LLMsResearch on AI Chatbot Inaccuracies, Retrieval-Augmented Generation, and LangChain as a RAG Tool

Juan Carlos Miguel Timoteo
Computer Studies & Engineering
José Rizal University
Mandaluyong City, Philippines
juancarlosmiguel.timoteo@my.jru.edu

***Abstract*—Artificial Intelligence (AI) chatbots have emerged as influential tools for natural language interaction, particularly with the rise of large language models (LLMs) trained on massive corpora. While their fluency and conversational capabilities are remarkable, chatbots often suffer from inaccuracies or "hallucinations," producing outputs that are factually incorrect yet delivered with confidence. This paper reviews the limitations of such chatbots, introduces Retrieval-Augmented Generation (RAG) as a methodological solution, and explores how LangChain serves as a practical framework for building RAG-based applications. The study situates these findings in the larger context of trustworthy AI development, emphasizing the need for credibility, transparency, and modular frameworks when deploying conversational systems across domains.**

***Keywords*—Large Language Models (LLMs), LangChain, Retrieval-Augmented Generation (RAG), Vector Databases, Information Retrieval (IR), Prompt Engineering, Evaluation, Approximate Nearest Neighbor (ANN).**

I. Introduction

The ability of machines to generate natural and coherent human language has rapidly advanced with large language models such as GPT, PaLM, and LLaMA. AI-powered chatbots based on these models now engage in customer support, creative writing, education, and healthcare advice. However, these models are not immune from critical challenges. Among the most troubling is their tendency to fabricate facts, generate outdated or misleading information, and misalign responses with domain-specific expectations. This discrepancy becomes particularly problematic in high-stakes areas such as medicine, law, and academia.

To address these persistent limitations, researchers have proposed Retrieval-Augmented Generation (RAG) as a mechanism to ground LLM outputs in external, verifiable knowledge sources. Instead of relying purely on model memorization, RAG integrates document retrieval with generative capabilities. Tools like LangChain further streamline the construction of RAG systems by connecting models with knowledge bases, retrieval pipelines, and prompt-engineering utilities. This paper explores the trajectory from chatbot inaccuracies to the development of RAG systems, ending with LangChain's role as a flexible solution.

II. The Problem of Inaccuracies in AI Chatbots

LLMs operate primarily as statistical pattern generators trained to predict the most likely next token in a sequence [1]. This probabilistic mechanism accounts for their impressive linguistic creativity but simultaneously generates inaccuracies. These inaccuracies take multiple forms:

A. Hallucinations and Fabricated Facts

One of the most prominent issues is "hallucination," where the chatbot produces plausible-sounding but entirely fabricated information [2]. For example, when asked about a real-life company or scientific finding, a chatbot may invent nonexistent references or merge details from multiple sources incorrectly. Such outputs remain coherent but factually incorrect.

B. Lack of Updated Knowledge

Training on static data means that chatbots often operate with "frozen" knowledge reflective of their pretraining cutoff. In fast-changing domains such as law, medicine, or news, this leads to immediate obsolescence [3].

C. Domain Fragility and Bias

General-purpose chatbots are not optimized for particular domains. When deployed in specialized contexts, they lack depth and accuracy. Further, biases in the training corpus often propagate into outputs, generating cultural inaccuracies, stereotypes, or exclusionary responses [4].

D. Confidence Mismatch

Unlike a human who can hedge their statements with uncertainty markers, chatbots frequently present wrong answers with unfounded certainty. This phenomenon misleads users into trusting incorrect responses—a dangerous situation if the AI is relied upon for decision-making.

Taken together, these issues illustrate why reliance on raw LLM-based chatbots remains insufficient without supplementary mechanisms for accuracy and accountability.

III. Retrieval-Augmented Generation (RAG) as a Solution

To mitigate inaccuracies, Retrieval-Augmented Generation (RAG) has emerged as a promising strategy. Conceptually, RAG adds an external retrieval component to the generative process [5]. Instead of depending solely on model memorization, an LLM enriched with RAG queries a knowledge base in real time, grounding its outputs in retrieved documents.

