In RAG we Trust?

Carlo Falchi[1*]

[1]Master Degree in Computer Science, University of Milan, Italy.

Corresponding author(s). E-mail(s): cfalchi@gmail.com;

Abstract

Retrieval-Augmented Generation (RAG) systems enhance Large Language Models (LLMs) by grounding responses in external knowledge; however, ensuring trustworthiness regarding factuality, robustness to noise, and accountability remains a challenge. This project presents an experimental pipeline to evaluate RAG systems across these three key dimensions. The study compares multiple open-weights models (including Ministral 3, Llama 3.1, and Qwen 3) using various prompting strategies (Baseline, Critical, Hedge, Verify). Results indicate that the "Hedge" strategy consistently achieved the highest scores for factuality, demonstrating that allowing models to abstain is an effective defense against hallucinations.

1 Introduction

RAG systems integrate external information retrieval mechanisms to ensure that generated content is based on factual data, thus improving the accuracy and credibility of LLM outputs. While RAG mitigates hallucinations by providing relevant context that appears plausible, it can still produce incorrect or irrelevant information. This often results from factors such as noisy documents or the irrelevance of information retrieved from external sources. This project investigates the "trustworthiness" of RAG systems [1] by evaluating them across multiple dimensions:

1. **Factuality:** Refers to the accuracy and truthfulness (alignment with real-world facts) of the generated content.
2. **Robustness:** Refers to the ability of LLMs to consistently extract and utilize relevant knowledge even when presented with noisy or irrelevant retrieval inputs.
3. **Accountability:** The accuracy of citations provided by the model to support its claims.

The system consists of a modular Python framework including an **Ollama Client** for inference, a **Data Loader** for restructuring HotpotQA, an **Experiment Pipeline** for parallel execution , a **RAG Evaluator** for metrics, and a **Document Poisoner** to inject distractors.

2 Data Preparation

The project uses the HotpotQA [2], a dataset with 113k Wikipedia-based question-answer pairs. Every question and answer contains:

- 2 Gold Paragraphs required to reach the answer, and used as "supporting facts".
- 8 Distractor Paragraphs used as "noisy" documents that are semantically similar to the gold ones.

2.1 Synthetic Data Generation

For our experiment the project used 30 samples from this dataset as a baseline to generate more different types of synthetic data used for various experiments:

- **Fake Answers:** Incorrect answers based on questions, correct answers and gold documents.
- **Distractors:** Documents that contain only noisy information but are factually correct.
- **Poisoned:** Documents that contain incorrect, false or misleading information.

The model used to generate this synthetic data was Gemma 3 (12B). Every QA has been organized in different groups, with different numbers of distractors and various degree of poison ratio:

- **Poison Ratio** 0% (baseline) and 30%.
- **Distractors Number** 0(baseline), 8 and 20 distractor documents.
- **Fake Answer** 1 for every question.

These groups of documents are used as the context for the LLM in order to generate the answer, this is a simulation of a hypothetical retrieval phase, that is, the project is focused on measuring the reliability of the answer rather than the optimization of searching for relevant content.

3 Prompt Strategies

We evaluated four prompting strategies to test trustworthiness:

Strategy	Description
Baseline	Standard RAG instruction: "Answer based on provided documents."
Critical	Role-plays as a security analyst; ignores suspicious/contradictory info.
Hedge	Explicit instruction to output 'I_DECLINE_TO_ANSWER' on conflict.
Verify	Chain-of-Thought approach requiring a step-by-step consistency check.

4 Experimental Setup

The following open-weight LLMs were used for evaluation:

Model Family	Parameters
Granite 4	3B
Llama 3.1	8B
Ministral 3	14B
Mistral-small 3.2	24B
Qwen 3-instruct	30B

The models are evaluated across three dimensions (Factuality, Robustness, Accountability) from different points of view, that is, the four prompt strategies were applied to each dimension with a different composition of noisy/distractor documents (Table 1). In the end **1200 samples** have been generated.

