and Phoenix evaluation methods for determining ground truth. This paper provides an in-depth explanation of the procedures undertaken, highlighting two key metrics for evaluation: Faithfulness and Answer Relevance. Notably, these metrics do not necessitate the use of human-annotated datasets or reference responses. Furthermore, the official RAGAs[7] portal introduces additional evaluation metrics, specifically Context Precision and Context Recall. While these were not incorporated into the current research, they present potential avenues for future inquiry and assessment.

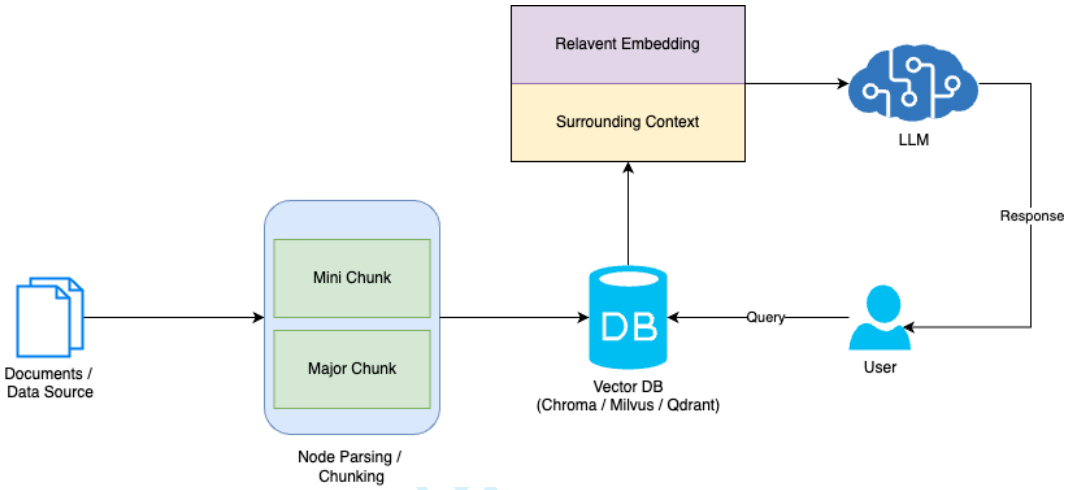

Figure 3: High level RAG Architecture

5.0.1 Retrieval Augmented Generation

The RAG (Retrieval-Augmented Generation)[25, 28] method is an advanced approach that enhances the capabilities of large language models by integrating external information retrieval into the response generation process. This technique starts with an input query, which is then used to search for relevant information across a wide database of texts. The retrieved content serves as a context or foundation upon which the language model constructs its response. By doing so, RAG combines the depth and breadth of external knowledge with the nuanced understanding and generative abilities of language models. This fusion allows for responses that are not only relevant and informed by up-to-date information but also well-articulated and contextually appropriate. The key advantage of this method lies in its ability to dynamically expand the knowledge base of the language model, making it possible to generate responses that are more accurate, detailed, and tailored to the specific query at hand. This makes RAG particularly useful for applications where the quality and informational content of the response are critical.

5.0.2 Retrieval Augmented Generation Assessment

Ragas[7, 8] is dedicated to setting a universal benchmark that empowers developers with the necessary resources and strategies to integrate continuous learning into their Retrieval-Augmented Generation (RAG) applications. This initiative allows developers to create varied synthetic test datasets, crucial for evaluating the effectiveness of their apps. Additionally, Ragas introduces evaluation metrics supported by Large Language Models (LLMs) to offer an objective method for assessing application performance. Beyond development, Ragas provides tools for monitoring app quality in real-world settings using more economical models. These models are capable of delivering actionable feedback, such as detecting inaccuracies in generated responses. By utilizing these insights, developers can continuously refine and enhance their applications, promoting a cycle of ongoing improvement and ensuring their applications remain effective and relevant.

