Reducing Hallucinations in LLMs Using Prompt Engineering Strategies*

Hrishita Patra, Jahnavi Bobba, Megha Bhat, Sujatha R. Upadhyaya
Department of Computer Science and Engineering
PES University, Bangalore, India
{hrishitapatra, janubobba1, meghajbhat}@gmail.com, sujathar@pes.edu

Abstract—**Large Language Models (LLMs) are very powerful across NLP tasks and prone to hallucinations—a fabricated and misleading output which runs counter to the desired reliability. In this explorative study, we look into prompt engineering techniques that can possibly minimize such hallucinations by focusing on factuality, coherence of logic, and overall trustworthiness of outputs. Experiments were run using the Mistral-7B Instruct model on Kaggle Notebooks and Hugging Face. This research work evaluates three prompting strategies—Chain-of-Verification (CoVe), Chain-of-Thought (CoT), and a Hybrid CoVe+CoT—against a baseline without structured prompting, across two tasks: multi-hop question answering and knowledge-grounded dialogue. We demonstrate via the HaluEval benchmark that CoT improves reasoning ability while CoVe improves post-generation factual correctness. The hybrid method achieves the best trade-off with respect to accuracy, precision, recall, and F1.**

I. Introduction

Large Language Models (LLMs) like Mistral-7B are effective in QA, dialogue, and summarization tasks across domains such as healthcare, law, education and many more. However, they often produce **hallucinations**—fluent but factually incorrect or fabricated content—which pose serious risks in high-stakes applications.

Common hallucination types include:

- *Factual Inaccuracy* (e.g., wrong dates/names)
- *Inference Errors* (invalid reasoning)
- *Unsupported Claims* (unbacked by source)
- *Entity Fabrication* (nonexistent entities)

These are challenging to detect due to their surface plausibility. **Prompt engineering** offers a lightweight mitigation strategy—modifying inputs to guide model behavior without retraining.

Rashkin et al. [1] highlight the importance of attribution in evaluating factual correctness, identifying hallucinations as a major trust barrier. Building on this, our study experiments with structured prompting techniques like Chain-of-Thought (CoT) [2] and Chain-of-Verification (CoVe) [3], which aid in reasoning and post-response validation. Prior work by Vatsal and Dubey [4] supports these as effective approaches.

We investigate four prompting strategies to reduce hallucinations:

1) **Baseline:** Plain input-output, used for control.
2) **CoVe:** Adds self-verification after generation.
3) **CoT:** Stepwise reasoning before answering.
4) **Hybrid:** Combines CoT reasoning with CoVe checking.

Evaluation uses the **HaluEval** benchmark across:

- **HotpotQA (Multi-hop QA):** Requires reasoning across facts.
- **OpenDialKG (Dialogue):** Needs accurate, grounded responses.

We use Mistral-7B Instruct (4-bit) on Kaggle via a modular pipeline built with Hugging Face and Python.

Key contributions:

- A task-agnostic, modular pipeline for hallucination analysis across QA and dialogue.
- Comparative study of Baseline, CoT, CoVe, and Hybrid prompts.
- Fine-grained evaluation across hallucination categories (e.g., factual errors, inference gaps).
- Reproducible setup with Kaggle, Hugging Face, and JSONL format.
- Evidence that CoT+CoVe reduces hallucinations without hurting fluency.
- Practical insights on prompt engineering as an alternative to fine-tuning.

Our results show structured prompting can substantially reduce LLM hallucinations, improving factual reliability in NLP applications.

II. Related Work

Hallucination in large language models (LLMs) refers to the generation of fluent yet factually incorrect or unsupported content. This issue is particularly concerning in sensitive domains like healthcare, education, and law, where accuracy is critical. Hallucinations often stem from noisy retrievals, conflicting input contexts, or inadequate reasoning during generation. As a result, measuring hallucinations reliably remains challenging, requiring annotated datasets, strong baselines, and standardized metrics [1], [5], [6].

To address this, **HaluEval** [7] offers a benchmark with annotated examples across QA, summarization, and dialogue tasks. It also standardizes evaluation using precision, recall, and F1 scores, which we adopt. Similarly, datasets like **HotpotQA** [8] support multi-hop reasoning and explanation evaluation—both relevant to hallucination detection and mitigation.

Prompt engineering has gained prominence as a lightweight and generalizable strategy to guide LLMs without the need for fine-tuning. Among the most effective is **Chain-of-Thought (CoT)** prompting [2], which encourages the model to reason step-by-step before answering. CoT improves logical coherence and reduces unsupported or shallow responses.

A complementary method is **Chain-of-Verification (CoVe)** [3], which prompts the model to verify its own output post-generation. This acts as a built-in fact-checking step, improving factual grounding and filtering unsupported claims. Our work explores a hybrid of CoT and CoVe to combine their respective strengths.