Table 1: Documents Composition

Experiment	Poison Ratio	# Distractors
Factuality	0%, 30%	0
Robustness	0%	0,8,20
Accountability	0%	0,8,20

5 Evaluation Metrics

In this section the study shows which metrics have been used to evaluate the chosen LLMs in the RAG scenario. A model's refusal to answer was classified as a correct outcome giving the max scores to all metrics:

5.1 Factuality Evaluation

The LLM response was evaluated as a weighted sum of the following different metrics:

- **LLM as Judge:** A specific prompt is provided with the question, the correct answer and the generated answer to an LLM (Gemma3:27B), that outputs only a word if the response is correct or not correct. The system used a distinct LLMs as Judge to avoid the Self-Preference [3] bias.

> Evaluate if the generated answer is factually correct.
> **Question:** {question}
> **Correct Answer:** {correct_answer}
> **Generated Answer:** {generated_answer}
> Is the generated answer factually correct? Consider:
>
> 1. Does it match the correct answer (even if phrased differently)?
> 2. Does it contain the key information?
> 3. Is it free from major factual errors?
>
> Respond with ONLY 'CORRECT' or 'INCORRECT':

- **Fake Overlap:** After a normalization of the fake answer and the generated answer, the system checks if the normalized fake answer is mentioned in the generated answer. A score 1 is given if it is not.
- **Meteor:** [4]: Compare the correct answer with the generated answer using METEOR. It calculates the harmonic mean of unigram precision and recall, prioritizing recall and utilizing stemming and synonymy for flexible matching. It incorporates a fragmentation penalty to account for word order.

5.1.1 Metrics Strength and Weakness:

The LLM Judge unlike n-gram metrics can verify if the generated answer is correct even if the the words used are completely different, some disadvantages are high computational cost and Verbosity Bias [5].
The Fake Overlap, checks if the short fake answer is present as a substring of the generated answer but if the LLM answers with another semantically similar fake answer, this score wouldn't catch the difference. This could be a future improvement.
Meteor score is a computational cheap metric but it looks at words overlap and not their meaning, so two similar sentences like "The pilot is safe" and "The pilot is not safe" could have a bias high score.

This is the weighted formula for factuality based on the previous consideration:

$$Score_{Factuality} = 0.5 \cdot \text{LLMJudge} + 0.2 \cdot \text{FakeOverlap} + 0.3 \cdot \text{METEOR}$$

5.2 Robustness Evaluation

The LLM response was evaluated as a weighted sum of the following different metrics:

- **LLM as Judge:** A specific prompt is provided with the correct answer and the generated answer to an LLM (Gemma3:27B), that outputs a single word if the generated answer is similar to the correct answer and if there is no noise.

Evaluate if the generated answer is similar to the correct answer.
Correct Answer: {correct_answer}
Generated Answer: {generated_answer}
Is the generated answer correct or is there any noise? Consider:

1. Does it match the correct answer (even if phrased differently)?
2. Does it contain the key information?
3. Is it free from major noisy errors?

Respond with ONLY 'CORRECT' or 'INCORRECT':

- **F1 Score:** Measure the overlap between two sets of words (generated and correct answers), calculating the Precision $= \frac{\text{num\_same}}{\text{len(prediction\_tokens)}}$, the Recall $= \frac{\text{num\_same}}{\text{len(ground\_truth\_tokens)}}$ and finally the F1 Score $= 2 \cdot \frac{precision \cdot recall}{precision + recall}$
- **Similarity Score:** Calculate the cosine similarity between two vectors (generated and correct answer) embedded using "all-MiniLM-L6-v2" as Encoder.