5.0.3 How Ragas work

RAGAs works on the four metrics 1. contextual precision, 2. Contextual Recall, 3. faithfulness, 4. answer relevance the classification of these metrics from the perspective of LLM is shown in Fig.4

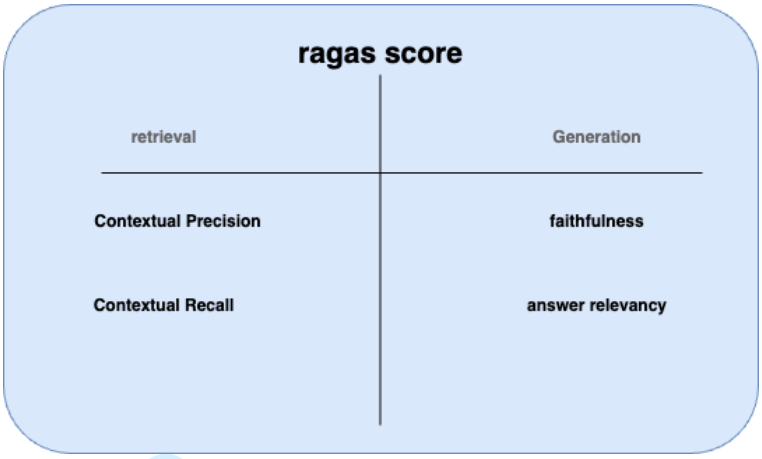

Figure 4: High level RAG Architecture

5.0.4 Faithfulness

This metric refers to evaluating how accurately the information produced in a response aligns with the background information it was based on. The accuracy is quantified on a scale from zero to one, where a higher score indicates better consistency. Essentially, for a response to be considered reliable, every statement it contains must be supported by the background context provided. To assess this, the process involves identifying specific statements within the response and verifying if these can indeed be derived from the context that was initially given. This method ensures that the generated response is not only relevant but also grounded in the provided information, enhancing its reliability and trustworthiness. the score is computed as shown in Equation 1.

$$FaithfulnessScore = \frac{|n|}{|N|} \tag{1}$$

where $|n|$: Number of claims in generated answer that can be inferred from given context and $|N|$: total number of claims in generated answer

5.0.5 Answer Relevance (AR)

This metric evaluates how closely a response generated by a language model matches the original question asked. The higher the score, the more relevant the answer is considered to be. To determine this relevance, the process begins by having the language model propose several possible questions that could arise from the answer it generated. Following this, a text embedding technique is used to transform both these potential questions and the original question into numerical representations, known as embedding. By comparing these embedding, specifically calculating the cosine similarity between the original question and each of the generated questions, it's possible to quantify how well the answer addresses the original query. A closer alignment between these embedding signifies a higher degree of relevance of the answer to the question posed. answer relevance is computed as below. Here $q(i)$ are the final questions and q are the original set of questions.

$$AR = \frac{1}{n} \sum_{i=1}^{n} sim(q, q_i) \tag{2}$$

5.0.6 Context Relevance Score

This metric is used to gauge how effectively the background information gathered aligns with the initial question asked. Essentially, this evaluates if the information fetched is pertinent and supportive of the query. A lower score on this metric suggests that a lot of the information retrieved doesn't contribute to answering the question, potentially leading to a less accurate or relevant response from the language model. To determine this metric, the process involves identifying key sentences within the fetched context through the use of a language model. These sentences are deemed essential as they hold the most value in formulating an accurate answer to the posed question.

$$CRS = \frac{s}{S} \tag{3}$$

where s: number of extracted sentences and S: total number of sentences in context.

6 Experiments

The Algorithm 1 explains the procedure of loading the datasets FIQA[1], MMLU[2] and private sales data one at a time by chunking them with right chunk size and chunk overlap then these chunks are sent to vector database for indexing. The algorithm specially shows the use case of FIQA dataset.

