Real or Hallucinated? Improving the Truthfulness of LLMs

Salil Fernandes, Shruti Pareshbhai Gandhi, Xenus Gonsalves, Vedant Hareshbhai Patel, Reuben Suju Varghese
[salilfer, gandhisp, xgonsalv, patelved, rsvarghe] @usc.edu
Group 40
University of Southern California

Abstract

This paper focuses on solving the issue of hallucinations in Large Language Models (LLMs) like Gemini, Qwen, Mistral, Bloom and LLaMA—where models generate factually incorrect or misleading responses. We develop a multi-stage detection pipeline combining data preprocessing, Natural Language Inference (NLI), vagueness detection, fact-checking using Wikipedia, and entropy-based uncertainty estimation. On top of this, we then reduce hallucinations with Retrieval Augmented Generation (RAG) by conditioning responses on verified knowledge. This hybrid approach significantly improves the factual accuracy and robustness of the generated outputs with respect to multiple LLMs.

1 Introduction

Hallucinations in Large Language Models (LLMs) refer to occurrences where the model produces inaccurate, deceptive, or invented information with great certainty. Our project aims to detect and mitigate these hallucinations by leveraging textual inference, vagueness detection, and fact-checking methods across multiple LLM outputs. It focus on hallucinations that occur in current Large Language Models (LLMs), including Gemini, Qwen, Mistral, Bloom and LLaMA.

These hallucinations appear mainly when users ask complex logical questions, request information about current events, specialized topics, or when they ask questions with ambiguous interpretations.

Examples include:

- How many “r”s are there in “strawberry”?
- Cadmium Chloride is slightly soluble in this chemical, it is also called what?
- I get out on the top floor (third floor) at street level. How many stories is the building above the ground?

The removal of hallucinations is essential as they affect the trust, dependability, and safety of AI-generated material. This raises a fundamental question: Should we trust LLMs blindly when they struggle with basic reasoning, such as counting letters in a word? Is hallucination inevitable?

To enhance our understanding and address these problems, we investigate different hallucination detection and mitigation methods, including Natural Language Inference (NLI) with Wikipedia fact checking and entropy-based uncertainty scoring.

We also introduce a hallucination reduction pipeline using Retrieval-Augmented Generation (RAG), where external factual knowledge is incorporated into the model’s response generation. Through this multi-faceted approach, we aim to evaluate and enhance LLM output factual accuracy in real-world scenarios.

2 Related Work

We expand the research on LLM hallucination detection and mitigation through the utilization of responses derived from five different models in the HaluEval QA dataset (Li, 2023) which received binary `is_hallucinated` annotations. The evaluation follows benchmark standards adopted by projects including HaluEval and FELM (Gunasekar, 2023) because they implement standardized hallucination detection methods.

We used Retrieval-Augmented Generation (RAG) as a hallucination reduction method because it follows suggested techniques Ram (2023) for integrating external knowledge. We designed a vagueness detection method through the use of carefully selected abstention phrases that originated from the requirement of both faithful and uncertain text generation (Maynez, 2020). The system included Wikipedia-based fact-checking and Natural Language Inference (NLI) as methods to determine answer consistency following recent research on truthfulness verification (Cheng, 2023).

3 Problem Description

Gemini, Mistral, Qwen, Bloom and LLaMA produce hallucinations through the release of factually incorrect or misleading responses at high confidence levels. The hallucinations frequently emerge during complex reasoning tasks or when processing current events and ambiguous requests thus jeopardizing AI system reliability. Our research tackles the issue through an approach that combines Natural Language Inference (NLI) with Wikipedia fact checking, vagueness detection, fact-checking and Retrieval-Augmented Generation (RAG) for detecting and minimizing hallucinations.

4 Methodology

4.1 Dataset Creation

We use the QA split of the HaluEval benchmark (Minervini, 2023), as the base for our dataset. For each question, we generated five distinct answers using different large language models (LLMs): Qwen (via Groq API) (Groq, 2025), LLaMA, Mistral, and Gemma (via Together API) (Together AI, 2025), and Bloom.

The binary `is_hallucinated` annotation was calculated through a custom similarity-based function which assessed each generated answer. The function applies `all-MiniLM-L6-v2` from SentenceTransformers library as an embedding model to analyze the similarity between the gold answer and the text produced by the model through cosine similarity analysis. It operates at multiple granularities—full text, sentence, and chunk level—flagging an answer as hallucinated if the maximum similarity score falls below a defined threshold (set to 0.7). This allowed for a scalable and consistent annotation pipeline across all model outputs.