Beyond CoT [9] and CoVe, other strategies have emerged, including **ReAct** [10], which blends reasoning with external actions like evidence retrieval, **Chain-of-Note (CoN)** [11], which improves handling of retrieved context, and **Chain-of-Knowledge (CoK)** [12], which integrates structured knowledge into the generation process.

These prompting techniques have been primarily applied to large-scale models such as **GPT-3** [13], **PaLM** [14], and **LLaMA** [15]. In contrast, our work applies these ideas to **Mistral-7B** [16], a smaller yet efficient dense transformer. As noted in the LLM survey by Zhao et al. [17], architecture, training methods, and scale significantly influence model behavior, motivating the study of hallucination mitigation in more compact models.

While most prior research focuses on hallucination detection post-generation, our study also explores prevention during generation. We propose a hybrid CoVe + CoT prompting strategy that embeds reasoning and verification into the generation process, aiming to improve factual reliability in LLM outputs.

III. Dataset and Resources

We evaluate hallucination generation and mitigation using three datasets across QA and dialogue tasks: HotpotQA, OpenDialKG, and HaluEval.

A. HotpotQA (QA Task)

HotpotQA [8] contains 100K+ multi-hop QA examples which need reasoning over several documents. Each example consists of a question, reference Wikipedia paragraphs and the ground-truth answer. In this work, we apply it to generate hallucinated QA samples as well as to test multi-step reasoning.

B. OpenDialKG (Dialogue Task)

In OpenDialKG, we model knowledge-grounded dialogues where assistant responses are linked to structured paths in a knowledge graph. Dialogue history, grounded response, and supporting path are provided in each example — a perfect setting for evaluating factuality in conversations.

C. HaluEval Benchmark

HaluEval [7][1] is an extensive hallucination benchmark. We exploit its QA and dialogue of labeled hallucinated and factual outputs. It has standardized evalutation in terms of P, R, A and F1.

Together, these datasets provide a diverse, replicable setup for testing hallucination detection and mitigation in LLMs.

TABLE I
HaluEval Dataset Files Used

File Name	Samples
`qa_data.json`	10,000
`dialogue_data.json`	10,000
`summarization_data.json`	10,000
`general_data.json`	5,000

IV. Methodology

A. General Methodology

All four prompting strategies follow a unified and modular pipeline that consists of input processing, prompt-based generation, pairwise filtering, and factuality evaluation. This consistent structure ensures a fair comparison across all experimental setups and facilitates reproducibility.

a) Model and Environment.: We employ the Mistral-7B Instruct model in a 4-bit quantized format using HuggingFace Transformers with `BitsAndBytesConfig`. Using this helps in enabling efficient inference without reducing performance. All experiments are executed on Kaggle Notebooks with a Tesla P100 GPU, ensuring a stable computational environment.

b) Input Processing.: Inputs are formatted according to the specific task. For question answering (HotpotQA and HaluEval QA), each input consists of a `knowledge` context, a `question`, and the `right_answer`. For dialogue tasks (OpenDialKG and HaluEval Dialogue), inputs comprise the `knowledge`, previous `dialogue history`, and the correct `right_response`. Each dataset is loaded and preprocessed using a standardized script to maintain input integrity.

c) Prompt-Based Generation.: Task-specific prompt templates are employed to guide the model in generating hallucinated responses. Depending on the strategy (Baseline, CoT, CoVe, or CoVe+CoT), different reasoning scaffolds are injected into the prompt.

L. Ouyang et al. [18] introduced a game-changing approach for making language models better at following instructions by using human feedback during training. Their work on InstructGPT showed that models can produce more helpful and reliable outputs when they're trained to align with what people actually want. While we don't retrain our model, we take a similar goal-driven approach by using structured prompts—like CoT and CoVe—to guide the model toward generating more accurate and fact-based responses. Inspired by the idea of human-aligned generation, our method focuses on shaping model behavior through smart prompt design rather than fine-tuning, especially in tasks like multi-hop QA and dialogue.

Regardless of variation, generation is handled via a unified script `generate.py`, and all outputs are saved in structured JSON format for traceability and downstream evaluation.

d) Pairwise Filtering.: For each input example, two hallucinated responses are sampled. These candidates are then compared using a follow-up prompt that asks the model to select the more factually accurate one. This pairwise filtering

[1] https://github.com/RUCAIBox/HaluEval

is performed via `filtering.py`, and the chosen outputs are again stored in a structured JSON format for clarity and reuse.

e) Factuality Evaluation.: The final step involves presenting the model with a single response—either factual or hallucinated—and prompting it to classify the response as `Factual` or `Not Factual`. This evaluation, managed by `evaluate.py`, also captures the model's rationale behind its judgment, enriching the interpretability of our results. These annotations are stored for further qualitative analysis.

f) Reproducibility.: To ensure replicability, all prompting strategies share the same data pipeline, model configuration, and file structure. The only variable is the prompt design, allowing us to isolate and attribute any observed performance differences solely to the prompting strategy.