Algorithm 1 LLMAgent - Indexing

$query\_engine \leftarrow None$
$docs \leftarrow []$
$openai\_service\_context = ServiceContext$
$STORAGE\_DIRECTORY \leftarrow path$
Ensure: $FIQA\_dataset \leftarrow data$
  for $ground\_truth \in FIQA\_dataset$ **do**
    for $document \in ground\_truth$ **do**
      if $docuemnt \neq empty$ **then**
        $docs \leftarrow docs + document$
      end if
    end for
  end for
  $index \leftarrow vector\_index(docs)$
  $STORAGE\_DIRECTORY \leftarrow index$
  $output \leftarrow index$

The Algorithm 2 explains the retrieval process of the user query or the question that is asked. First the query is transformed to appropriate embedding vector and then using similarity the context is retrieved from the vector store that is indexed as per the previous algorithm 1. Now the question and the context both are sent to query engine where the agent generate an appropriate answer to the question based on the provided data.

Algorithm 2 LLMAgent - Querying

$storage\_context \leftarrow storage\_context\_index$
$index \leftarrow index\_from\_SC$
Ensure: $retriever = VectorStoreIndex \leftarrow top_k = 1$
$query_engine = response\_synthesizer$
if $query_engine \neq empty$ **then**
  $resp \leftarrow query\_engine.query(row["question"])$
end if
$row["answer"] \leftarrow response.response$
$row["contexts"] \leftarrow response.source\_nodes$

GPT4-turbo powered RAG agent demonstrated high fidelity in its responses, while Mistral powered RAG agent consistently provided answers with high relevance. It's important to note that the llama2

model used in the experiment was an older version; thus, its performance metrics might differ from those of its more recent updates. Additionally, the study is contemplating the adoption of fine-tuning approaches such as LoRa(Low Rank Adaptation)[34], QLoRa(Quantized Low Rank adaptation)[35, 36], and PEFT(Performance Efficient Fine Tuning)[37, 38], which may further influence the outcomes and open for future scope of work

Table 1: Results depicting the scores of answer relevance and faithfulness of randomly selected 30 question on FIQA dataset

FIQA dataset Evaluation Metrics		
Model	Average Faithfulness	Average Answer relevance
GPT-3.5	85	85
GPT-4	90	90
Llama-7b	85	80
Llama-13b	90	85
Mistral-7b	80	80
Falcon-7b	85	80
Gemma-7b	70	80

The table 1 presents a comparative evaluation of various models on the Financial Question Answering (FIQA) dataset, focusing on two critical metrics: Average Faithfulness and Average Answer Relevance. These metrics were derived from a sample of 30 randomly selected questions.

The models in question include several iterations of the Generative Pre-trained Transformer (GPT) series, namely GPT-3.5 and GPT-4, as well as models from the Llama series—Llama-7b and Llama-13b. Additionally, the table features results from the Mistral-7b, Falcon-7b, and Gemma-7b models.

GPT-4 stands out with the highest scores in both evaluated metrics, indicating its superior capability to generate responses that are not only relevant to the questions asked but also faithful to the facts or data. This suggests GPT-4's proficiency in understanding and processing financial information with a high degree of accuracy.

Models from the Llama series and Falcon-7b present moderate scores, suggesting a competent, yet less exceptional performance compared to GPT-4. On the other hand, Mistral-7b and Gemma-7b appear to lag slightly behind the others, with Gemma-7b, in particular, showing the lowest faithfulness score, which implies a need for improvement in generating trustworthy and accurate responses.

Table 2: Results depicting the scores of answer relevance and faithfulness of randomly selected 30 question on MMLU dataset

MMLU dataset Evaluation Metrics		
Model	Average Faithfulness	Average Answer relevance
GPT-3.5	75	80
GPT-4	85	90
Llama-75	70	80
Llama-13b	85	85
Mistral-7b	75	80
Falcon-7b	85	80
Gemma-7b	75	70

The table 2 provides an analytical comparison of various language models based on their performance on the Massive Multitask Language Understanding (MMLU) dataset, with a specific focus on two evaluation metrics: Average Faithfulness and Average Answer Relevance. The data reflects the outcomes from a subset of 30 questions selected at random.

The models evaluated encompass different versions of the Generative Pre-trained Transformer series—GPT-3.5 and GPT-4—as well as a range of models named Llama-75, Llama-13b, Mistral-7b, Falcon-7b, and Gemma-7b.