4.2 Hallucination Detection

4.2.1 Preprocessing

In order to ensure consistency, we cleaned the raw LLM responses for each prompt in the dataset. This cleaner function removed markdown, AI disclaimers, boilerplate code, templated phrases, HTML tags, and normalized whitespaces. We also created a predefined list of vague phrases (e.g. “I don’t know”, “Not enough context”, “I do not have enough information”) which was used to filter out evasive responses given by the LLM.

4.2.2 Detection via NLI

The first stage in our detection pipeline performs textual inference. For this purpose, we used Facebook’s RoBERTa-large-MNLI transformer model offered by HuggingFace to perform natural language inference (NLI). Each data sample was transformed into a premise-hypothesis pair format: premise [SEPARATOR] hypothesis.

If the transformer gives a classification label of "contradiction" with a confidence/BERT score above the threshold of 51% or if the LLM response did not pass the vagueness test via a rule based approach from the list of vague phrases, the response was treated as a signal of a hallucination. To make the classification process faster, we processed the data in batches using the HuggingFace pipeline and datasets library for scalability.

4.2.3 Factual Verification

All the misclassifications from NLI i.e. the false positives (FP) and false negatives (FN) were fed into the fact checking stage of the detection pipeline. We made use of the Wikipedia Python API for this stage. For each misclassified sample in the dataset, we extracted the key sentence from the LLM output which most likely contained the LLM’s intended answer and fetched pages relevant to the question from Wikipedia. Now taking the summary of the top three pages as the premise and keeping the LLM response as the hypothesis, we fed the new premise-hypothesis pairs into RoBERTa for reclassification. If the output label was “contradiction” or if it was “neutral” but with a low confidence score ($< 50\%$), the response was flagged as a hallucination.

4.2.4 Entropy-Based Reclassification

At this point, we’ve filtered out most hallucinations using inference and fact validation. However, LLMs may still produce confident-sounding hallucinations, so we bring in uncertainty entropy-based modeling. The residual misclassifications from the fact checking stage were fed into the entropy scoring module. We computed token-level entropy from the GPT-2 tokenizer over each response. Entropy here simply measures uncertainty in the model’s next-word prediction. If the average entropy of a response exceeds a threshold (> 4 nats), the response is reclassified as hallucinated.

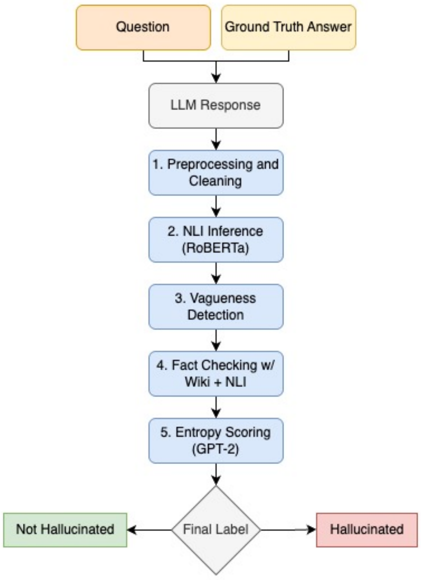
Question
Ground Truth Answer
LLM Response
1. Preprocessing and Cleaning
2. NLI Inference (RoBERTa)
3. Vagueness Detection
4. Fact Checking w/ Wiki + NLI
5. Entropy Scoring (GPT-2)
Not Hallucinated
Final Label
Hallucinated

Figure 1: Detection Pipeline

4.3 RAG Pipeline

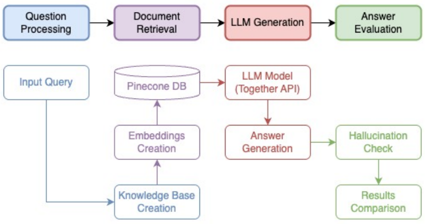

Figure 2: RAG Pipeline

To increase answer accuracy, the Retrieval-Augmented Generation (RAG) process grounds LLM responses on relevant external documents, as shown in Figure 2. An input query is used at the beginning of the process to search a vector database (Pinecone) that contains a pre-calculated knowledge base. Effective semantic search is made possible in this knowledge base by embedding a well chosen set of texts using sentence-level transformers. The top-matching documents, which are obtained based on their similarity to the query, are combined with the original question to form a context-rich prompt. This prompt is passed to a large language model (via the Together API), which generates a response using both the input query and the retrieved information. To assess the factual consistency of the output, a hallucination detection step compares the generated answer with known ground-truth references. By anchoring the model's output in real evidence, this pipeline significantly reduces hallucination rates while preserving the generative strengths of the LLM.

