Hallucination Detection System For Large Language Models(LLMs) Using GenAi

Keerthi .K[1], Jayashree.S[2], Dharani.R[3], Archana.P[4], Mrs.J. jenila[5]

Abstract- *Large language models (LLMs) often generate plausible yet incorrect information, known as hallucinations. This paper proposes a real-time Hallucination Detection System that evaluates the reliability of LLM outputs. The system combines evidence retrieval from trusted sources, semantic similarity using sentence embeddings, and self-consistency checks across multiple responses. A unified decision module classifies outputs as factual or hallucinated. Implemented as a Streamlit web application, the system provides an intuitive interface for evaluating responses. This approach enhances transparency, reliability, and trust in AI-generated content for research and professional use.*

Keywords: Large Language Models (LLMs), Hallucination Detection, Sentence Embeddings, Evidence Retrieval, Self-Consistency, Fact Verification.

I. INTRODUCTION

This Large Language Models (LLMs) enable powerful text generation but often produce incorrect information, known as hallucinations. This work proposes a Hallucination Detection System that verifies LLM outputs using evidence retrieval, semantic similarity, and self-consistency checks. Implemented as a Streamlit application, it improves the reliability, transparency, and trustworthiness of AI-generated content. Furthermore, the solution is implemented as an interactive web application using Streamlit, allowing users to easily input queries and analyze model responses. This approach not only improves transparency but also builds trust in AI systems by enabling users to validate the information they receive.Overall, this project contributes toward enhancing the reliability, accountability, and practical usability of LLM-based applications in real-world scenarios.

In Introduction about research:

1.1 Background & Motivation:

Large Language Models (LLMs) have revolutionized AI with their ability to generate human-like text and assist in complex decision-making. However, their widespread adoption is threatened by AI "hallucination"—the generation of grammatically correct and plausible content that is factually incorrect or unsupported. This poses severe risks in sensitive fields like healthcare, education, and research.

1.2 The Problem:

LLMs operate on the statistical probabilities of word sequences rather than a genuine understanding of truth. Because manual fact-checking cannot scale with AI output, there is an urgent need for automated, real-time frameworks to evaluate the reliability of generated content.

1.3 Proposed Research & System:

To bridge the gap between AI generation and factual truth, this research proposes an automated Hallucination Detection System leveraging Python and Generative AI. The framework ensures factual accountability through a multi-layered pipeline: Automated Querying: Using advanced LLMs (like Llama) to generate initial responses. External Evidence Retrieval: Fetching real-time data from trusted sources (e.g., Wikipedia) as a factual baseline. Semantic Verification: Using sentence embeddings to mathematically measure similarity between the AI's answer and the retrieved evidence. Self-Consistency Checks: Comparing multiple AI outputs for the same query to test for reliability. Streamlit Web Interface: Packaging the pipeline into an interactive app for easy testing and evaluation.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

Project Title: Hallucination Detection System for LLMs Using Generative AI.Core Objective: To create a real-time system that checks if an AI's answer is factually correct or just a confident-sounding lie (hallucination). Key Tech Stack: Python, Llama (LLM), and Streamlit (for the web interface).

Research Areas : To build this successfully, you will need to research and understand these specific concepts: Retrieval-Augmented Generation (RAG) concepts: How to programmatically query trusted sources like Wikipedia or custom databases to fetch evidence. Sentence Embeddings & Semantic Similarity: Learning how to use

libraries like sentence-transformers in Python to mathematically compare the AI's answer with your fetched evidence .Self-Consistency Techniques: Methods for prompting the LLM multiple times with the same question and evaluating the variance in its answers.

Evidence Fetcher: A Python module using APIs (like the Wikipedia API) or web scraping to grab source truths based on the user's prompt. The Scorer/Decision Module: An algorithm that takes the similarity score and the self-consistency score and calculates a final "Trust/Hallucination" percentage. Streamlit UI Elements: A clean layout featuring a text box for the prompt, a loading spinner while it checks facts, and a color-coded output (e.g., Green for Factual, Red for Hallucinated).

III. CORE STUDIES AND FINDINGS

Now The "Black-Box" Reality: Since leading models are often closed-source, internal probability scores are hard to access. Therefore, "black-box" post-processing methods (like yours) are the most practical and widely adopted industry standards. The Power of RAG: Retrieval-Augmented Generation—fetching live data from trusted sources like Wikipedia—is proven to be the single most effective way to anchor an LLM in reality. The Multi-Step Advantage: Studies show that combining multiple detection methods yields the highest accuracy. Relying on just semantic similarity or just self-consistency leaves blind spots.