GPT-4 emerges as the leader in this assessment, securing the highest scores in both Average Faithfulness and Average Answer Relevance. This indicates GPT-4's superior capability in generating responses that are not only factually accurate but also closely aligned with the questions posed. Its

performance suggests a robust understanding of the varied tasks encompassed within the MMLU dataset, setting a benchmark for other models.

Llama-13b and Falcon-7b exhibit commendable performance with balanced scores in both metrics, showing their effectiveness in providing relevant and reliable answers. On the other hand, Llama-75, Mistral-7b, and Gemma-7b reflect moderate performance, with Gemma-7b in particular indicating potential areas for improvement, particularly in terms of faithfulness.

Table 3: Results depicting the scores of answer relevance and faithfulness of randomly selected 30 question on Sales Question and Answer dataset

Private Dataset on Sales Question and Answers		
Model	Average Faithfulness	Average Answer relevance
GPT-3.5	80.3	89
GPT-4	85.6	90
Llama-75	71.4	82
Llama-13b	85	83
Mistral-7b	78	70
Falcon-7b	70.8	89
Gemma-7b	78	77

The table 3 illustrates a comparative analysis of different language models as they perform on a Sales Question and Answer dataset, which is proprietary in nature. The evaluation is centered around two principal metrics: Average Faithfulness and Average Answer Relevance, with results aggregated from 30 questions sampled randomly from the dataset.

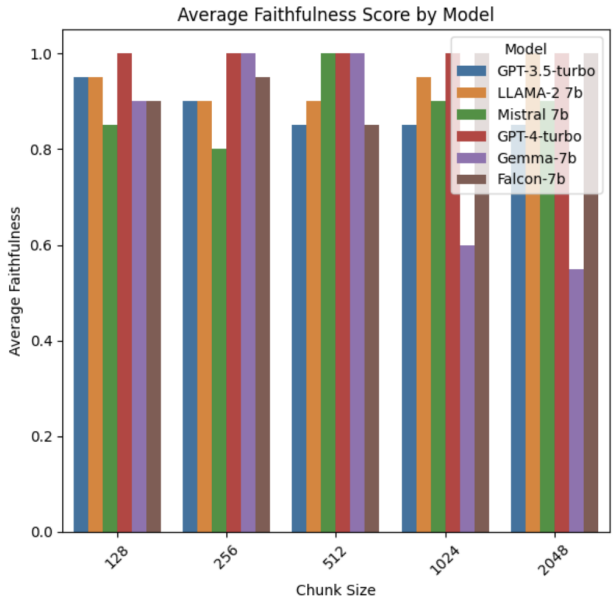

Figure 5: Faithfulness score comparison

In this analysis, the GPT-3.5 and GPT-4 models demonstrate strong performance, with GPT-4 slightly outpacing GPT-3.5, indicating its enhanced ability to provide relevant and faithful answers in the context of sales-related queries. The scores reflect GPT-4's superior understanding and its effectiveness in producing accurate and contextually appropriate responses.

The Llama series models, Llama-75 and Llama-13b, show a varied performance with Llama-13b significantly outperforming Llama-75 in terms of faithfulness to the actual data. Llama-75, while offering a decent level of answer relevance, falls behind in providing responses that are as faithful as those produced by its higher-scoring counterparts.

Mistral-7b and Gemma-7b models present moderate performance in both metrics, with Mistral-7b scoring lower in answer relevance, suggesting a discrepancy between the answers provided and the

questions posed. Falcon-7b, despite a lower average faithfulness score, shows a high level of answer relevance, on par with the leading models, indicating its potential effectiveness in user engagement despite some issues with accuracy.

The Figure 5 in question, labeled as Figure 5, presents a comparative analysis of the Average Faithfulness Scores achieved by various language models. The scores are plotted against different chunk sizes, ranging from 128 to 2048, indicating the amount of text processed by the models in each instance.