5 Experimental Results

5.1 Detection Metrics Results

	Precision	Recall	F1-score	Support
False	0.68	0.69	0.68	3337
True	0.84	0.84	0.84	6663
Accuracy			0.79	10000
Macro Avg	0.76	0.76	0.76	10000
Weighted Avg	0.79	0.79	0.79	10000

Table 1: LLAMA Metrics (Post NLI)

	Precision	Recall	F1-score	Support
False	0.81	0.84	0.82	3337
True	0.87	0.86	0.91	6663
Accuracy			0.88	10000
Macro Avg	0.86	0.87	0.87	10000
Weighted Avg	0.88	0.88	0.88	10000

Table 2: LLAMA Metrics (Post Fact Check + Entropy)

	Precision	Recall	F1-score	Support
False	0.68	0.92	0.78	5268
True	0.85	0.52	0.64	4732
Accuracy			0.73	10000
Macro Avg	0.76	0.72	0.71	10000
Weighted Avg	0.76	0.73	0.71	10000

Table 3: Gemma Metrics (Post NLI)

	Precision	Recall	F1-score	Support
False	0.77	0.98	0.86	5268
True	0.97	0.68	0.80	4732
Accuracy			0.84	10000
Macro Avg	0.87	0.83	0.83	10000
Weighted Avg	0.87	0.84	0.83	10000

Table 4: Gemma Metrics (Post Fact Check + Entropy)

	Precision	Recall	F1-score	Support
False	0.74	0.83	0.78	6790
True	0.41	0.29	0.34	2728
Accuracy			0.67	9518
Macro Avg	0.57	0.56	0.56	9518
Weighted Avg	0.65	0.67	0.66	9518

Table 5: QWEN Metrics (Post NLI)

	Precision	Recall	F1-score	Support
False	0.81	0.98	0.89	6790
True	0.89	0.42	0.57	2728
Accuracy			0.82	9518
Macro Avg	0.85	0.70	0.73	9518
Weighted Avg	0.83	0.82	0.80	9518

Table 6: QWEN Metrics (Post Fact Check + Entropy)

	Precision	Recall	F1-score	Support
False	0.78	0.75	0.76	3337
True	0.40	0.44	0.42	6663
Accuracy			0.66	10000
Macro Avg	0.59	0.59	0.59	10000
Weighted Avg	0.67	0.66	0.67	10000

Table 7: Mistral Metrics (Post NLI)

	Precision	Recall	F1-score	Support
False	0.86	0.96	0.91	3337
True	0.86	0.58	0.69	6663
Accuracy			0.86	10000
Macro Avg	0.86	0.77	0.80	10000
Weighted Avg	0.86	0.86	0.85	10000

Table 8: Mistral Metrics (Post Fact Check + Entropy)

	Precision	Recall	F1-score	Support
False	0.42	0.57	0.49	3337
True	0.65	0.50	0.57	6663
Accuracy			0.53	10000
Macro Avg	0.53	0.54	0.53	10000
Weighted Avg	0.56	0.53	0.53	10000

Table 9: Bloom Metrics (Post NLI)

	Precision	Recall	F1-score	Support
False	0.97	0.76	0.85	3337
True	0.87	0.98	0.92	6663
Accuracy			0.90	10000
Macro Avg	0.92	0.87	0.89	10000
Weighted Avg	0.91	0.90	0.90	10000

Table 10: Bloom Metrics (Post Fact Check + Entropy)

5.2 Detection Observations

We evaluated metrics (Accuracy, Precision, Recall, F1) for all models at each stage in the pipeline to see the positive effect fact checking and entropy scoring had. The scores at the end of the pipeline were as follows:

Metric	LLAMA	Gemma	Qwen	Mistral	Bloom
Accuracy	0.88	0.84	0.82	0.86	0.90
F1-score	0.91	0.80	0.57	0.69	0.92

Table 11: Comparison of Accuracy and F1-score Across Models

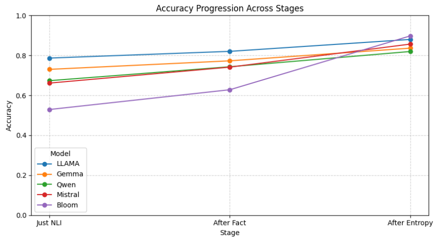

Figure 3: Hallucination Detection: Accuracy Progression

- The upward trend across all models reinforces that our detection pipeline is model-agnostic and scales well regardless of the initial performance provided by vanilla NLI.

- A surprising observation is that even a relatively lesser-known, non-mainstream model like Bloom significantly closed the performance gap, achieving competitive accuracy after the final stage—surpassing even LLaMA.
- LLaMA, despite its strong baseline, showed relatively smaller gains, suggesting that high-performing models may saturate earlier in improvement.
- The convergence in accuracy scores across all models by the final stage demonstrates that hallucination detection can act as a quality equalizer, reducing disparities between frontier and smaller models in downstream reliability.

