

GUARDRAIL: HALLUCINATION DETECTION AND PREVENTION IN LLMS

Presented by:

Sreenidhi Hayagreevan
Shivani Atul Beri
Laxmi Thrishitha Kalvakota
Bhavana Meravanige Veerappa

Project Advisors:

Dr. Simon Shim
Dr. Lee C. Chang

Introduction

Large Language Models (LLMs) are powerful but often generate hallucinated, unsafe, or policy-violating responses, making them unreliable for real-world applications. They also remain vulnerable to prompt injection and adversarial inputs, with limited mechanisms for enforcing safety in real time.

To address this, we developed GuardRail for LLM, a modular safety framework that acts as an intermediate layer between users and the model. The system incorporates input validation, prompt control, and output filtering to detect and block harmful or irrelevant content.



By enforcing structured constraints at both the input and output stages, GuardRail significantly improves response reliability, safety, and consistency, enabling more trustworthy deployment of LLM-based applications.

Methodology

We implemented multi-layer guardrail system combining machine learning, rule-based filtering, and retrieval-augmented validation (RAG) to enforce safety. A knowledge base (Wikipedia + domain text) was built, preprocessed and chunked (~600 chars), and embeddings were generated using SentenceTransformer for semantic search.

For input safety, we trained a binary classifier (Logistic Regression) on text embeddings to classify safe vs unsafe prompts. The model is stored as a `joblib` for real-time inference and captures contextual unsafe intent beyond keyword matching. This is complemented by regex filters for PII, hate speech, bias, violence, and fraud. The RAG pipeline retrieves relevant chunks to provide context grounding and reduce hallucinations. The LLM (Ollama) processes only validated inputs using retrieved context and structured prompts.

A post-processing layer applies output guardrails, including hallucination and PII detection. Unverifiable responses are replaced with a safe fallback. The system is optimized using a shared embedding model, similarity threshold (~0.3), background threading, and modular middleware design.



Analysis and Results

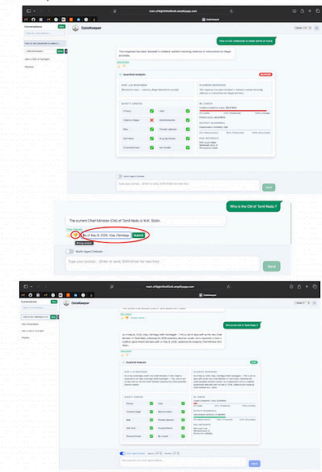
The GuardRail system was evaluated across safety classification, hallucination detection, and end-to-end system validation using benchmark datasets and real-time deployment tests.

For input safety classification, multiple models were compared. Logistic Regression achieved 97.79% accuracy with high recall (0.8378), ensuring most unsafe prompts are detected, though with some false positives. Random Forest achieved the best overall performance (99.47% accuracy, F1 = 0.8183), providing a balanced trade-off between precision and recall, while Decision Tree showed moderate performance.

The system was validated using 7 functional test cases (unsafe prompts, hallucination correction, bias detection, privacy protection), all resulting in PASS, confirming reliable end-to-end behavior.

The GateKeeper UI provides full transparency of guardrail decisions. As shown, unsafe prompts (e.g., harmful queries) are blocked at input stage, with the interface displaying:

- Raw LLM response (blocked before execution)
 - Guarded response (policy-compliant fallback)
 - Safety checks across categories (PII, bias, violence, etc.)
 - ML unsafe probability score (~96%)
 - RAG metadata and hallucination score
- This makes the system explainable and auditable, unlike black-box LLM outputs.



The RAG-based knowledge layer improves factual reliability by retrieving context from structured datasets and Wikipedia. Incorrect LLM outputs are corrected using retrieved evidence, reducing hallucinations and ensuring grounded responses.

The output guardrail layer further validates responses using context similarity and PII detection, blocking or replacing unsafe outputs with safe fallbacks.