The models compared in this visualization include GPT-3.5-turbo, LLAMA-2 7b, Mistral 7b, GPT-4-turbo, Gemma-7b, and Falcon-7b. Each bar in the chart represents the average faithfulness score for a model at a particular chunk size, with faithfulness referring to the degree to which the models' responses are accurate and true to the source data.

From the chart, it can be observed that the models exhibit varying levels of faithfulness across different chunk sizes. It appears that all models maintain relatively high faithfulness scores, staying close to or above the 0.8 mark on the scale, which suggests a strong ability to generate trustworthy content. The chart also allows for the examination of how model performance may fluctuate with the increase in chunk size, providing insights into the scalability and robustness of each model's performance.

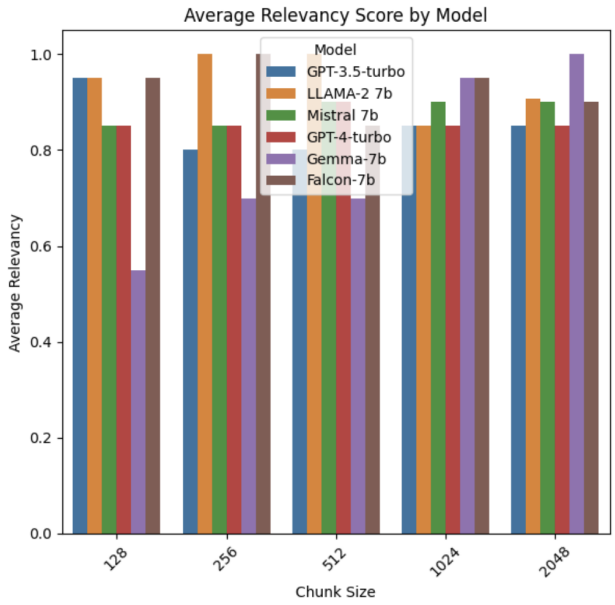

Figure 6: Relevance score comparison

Figure 6 presents an evaluation of various language models in terms of Average Relevancy Score, with the performance of each model assessed at different chunk sizes. The models being compared include GPT-3.5-turbo, LLAMA-2 7b, Mistral 7b, GPT-4-turbo, Gemma-7b, and Falcon-7b.

The relevancy score is indicative of how well the models' responses align with the context and intent of the input data. A higher score on the scale, which peaks at 1.0, suggests a greater alignment between the model's output and the expected response.

From the visualization, it can be discerned that each model's performance varies across the different chunk sizes, which are segments of text that the models process. The bar graph allows for a comparative analysis of the models' ability to maintain context relevance with increasing amounts of information, a crucial factor in applications where precision and context-awareness are of utmost importance.

7 Inferences on the behaviour of LLM Agents

Within the study of advancements brought forth by Large Language Models (LLMs), several key benefits underscore the transformative potential of LLM-powered agents, each contributing to a more secure, flexible, and efficient digital ecosystem as below.[22, 32]

7.1 Advantages

7.1.1 Enhanced Security

One of the paramount advantages is the enhanced security these agents offer. Encapsulated within secure environments and interfacing through protected APIs, these agents significantly reduce the risk of exposure to potential vulnerabilities. The oversight and authentication of their transactions are meticulous, ensuring adherence to stringent security standards. This layer of protection is crucial in maintaining the integrity and confidentiality of data, a cornerstone in the deployment of AI technologies.

7.1.2 Structural Flexibility

the structural flexibility inherent in these agents marks a significant leap forward. With their diverse skill sets, these agents can be seamlessly integrated into existing systems or swapped as needed, catering to the evolving requirements of various applications. This architectural adaptability facilitates the effortless expansion or modification of functionalities, underscoring the versatility of LLM-powered agents.

7.1.3 Dynamic Adaptation

The dynamic nature of these agents allows for real-time adaptation, where their functions and actions can be tailored through specific prompts. This capability for instant customization empowers users to fine-tune the roles and responses of agents according to immediate needs, showcasing an unprecedented level of responsiveness and agility in AI systems.