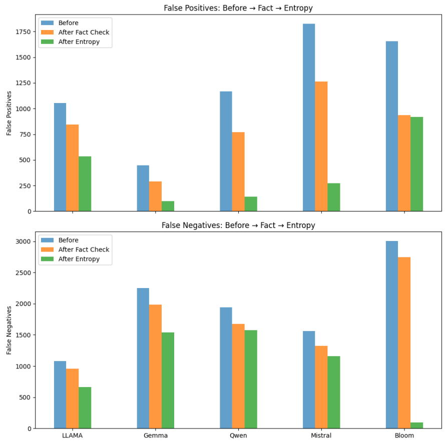

Figure 4: Hallucination Detection: FP and FN across Stages

- A key takeaway is that entropy helped detect borderline cases where the model appeared confident despite lacking factual grounding—cases where fact-checking combined with NLI alone was insufficient.
- Fact-checking had a more pronounced effect on reducing false negatives, whereas entropy refinement was more effective at minimizing false positives, highlighting the complementary strengths of these components in the pipeline.

5.3 Hallucination Reduction using RAG

We assessed our RAG-based hallucination reduction pipeline's performance using the Gemma, LLaMA, and Mistral language models. Following the application of retrieval-augmented generation, each model demonstrated a significant reduction in hallucination rates. LLaMA demonstrated an even more striking improvement, falling from about 66% to barely 4%, while the Gemma model decreased its hallucination rate from roughly 46% to less than 9%. Mistral, whose baseline was more moderate, also saw a considerable improvement, with hallucinations decreasing from 28% to about 12%. These steady gains across a range of models demonstrate our pipeline's resilience and model-agnostic nature. The findings imply that retrieval grounding can help even high-performing models, such as LLaMA, while underperforming models gain even more.

6 Conclusions and future work

6.1 Conclusion

In this study, we investigated a multimodal approach to hallucination identification in large language models (LLMs) by combining natural language inference (NLI), heuristic vagueness detection, entropy-based uncertainty estimation, and external fact-checking with Wikipedia. Our pipeline was able to identify and filter hallucinogenic material from a range of LLMs, including as Mistral, Gemma, and LLaMA, with greater assurance. Incorporating Retrieval-Augmented Generation (RAG) further improved factual accuracy, demonstrating that context-aware generation dramatically reduces the likelihood of hallucinations. Our findings demonstrated steady F1-score gains for all models, with Mistral, showing particularly significant gains.

6.2 Future Work

Despite our hybrid pipeline's promising nature, there are still a few areas that may need work. First, boosting NLI resilience may lower misclassifications in intricate or domain-specific scenarios. Instruction-tuned verifiers or NLI models customized for the training domain could bridge this gap. Second, expanding the range of fact-checking sources beyond Wikipedia, including academic databases, news datasets, or structured knowledge graphs, could improve coverage of subjects that are obscure or dynamic. Third, including more advanced ambiguity detection utilizing large-scale language understanding models may reduce the requirement for preset heuristics. Lastly, we want to test this pipeline in real-time applications and various LLM use cases, including legal text creation or healthcare, to determine its practical impact and domain transferability.

7 Individual Contributions

- **Shruti** and **Vedant** focused primarily on dataset creation. They researched suitable models, explored available APIs, and built the dataset using resources accessible to the team. The final dataset, which was human-evaluated, achieved an accuracy of 95%.
- **Salil** led the implementation of the Natural Language Inference (NLI) and fact-checking modules. He also played a key role in integrating entropy-based scoring to improve the hallucination detection accuracy.
- **Reuben** led the development of the RAG pipeline, with **Shruti** contributing significantly to embedding creation, vector store setup and API integration.
- **Xenus** supported the execution and debugging of the RAG pipeline, and was actively involved throughout the project in testing, evaluation, and final integration.

So far, we have all contributed equally to the project, and we have made sure to divide the tasks evenly in terms of effort, collaboration, and decision-making.

References

et al. Cheng. 2023. Truthfulqa: Benchmarking llms on hallucination. *arXiv preprint arXiv:2305.13669*.

Groq. 2025. Groq api keys management. Accessed: 2025-04-03.

et al. Gunasekar. 2023. Factually data enhancement for reducing hallucinations in llms. *arXiv preprint arXiv:2310.07289*.

et al. Li. 2023. Halueval: A large-scale hallucination evaluation benchmark. *arXiv preprint arXiv:2305.11747*.

et al. Maynez. 2020. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

Pasquale Minervini. 2023. Halueval: Hallucination evaluation benchmark. Accessed: 2025-04-03.

et al. Ram. 2023. Retrieval augmentation as a hallucination mitigation strategy. *arXiv preprint arXiv:2310.03214*.

Together AI. 2025. Together ai api documentation. Accessed: 2025-04-03.