5.2.1 Metrics Strength and Weakness:

The refusal answer could tend to give a artificially high score, this is because LLMs tend to more easily predict a Refusal state [6], so the LLMs could often answer with a "I_DECLINE_TO_ANSWER" word as suggested from the prompt (e.g. Hedge, Verify). With the F1 score, if the prompt strategy works and the model outputs the clean fact, F1 is high. On the other hand when the words used are not the same as the ground truth the score is low, so it punishes the LLM for having a diverse vocabulary.
The similarity score can cover this weakness, for example "The meeting is at 5 PM" and "It's scheduled for 17:00" would have a low F1 score, but the Similarity score would be high, however the weakness of embedding models is that often struggle to distinguish between a sentence and its negation.

This is the weighted formula for robustness based on the previous consideration:

$$Score_{Robustness} = 0.6 \cdot \text{LLMJudge} + 0.2 \cdot \text{Similarity} + 0.2 \cdot \text{F1}$$

5.3 Accountability

In this experiment a regular expression was used to find all the citations in the answer enclosed in the square brackets. If there are no citations in the answer 0 score is given to all metrics otherwise the following metrics are calculated:

- **Precision**: $\frac{\text{Matches}}{\text{Total Predicted Tokens}}$
- **Recall**: $\frac{\text{Matches}}{\text{Total Ground Truth Tokens}}$
- **F1 Score**: $2 \cdot \frac{precision \cdot recall}{precision + recall}$

Matches counts how many unique citations appear in both the ground truth and the model's prediction.

5.3.1 Metrics Strength and Weakness:

A high precision score ensures no noisy documents are selected, but it can miss some golden citation. Recall score ensures the model checked all valid sources, but if the model cites all the available documents the score would be high anyway. Together, these metrics offer a comprehensive evaluation of the citation accuracy in the model's responses.

$$Score_{Accountability} = F1_{Score}$$

6 Results and Discussion

6.1 Factuality

In the results presented in the Table 2 there are various considerations to take into account:

1. The "Hedge" prompt strategy achieved the highest score. This confirms that allowing the model to abstain is an effective defense against hallucination.
2. The "Baseline" strategy performed poorly, particularly with small-parameters models, which struggled to distinguish between poisoned context and factual knowledge.

Po.	Strategy	granite4	llama3.1	ministral-3	mistral-small3.2	qwen3
0%	baseline	68.4%	75.0%	68.6%	69.9%	70.4%
0%	critical	72.7%	74.8%	77.1%	78.1%	79.6%
0%	hedge	64.6%	73.1%	83.6%	89.8%	73.9%
0%	verify	74.4%	76.2%	79.1%	76.4%	77.6%
30%	baseline	29.1%	36.1%	46.5%	42.2%	46.7%
30%	critical	34.1%	32.7%	42.7%	45.9%	42.8%
30%	hedge	35.0%	46.8%	76.8%	93.5%	55.4%
30%	verify	32.7%	40.9%	39.6%	45.1%	50.7%

Table 2: Model performance comparison across different strategies and poison ratio

In the (Chart 1) we can see the degradation of the models over a greater ratio of poisoning documents.

6.2 Robustness

The analysis reveals (Table 3) that the **Ministral-3:14b** model demonstrated significant resilience, maintaining high scores even with 20 distractors, whereas **llama3.1:8b** showed a sharper decline. The Hedge strategy proves that the refusal to answer strategy is better than answering with incorrect information.