A. Architecture of RAG

The system combines two main stages. First, a retriever locates relevant documents from a corpus, knowledge graph, or search index using embeddings and similarity search algorithms. Second, the generator (the LLM) produces responses by conditioning on both user input and the retrieved passages [6].

B. Benefits of RAG

- Fact-grounding: RAG reduces hallucinations by forcing the LLM to rely on cited snippets.
- Fresh knowledge: Retrieval modules can query constantly updated databases, ensuring more current answers.
- Domain adaptability: RAG easily customizes to domain-specific repositories, making systems more reliable in specialized contexts such as law or healthcare.
- Explainability: Since retrieved documents can be presented alongside responses, users can verify sources, enhancing trustworthiness.

C. Constraints of RAG

Nevertheless, RAG is not a universal fix. Retrieval quality depends heavily on the underlying indexing method, and poorly structured knowledge bases will undermine accuracy. Moreover, integrating retrieval seamlessly with generation still requires careful system design to avoid irrelevant or redundant outputs.

Despite these challenges, RAG significantly elevates the reliability of AI chatbots and forms the foundation of a new paradigm for practical deployments.

IV. LangChain as a Tool to Build RAG Systems

While the theoretical benefits of RAG are understood, building effective retrieval-to-generation pipelines is technically complex. LangChain has emerged as a leading open-source framework to simplify building such applications [7].

A. Framework Overview

LangChain provides abstractions that connect LLMs with external data. It organizes functionalities into modules, including prompts, memory, retrieval, and chains. Its modularity reduces development friction by enabling developers to integrate diverse backends such as vector databases (Pinecone, Weaviate, FAISS) and different LLM providers.

B. Enabling Retrieval Functions

The framework's retriever module is central to implementing RAG. Developers can specify retrievers connected to document embeddings, ensuring queries search relevant portions of corporate handbooks, academic datasets, or regulatory updates.

C. Chaining and Orchestration

Beyond retrieval, LangChain supports orchestrating multi-step reasoning. For example, a RAG pipeline may involve querying a vector store, combining retrieved documents, compressing context, and then passing the condensed data to the LLM. LangChain's chain abstractions reduce the need for manual orchestration and error-prone coding.

D. Application Domains

LangChain-powered RAG systems are already being applied in question answering, legal research assistance, customer support bots, and academic summarization tools [8]. In each case, the key advantage lies in orchestrating generative power with verifiable retrieval.

Thus, LangChain operationalizes the theoretical promise of RAG, allowing researchers and practitioners to deploy more trustworthy conversational agents.

V. Conclusion

AI chatbots display remarkable language fluency but suffer from accuracy problems due to their reliance on pretraining and probabilistic prediction. These limitations manifest in fabricated information, outdated knowledge, bias transmission, and misleading confidence. Retrieval-Augmented Generation addresses these problems by grounding outputs in external documents, thereby improving factual accuracy, timeliness, and domain applicability. LangChain functions as a powerful framework to translate the concept of RAG into practical systems. By seamlessly integrating LLMs with retrieval pipelines, LangChain ensures that developers can build chatbots that inspire greater trust and credibility across specialized and sensitive domains. Future research must further refine retrieval methods, enhance long-context accuracy, and ensure ethical transparency. Nevertheless, the move from raw LLMs to RAG-powered systems marks a significant progression toward responsible AI deployment.

References

[1] T. B. Brown, et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] S. Ji, T. Yu, C. Xu, and B. Yang, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[3] A. Borji, "A Categorical Archive of ChatGPT Failures," *arXiv preprint arXiv:2302.03494*, 2023.

[4] E. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

[5] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] S. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," *arXiv preprint arXiv:2007.01282*, 2020.

[7] H. Chase, et al., "LangChain: Building Applications with LLMs through Modular Abstractions," *GitHub Repository*, 2023, Available: https://www.langchain.com/.

[8] K. Shuster, et al., "The Limitations of Retrieval-Augmented Generation for Knowledge-Intensive Systems," *arXiv preprint arXiv:2306.01974*, 2023.