RAG in Cybersecurity: A Next-Gen AI Pipeline

MARTIN HOLGUIN, JAIME ANDRADE, NATALIE VALENZUELA, MD NURUZZAMAN SOJIB, DR. ARITRAN PIPLAI
NSF RET Site: Cybersecurity Research Experience For Educators Through Data Science (CREEDS)
The University of Texas at El Paso

Abstract

This project investigates the use of foundation models and AI agents in cybersecurity, focusing on adaptability and confidence in decision-making. Unlike static traditional models, foundation models learn from few-shot examples, self-reflect, and improve continuously, enabling broad task generalization. A key method is Retrieval Augmented Generation (RAG), which integrates external knowledge to enhance relevance and adaptability. In a six-week Research Experience for Teachers (RET), training a system that parses diverse data sources, builds vector databases, and uses RAG pipelines to refine retrieval quality was implemented. To assess response certainty, confidence measures were introduced such as entropy-based calculations during retrieval and generation. When uncertainty is high, the system gathered additional data. Our goal is to build AI agents that deliver optimal cybersecurity responses and autonomously seek better solutions when needed.

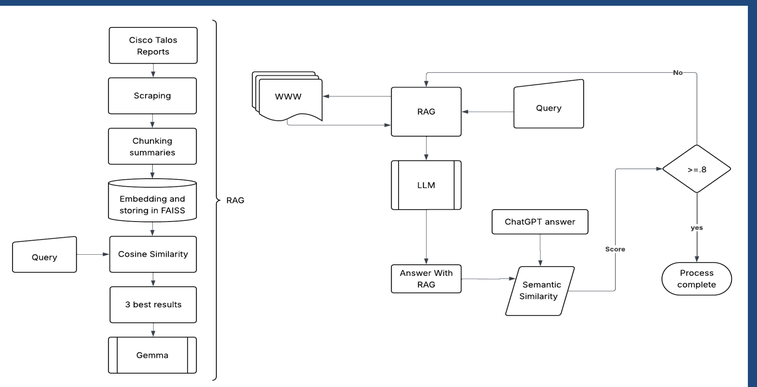
Introduction

Large Language Models (LLMs) have transformed natural language processing but are limited to their training data, making them less effective in fast-evolving fields like cybersecurity. To address this, we use Retrieval Augmented Generation (RAG)—a method that pulls relevant, up-to-date information from external sources and feeds it into the LLM. This integration allows LLMs to better recognize and respond to emerging cyber threats.

We evaluate whether RAG improves response quality by comparing LLM outputs with and without retrieved context. This approach enhances threat detection by combining historical data, recent developments, and LLM reasoning to help secure systems more effectively.

Ultimately, combining LLMs with RAG enhances our access to information related to cyberthreats and protections for critical systems more effectively.

Process



Methodology

1 Data Collection & Preprocessing
- Used Python's requests library to extract raw HTML content from web pages.
- Parsed and cleaned text using BeautifulSoup.
- Converted clean text into vector embeddings using LangChain.
- Stored vectors in a FAISS vector database in manageable chunks.

2 Question Answering with RAG
- Incorporated a user question and vectorized it.
- Retrieved the top 3 most relevant chunks using cosine similarity.
- Provided retrieved chunks as context to an LLM (Gemma or LLaMA).
- Generated answers with (RAG) and without context (non-RAG).

3. Evaluation & Confidence Scoring
- Compared each generated answer to a reference answer using a semantic similarity metric.
- If similarity ≥ 0.80 → accepted; if < 0.80 → retrieved more context and repeated the process.
- The system determined confidence levels based on entropy scores.

Results and Discussion

The experiment shows that both models performed better using RAG. (Figure 1)

Though one model performed better, the data shows that confidence increases with RAG. (Figure 3) A lower entropy score indicates better alignment with the question.

Figure 2 compares the distribution of similarity scores using RAG for Llama and Gemma.