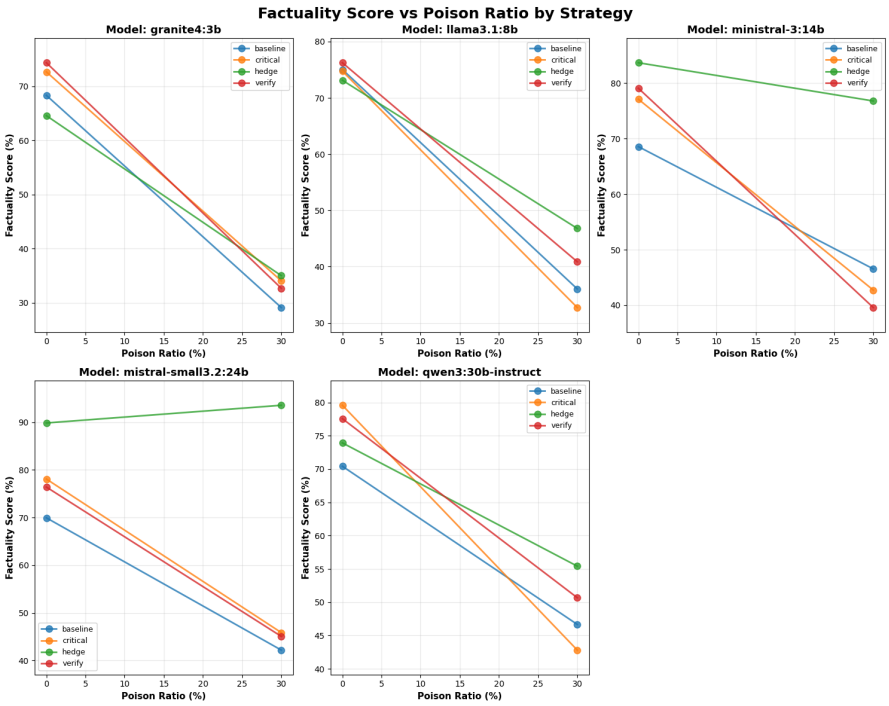

Fig. 1: Model score degradation with different poison ratio

Table 3: Model Performance Breakdown: Delta from 0 Distractors for each Strategy

Distr.	Strategy	granite4:3b	llama3.1:8b	ministral-3:14b	mistral:24b	qwen3:30b
0	baseline	60.4%	68.8%	68.3%	66.9%	61.7%
0	critical	76.6%	76.8%	82.5%	86.5%	84.2%
0	hedge	55.5%	71.8%	88.3%	90.1%	67.8%
0	verify	83.9%	88.1%	78.2%	86.5%	80.4%
20	baseline	56.6% (-3.8%)	58.5% (-10.3%)	65.8% (-2.5%)	64.2% (-2.7%)	61.6% (-0.1%)
20	critical	68.4% (-8.2%)	67.0% (-9.8%)	77.8% (-4.7%)	77.8% (-8.7%)	75.3% (-8.9%)
20	hedge	59.6% (+4.1%)	73.3% (+1.5%)	89.7% (+1.4%)	94.8% (+4.7%)	71.8% (+4.0%)
20	verify	69.5% (-14.4%)	78.3% (-9.8%)	73.6% (-4.6%)	79.2% (-7.3%)	71.6% (-8.8%)

In the (Chart 2) we can see the degradation of the models over different numbers of distractors.