7.1.4 Increased Efficiency

Efficiency is another critical advantage, as these agents operate autonomously with minimal human oversight, diverging from the more static AI models that require extensive manual intervention. By automating complex processes and decision-making, they streamline operations and alleviate the workload on human resources, thereby enhancing productivity across various domains.

7.1.5 Domain Expertise

In terms of domain expertise, these agents have the capability to develop deep knowledge within specific areas, thanks to targeted training and prompting. This specialization results in a higher level of proficiency and accuracy in their outputs, making them invaluable resources in fields where expert-level understanding is paramount.

7.1.6 Continuous Improvement

The framework for continuous improvement embedded within these systems allows for the ongoing optimization of agent interactions. Through the systematic refinement of prompts, the precision and relevance of the agents' responses are perpetually enhanced, ensuring that outputs remain aligned with the evolving context and requirements.

7.1.7 Data Privacy

The aspect of data privacy is meticulously addressed. Agents are designed to process and act upon data in an abstracted form, ensuring the privacy of user information without compromising the functionality of the agent. This balance between privacy preservation and operational efficiency highlights the thoughtful design and implementation of LLM-powered agents, setting a new standard for AI interactions.

7.2 Limitations of LLM Agents

7.2.1 Long-term Strategy and Problem Breakdown

The challenge of long-term strategy formulation and exhaustive exploration of potential solutions remains a significant hurdle[22, 32]. LLMs lack the robust adaptability of humans, who naturally learn through iterative trial and error and can dynamically adjust their plans in response to unexpected complications. This lack of resilience can be a critical shortcoming, as LLMs may not seamlessly pivot when a chosen plan encounters unforeseen issues.

7.2.2 User Autonomy and Data Security

The control that users exert over the operation of tools by LLMs is limited—they do not decide when or how these tools are activated. This raises significant concerns, especially when utilizing any third-party plugin or tool that hasn't been thoroughly vetted. The risk becomes pronounced if the system has the autonomy to access sensitive data or execute high-level tasks, such as sending emails without direct oversight. Ensuring that third-party (3P) plugins and tools do not act with malicious intent or inadvertently introduce vulnerabilities is a pressing concern.

7.2.3 Tool Efficacy and Contextual Limitations

The finite nature of context length imposes constraints on how much background information, specific instructions, context for API calls, and corresponding responses can be encapsulated. Although methods like self-reflection for learning from previous errors would benefit from extended or unlimited context spans, the system's architecture must operate effectively within the confines of its communicative capacity. While accessing a larger body of knowledge is feasible through vector stores and information retrieval systems, this does not equate to an equally effective representation of knowledge as would be the case with a system that has an undivided and uninterrupted focus.

8 Conclusion and Future Work

8.1 Capability of Hybrid LLM Agents

Envision a virtual assistant that navigates the intricacies of financial data with ease, adeptly analyzing and providing profound insights. This vision is brought closer to reality through the integration of Retrieval-Augmented Generation (RAG) with the fine-tuning of Large Language Models (LLMs), a method poised to revolutionize the analysis of financial inquiries.[8, 10, 17]

Take, for example, a user querying the reasons behind the exclusion of prominent corporations like Apple or Google from the Dow Jones Industrial Average (DJIA) index. A fine-tuned LLM, specialized in financial data, can grasp the fundamental aspects of this inquiry. Its forte is in offering overarching insights based on its extensive knowledge base. Yet, it may not delve into the minutiae, such as identifying subtle trends or revealing concealed patterns within the data.

This scenario underscores the value of RAG[22, 32]. By tapping into a comprehensive external database, RAG can fetch pertinent data points, elucidating Company Z's investment behaviors and financial trajectories, and shedding light on regional specificities and undiscovered insights. Nonetheless, the effectiveness of RAG could be compromised when faced with highly specialized financial terminology, as its retrieval capabilities might falter amidst complex jargon.