Figure 1

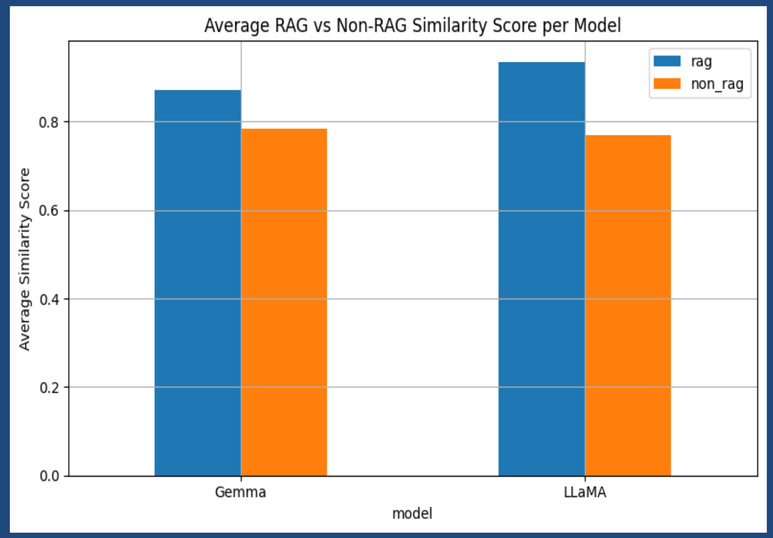


Figure 2

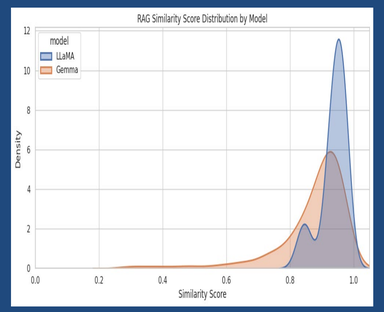
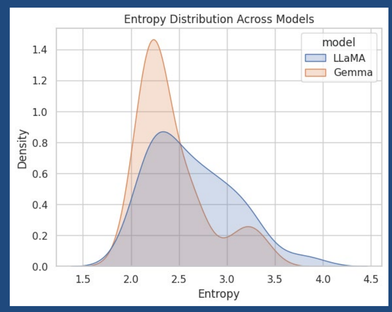


Figure 3



Potential Lesson Plan Elements

Web Scraping in Python (TEKS D(8–11))
- Objective: Scrape content using requests & BeautifulSoup in Google Colab.
- Security Focus: Apply secure scraping strategies—respect robots.txt, handle rate limits, and avoid IP rate-limits.
- TEKS Integration: Discuss system vulnerabilities via insecure scraping; link to CIA triad (d(8.c)) and network/device issues (d(11)).

Technology Operations and Concepts (TEKS Q)
- Objective: Add context to LLM searches to enhance accuracy and reliability of responses. Understand basic value of RAG system.
- AI Focus: Introduce students to RAG and ChatBots.
- TEKS Integration: demonstrate an understanding of artificial intelligence and implement artificial intelligence (q).

Conclusion

Future implementation of Retrieval-Augmented Generation (RAG) with LLMs should include model-specific tuning to maximize the benefits

Overall, models demonstrated enhanced performance with RAG integration, though results varied depending on the architecture and input complexity

These findings support the potential of RAG to improve LLM output quality when properly aligned with the task and model configuration.

References

"Vulnerability Reports || Cisco Talos Intelligence Group - Comprehensive Threat Intelligence." *Talosintelligence.com*, talosintelligence.com/vulnerability_reports.

Upadhyay, Akriti. "Efficient Information Retrieval with RAG Workflow - Akriti Upadhyay - Medium." *Medium*, Medium, 9 Oct. 2023, medium.com/@akriti.upadhyay/efficient-information-retrieval-with-rag-workflow-afdfc2619171.

Tuckfield, Bradford. *Dive into Algorithms : A Pythonic Adventure for the Intrepid Beginner*. San Francisco, Ca, No Starch Press, 2021.

V Anton Spraul. *Think like a Programmer : A Beginner's Guide to Programming and Problem Solving*. San Francisco, No Starch Press, 2017.

Acknowledgements

Special thanks to Md Nuruzzaman Sojib, Candidate for PhD in Computer Science, Teaching Assistant, University of Texas at El Paso

This work is supported by National Science Foundation Award # 2206982