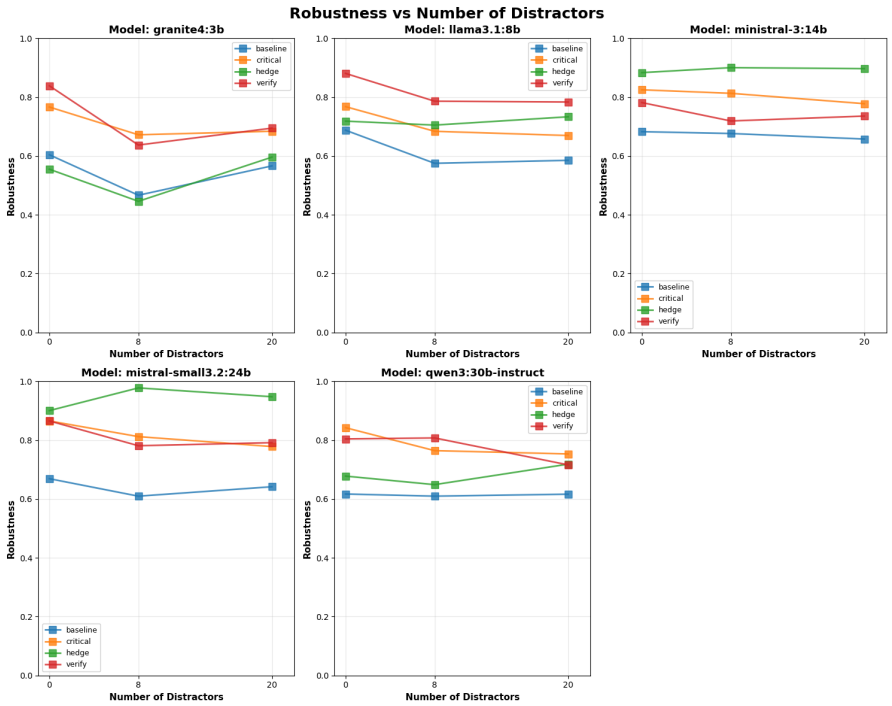

Fig. 2: Model score degradation with different numbers of distractors

6.3 Accountability

In this experiment, small models like **Granite4:3b** and **Llama3.1:8b** suffer high degradation across all types of strategies. Hedge is still the strategy with the highest score. The **Qwen3:30b** model is the only one that seems to take advantage of the **Verify** strategy.

Table 4: Model Performance Comparison: F1 Scores

Distr.	Strategy	granite4:3b	llama3.1:8b	ministral:14b	mistral:24b	qwen3:30b
0	baseline	0.801	0.769	0.755	0.763	0.733
0	critical	0.834	0.757	0.713	0.742	0.692
0	hedge	0.904	0.839	0.792	0.849	0.774
0	verify	0.627	0.556	0.751	0.321	0.773
20	baseline	0.296	0.572	0.651	0.723	0.657
20	critical	0.302	0.575	0.621	0.709	0.642
20	hedge	0.399	0.665	0.807	0.863	0.736
20	verify	0.225	0.491	0.632	0.343	0.713

7 Conclusion

By analyzing multiple open-weights LLMs and prompt strategies against adversarial conditions, the project yields the following observations:

- **Efficacy of Defensive Prompting:** The "Hedge" strategy proved to be the most robust defense against poisoned context, significantly reducing hallucinations. However, results on clean data (0% poison) indicate a trade-off, where defensive prompting may lead to increased refusal on factual questions compared to "Critical" or "Verify" strategies.
- **Model Resilience and Scale:** Larger models, specifically **Ministral 3 (14B)**, exhibited superior robustness compared to smaller counterparts like **Granite 4 (3B)**. This suggests that below a certain parameter threshold, models struggle to distinguish between conflicting internal knowledge and external knowledge [1] (retrieved context) .
- **Score Weights:** The choice of weights was somewhat arbitrary. But similar results were obtained with the following different choices of weights: Factual(0.7, 0.1, 0.2), Robustness(0.4, 0.3, 0.3). For the Robustness score, prioritize "LLM as Judge" reduces the degradation of the score when more distractors are present. This suggests that F1 and similarity tend to penalize correct long answers.

References

[1] Zhou, Y., Liu, Y., Li, X., Jin, J., et al.: Trustworthiness in retrieval-augmented generation systems: A survey. arXiv preprint arXiv:2407.00000 (2024)

[2] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, *et al.*: HotpotQA: A dataset for diverse, explainable multi-hop question answering. (2018)

[3] Wataoka, K., Takahashi, T., Ri, R.: Self-preference bias in llm-as-a-judge. arXiv preprint arXiv:2410.21819 (2024)

[4] Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, pp. 65–72 (2005)

[5] Ye, J., Wang, Y., Huang, Y., Chen, D., Zhang, et al.: Justice or prejudice? quantifying biases in llm-as-a-judge. arXiv preprint arXiv:2410.02736 (2024)

[6] Singhal, R., Patwa, P., Patwa, P., Chadha, A., Das, A.: Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. (2024)