How Your Proposed System Solves This: Your abstract outlined a highly effective, modern architecture that directly utilizes these findings by splitting the task into three logical steps: Retrieval (Factual Grounding): Using Python to pull trusted reference text to act as the "source of truth." Cross-Examination (Self-Consistency): Prompting the Llama model multiple times to see if it confidently repeats the same answer or wavers (wavering indicates a high chance of hallucination). Mathematical Scoring (Semantic Similarity): Using sentence embeddings to calculate a hard mathematical score of how closely the AI's generated response actually matches the trusted retrievedevidence.

IV. GET PEER REVIEWED

Strengths: Hybrid Approach: Relying on a single method for hallucination detection usually fails. Combining RAG (external evidence), Self-Consistency, and Semantic Similarity is considered an industry best practice. Practicality: Using Streamlit for a real-time UI proves the project isn't just a theoretical paper but a functional tool. Open Source Focus: Grounding the research in Python and the Llama model ensures accessibility and reproducibility for other researchers.

Critical Questions : An expert reviewer would likely ask you to clarify or defend the following points in your final paper: The "Golden Truth" Problem: If you are querying an external source (like Wikipedia) to check the LLM's answer, how do you ensure the fetched source itself is 100% correct and relevant? If the search returns a bad snippet, your scoring will fail. Computational Cost (Latency): Running Self-Consistency requires prompting an LLM 3 to 5 separate times for a single question. A reviewer will ask: “Is this system too slow or expensive for real-time production use?” Benchmark Metrics: How will you prove your system actually works? Reviewers will expect you to test your system against established hallucination datasets and provide your Precision, Recall ,and F1-Scores.

To make your project bulletproof for a college presentation or an actual publication, consider adding these elements: Acknowledge RAG Limitations: State in your paper that your system relies on the assumption that the retrieved documents are accurate, or implement a secondary check for search result relevance. Define

Your Thresholds: Explain how you calculated the "Trust/Hallucination" percentage. Did you use standard Cosine Similarity? What score constitutes a fail (e.g., below 0.7)? Propose a "Fallback" State: Explain what happens when a hallucination is detected. Does the system just throw an error, or does it try to rewrite the answer using only the fetched evidence?

V. IMPROVEMENT AS PER REVIEWER COMMENTS

1. Solving the "Golden Truth" Problem (Source Verification):

Critique: How do you ensure the fetchedexternal evidence (like Wikipedia) is actually correct and hasn't led the system astray? Improvement: We will implement a Contextual Relevancy ranker before the scoring phase. Instead of trusting the fetched data blindly, the system will use a smaller, faster cross-encoder model to score the retrieved snippets against the user's prompt. Snippets falling below a certain semantic threshold will be discarded to ensure the "source of truth" is actually relevant.

2.Solving Computational Latency:

Critique: Running an LLM multiple times for self-consistency is too slow for a real-time Streamlit app.Improvement: To reduce latency, we will shift from full-text self-consistency to

Token-Level Probability or a light LLM-as-a-Judge prompt. Instead of generating 5 separate full answers, we will ask the Llama model once to generate the answer and a small secondary prompt to score its own confidence. This reduces API/computation calls from 5x to 2x, making the Streamlit UI much faster.

3. Adding Standardized Benchmarks:

Critique: How do we prove mathematically that this system actually works? Improvement: We will evaluate our system's precision and recall using established public datasets specifically designed for this, such as HaluEval or Vectara's HHEM (Hallucination Evaluation Model). By testing our system against hundreds of known hallucinated vs. factual AI responses, we can publish hard percentage metrics in our final report.

VI. CONCLUSION

This project presents a real-time Hallucination Detection System for LLMs, combining evidence retrieval, semantic verification, and self-consistency checks. The system enhances the reliability, trustworthiness and accountability of AI-generated content, making LLMs safer for research, education, and professional use.

VII. ACKNOWLEDGMENT

We express our sincere gratitude to our project guide, Mrs. J. Jenila, for her invaluable guidance, continuous support, And insightful feedback throughout the development of this paper. Her expertise was vital in shaping our research on hallucination detection systems. We also extend our heartfelt thanks to the Department of Computer Science and Engineering for providing The necessary resources and environment to conduct this research. Finally, we thank our peers and families for their constant encouragement.

REFERENCES

[1] Alansari and M.Luqman, survey of hallucination in LLM, 2025
[2] D.Anh-hoang, V.Tran,Truthful evaluation in LLM, 2025
[3] GueJ.Chen, Discrimination modeling for hallucination detection in LLMs, 2024
[4] Y.Kang, Uncertainty quantification for hallucination detection in LLMs, 2025