The confluence of these technologies—fine-tuned LLMs and RAG—ushers in a new paradigm. Initially, the fine-tuned LLM lays a solid foundation by accurately interpreting the query's purpose and the relevant industry context. Subsequently, RAG expands on this groundwork, diving deep into the dataset to weave a detailed narrative of insights. It elucidates the performance of Company Z across various regions, pinpointing investment patterns, trading anomalies, and trends that might have eluded the LLM.

This collaborative mechanism effectively overcomes their individual limitations. The LLM addresses the nuances of financial terminology, while RAG excels in the extensive retrieval of data. Together, they elevate the user experience, offering not merely a superficial understanding but a comprehensive exploration tailored to the specific intricacies of the query.

This integrative approach extends beyond the realm of financial data analysis. It holds promising applications across diverse fields such as education, research, healthcare, and beyond, wherever in-depth data analysis intersects with specialized terminology. By harnessing the strengths of both RAG and LLM fine-tuning, this strategy inaugurates a new frontier in intelligent data discovery, equipping users with immediate access to insightful knowledge, thereby encapsulating the essence of advanced artificial intelligence exploration in an era characterized by data-driven decision-making.

The implications of this hybrid integration extend across various domain-specific applications of LLMs, such as in the medical, financial, and sales sectors. By analyzing data with a heightened level of precision and depth, this approach not only facilitates the extraction of deeper insights but also significantly improves the accuracy of decision-making processes. As such, the collaboration between RAG and fine-tuning serves as a transformative force, redefining the landscape of domain-specific LLM applications and setting a new standard for intelligent data analysis and interpretation. This advancement heralds a new era of innovation, where the convergence of these technologies paves the way for a future marked by enhanced understanding, strategic clarity, and empowered decision-making across diverse fields.

References

[1] Peiyuan Liu. (2023). MMLU Dataset [Data set]. Kaggle. doi.org:10.34740:KAGGLE:DS:3638509

[2] Macedo Maia, André Freitas, Alexandra Balahur, Siegfried Handschuh., 2018 FIQA Data, ssix-project.eu: 645425

[3] Es, S., James, J., Espinosa-Anke, L. and Schockaert, S., 2023. Ragas: Automated evaluation of retrieval augmented generation. arXiv preprint arXiv:2309.15217.

[4] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X. and Wang, C., 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155.

[5] Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.J. and Huang, G., 2023. Expel: Llm agents are experiential learners. arXiv preprint arXiv:2308.10144.

[6] Kagaya, T., Yuan, T.J., Lou, Y., Karlekar, J., Pranata, S., Kinose, A., Oguri, K., Wick, F. and You, Y., 2024. RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents. arXiv preprint arXiv:2402.03610.

[7] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y. and Zhao, W.X., 2023. A survey on large language model based autonomous agents. arXiv preprint arXiv:2308.11432.

[8] Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K. and Zhang, S., 2023. Agentbench: Evaluating llms as agents. arXiv preprint arXiv:2308.03688.

[9] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X. and Wang, C., 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155.

[10] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E. and Zheng, R., 2023. The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864.

[11] Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.L. and Su, Y., 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2998-3009).

[12] Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F. and Ji, H., 2023. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. arXiv preprint arXiv:2307.05300, 1(2), p.3.

[13] Huang, W., Abbeel, P., Pathak, D. and Mordatch, I., 2022, June. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In International Conference on Machine Learning (pp. 9118-9147). PMLR.

[14] Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D. and Xie, X., 2023. Recommender ai agent: Integrating large language models for interactive recommendations. arXiv preprint arXiv:2308.16505.

[15] Li, G., Hammoud, H., Itani, H., Khizbullin, D. and Ghanem, B., 2024. Camel: Communicative agents for" mind" exploration of large language model society. Advances in Neural Information Processing Systems, 36.

[16] Ruan, J., Chen, Y., Zhang, B., Xu, Z., Bao, T., Du, G., Shi, S., Mao, H., Zeng, X. and Zhao, R., 2023. Tptu: Task planning and tool usage of large language model-based ai agents. arXiv preprint arXiv:2308.03427.

[17] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O. and Zhang, X., 2024. Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680.