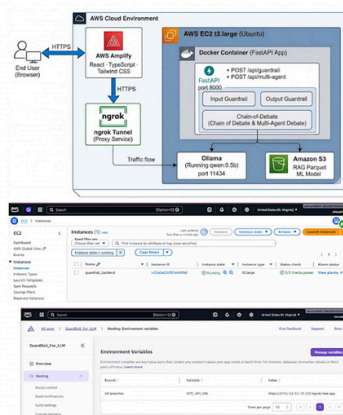
Overall, the system achieves:

- High recall in unsafe prompt detection
- Effective hallucination reduction via RAG grounding
- End-to-end safety enforcement (input + output layers)
- Explainable decisions through interactive UI



Cloud Deployment

The system is deployed as a full-stack cloud application. The frontend (React + TypeScript) is hosted on AWS Amplify, while the backend (FastAPI) runs on an AWS EC2 instance with Dockerized services. The LLMs are served locally via Ollama, and datasets, embeddings, and model artifacts are stored in Amazon S3. API communication is secured via HTTPS (ngrok tunneling), enabling a scalable, modular, and real-time guardrail system suitable for enterprise deployment.



Summary/Conclusions

This project presents a multi-layer GuardRail framework for Large Language Models that integrates input validation, RAG-based grounding, multi-model reasoning, and output verification to address key challenges such as unsafe content generation and hallucinations.

The system combines ML-based safety classification, rule-based filtering, and retrieval-augmented validation to enforce policy compliance and factual accuracy. Evaluation across benchmark datasets and real-time test cases shows that the framework effectively blocks unsafe prompts, reduces hallucinations, and produces context-grounded responses while maintaining low latency.

The GateKeeper UI enhances transparency by exposing guardrail decisions, including safety checks, ML scores, and response validation, enabling explainable and auditable AI behavior.

This project is beneficial as it provides a reliable safety layer for deploying LLMs in real-world applications, reducing risks associated with harmful content, misinformation, and bias. By ensuring trustworthy, verifiable, and policy-compliant outputs, the system supports safer adoption of AI in high-stakes domains such as healthcare, finance, and legal systems, where accuracy and accountability are critical.

Overall, the framework offers a scalable, modular, and enterprise-ready solution that advances LLM reliability by combining detection, grounding, and verification into a unified pipeline.

Key References

- [1] Lin, S., Hilton, J., & Evans, O. (2021). TruthfulQA: Measuring How Models Mimic Human Falsehoods. ACL. <https://arxiv.org/abs/2109.07958>
- [2] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS. <https://arxiv.org/abs/2005.11401>
- [3] Rajpurkar, P., et al. (2016). SQuAD: 100,000+ Questions for Machine Comprehension. EMNLP. <https://rajpurkar.github.io/SQuAD-explorer/>
- [4] Bowman, S., et al. (2015). A Large Annotated Corpus for Learning Natural Language Inference (SNLI). EMNLP. <https://nlp.stanford.edu/projects/snli/>
- [5] Williams, A., et al. (2018). MultiNLI: A Multi-Genre Natural Language Inference Corpus. NAACL. <https://cims.nyu.edu/~showman/multinli/>
- [6] Davidson, T., et al. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. ICWSM. <https://arxiv.org/abs/1703.04009>

Acknowledgements

We express our gratitude to our project advisors, Dr. Simon Shim, for guidance throughout this project. We thank the Department of Applied Data Science, SJSU, for providing all academic resources. We acknowledge the open-source community (Hugging Face, PyTorch, Ollama, Apache Airflow, scikit-learn) and the benchmark datasets used, including TruthfulQA, SQuAD, Safety Prompts, Hate Speech, and NLI datasets.

Interested in partnering with us on a capstone project or exploring other ways to collaborate with the College of IDS? Contact CIDS_CAP@sjsu.edu to discuss potential projects and partnership opportunities.