[18] Huang, Q., Vora, J., Liang, P. and Leskovec, J., 2023. Benchmarking large language models as ai research agents. arXiv preprint arXiv:2310.03302.

[19] Muthusamy, V., Rizk, Y., Kate, K., Venkateswaran, P., Isahagian, V., Gulati, A. and Dube, P., 2023, December. Towards large language model-based personal agents in the enterprise: Current trends and open problems. In The 2023 Conference on Empirical Methods in Natural Language Processing.

[20] Feldt, R., Kang, S., Yoon, J. and Yoo, S., 2023. Towards Autonomous Testing Agents via Conversational Large Language Models. arXiv preprint arXiv:2306.05152.

[21] Junprung, E., 2023. Exploring the intersection of large language models and agent-based modeling via prompt engineering. arXiv preprint arXiv:2308.07411.

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

[23] Chen, J., Lin, H., Han, X. and Sun, L., 2023. Benchmarking large language models in retrieval-augmented generation. arXiv preprint arXiv:2309.01431.

[24] Kang, M., Gürel, N.M., Yu, N., Song, D. and Li, B., 2024. C-RAG: Certified Generation Risks for Retrieval-Augmented Language Models. arXiv preprint arXiv:2402.03181.

[25] Zhu, Y., Ren, C., Xie, S., Liu, S., Ji, H., Wang, Z., Sun, T., He, L., Li, Z., Zhu, X. and Pan, C., 2024. REALM: RAG-Driven Enhancement of Multimodal Electronic Health Records Analysis via Large Language Models. arXiv preprint arXiv:2402.07016.

[26] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, pp.9459-9474.

[27] Chen, S., Liu, Y., Wu, J. and Hou, M., 2022, December. Retrieval Augmented via Execution Guidance in Open-domain Table QA. In Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence (pp. 1-6).

[28] Sticha, A., 2023. Utilizing Large Language Models for Question Answering in Task-Oriented Dialogues.

[29] Bhayana, R., 2024. Chatbots and Large Language Models in Radiology: A Practical Primer for Clinical and Research Applications. Radiology, 310(1), p.e232756.

[30] Wang, J., Li, J. and Zhao, H., 2023. Self-prompted Chain-of-Thought on Large Language Models for Open-domain Multi-hop Reasoning. arXiv preprint arXiv:2310.13552.

[31] Zheng, H.S., Mishra, S., Chen, X., Cheng, H.T., Chi, E.H., Le, Q.V. and Zhou, D., 2023, October. Step-Back Prompting Enables Reasoning Via Abstraction in Large Language Models. In The Twelfth International Conference on Learning Representations.

[32] Wang, C., Liu, X., Yue, Y., Tang, X., Zhang, T., Jiayang, C., Yao, Y., Gao, W., Hu, X., Qi, Z. and Wang, Y., 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. arXiv preprint arXiv:2310.07521.

[33] Talebirad, Y. and Nadiri, A., 2023. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. arXiv preprint arXiv:2306.03314.

[34] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W., 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

[35] Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L., 2024. Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems, 36.

[36] Zhang, X., Rajabi, N., Duh, K. and Koehn, P., 2023, December. Machine Translation with Large Language Models: Prompting, Few-shot Learning, and Fine-tuning with QLoRA. In Proceedings of the Eighth Conference on Machine Translation (pp. 468-481).

[37] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M. and Gelly, S., 2019, May. Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning (pp. 2790-2799). PMLR.

[38] Lester, B., Al-Rfou, R. and Constant, N., 2021. The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

[39] Shashank.P, antz.ai, Road No. 1, Kothapet, Hyderabad 500035, India.

[40] Besta, M., Memedi, F., Zhang, Z., Gerstenberger, R., Blach, N., Nyczyk, P., Copik, M., Kwaśniewski, G., Müller, J., Gianinazzi, L. and Kubicek, A., 2024. Topologies of Reasoning: Demystifying Chains, Trees, and Graphs of Thoughts. arXiv preprint arXiv:2401.14295.