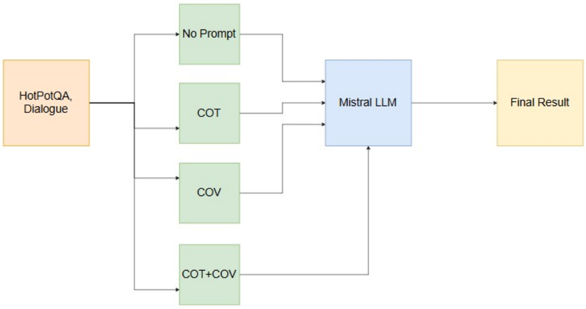


Fig. 1. Block diagram illustrating the end-to-end pipeline for generation, filtering, and evaluation.

In the following subsections, we detail each of the four prompting schemes used in our experiments.

B. Baseline Strategy – Plain Prompt

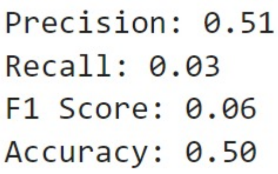
This strategy serves as the control setup to assess the hallucination behavior of the Mistral-7B model without applying structured prompting methods like CoT or CoVe.

a) Prompt Design.: For QA, the model is provided with Wikipedia-based context and a multi-hop question. In the dialogue task, it receives the conversation history and relevant knowledge graph information. No reasoning or verification instructions are included.

b) Generation.: The model generates two responses per input using the plain prompt. These outputs reflect its default behavior, without guidance to reason step-by-step or assess factuality.

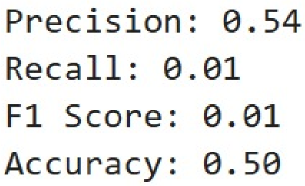
c) Filtering.: The model is then asked to compare the two responses and select the one that appears more accurate. This step retains the more plausible hallucinated response.

d) Evaluation.: Finally, the chosen response is evaluated for factual correctness. The model classifies it as either factual or hallucinated, helping to establish a baseline for comparison with more structured prompting strategies.



```
Precision: 0.51
Recall: 0.03
F1 Score: 0.06
Accuracy: 0.50
```

Fig. 2. QA Task: Baseline evaluation metrics



```
Precision: 0.54
Recall: 0.01
F1 Score: 0.01
Accuracy: 0.50
```

Fig. 3. Dialogue Task: Baseline evaluation metrics

e) Observations: Without structured prompting, the model struggles to detect hallucinations. QA recall (0.03) and F1 (0.06) are low, and Dialogue results show similarly poor recall (0.01) and F1 (0.01), despite moderate precision. These results highlight the limitations of the baseline and the need for structured prompts.

TABLE II
EVALUATION METRICS FOR BASELINE STRATEGY SETUP

Task	Precision	Recall	F1 Score	Accuracy
QA	0.51	0.03	0.06	0.50
Dialogue	0.54	0.01	0.01	0.50

C. Chain-of-Verification (CoVe): Self-Evaluation of Hallucinations

The Chain-of-Verification (CoVe) strategy enhances factual reliability by prompting the model to verify its own response after generation. This encourages the model to reflect on its output and assess whether it aligns with the input context.

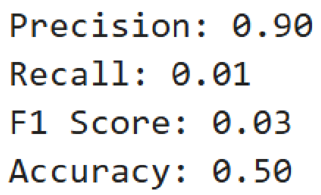
a) Prompt Design.: For QA, the model is asked to generate an answer and then judge its factual consistency with the provided context. In dialogue, it verifies its reply against the input knowledge and conversation history. These prompts guide the model toward self-evaluation.

b) Generation and Verification.: The model first generates a response, then reprocesses it through a verification prompt to assess alignment with the input. This two-step process helps reinforce factual correctness.

c) Filtering.: Two candidate responses are produced per input. The model is prompted to compare them and choose the one that is more accurate and contextually grounded.

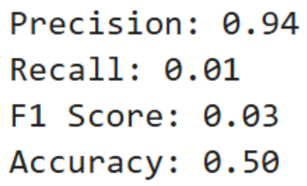
d) Evaluation.: The selected response is classified as either factual or hallucinated. This final step measures how well CoVe improves reliability compared to the unstructured baseline.

e) Observations: CoVe greatly improves precision in QA (0.90) and Dialogue (0.94), meaning correct detections are



```
Precision: 0.90
Recall: 0.01
F1 Score: 0.03
Accuracy: 0.50
```

Fig. 4. QA Task: CoVe Evaluation



```
Precision: 0.94
Recall: 0.01
F1 Score: 0.03
Accuracy: 0.50
```

Fig. 5. Dialogue Task: CoVe Evaluation

highly accurate. However, recall remains very low (0.01), leading to low F1 scores (0.03). This suggests that while CoVe boosts accuracy, it misses most hallucinations.

TABLE III
EVALUATION METRICS FOR CHAIN-OF-VERIFICATION (COVE) SETUP

Task	Precision	Recall	F1 Score	Accuracy
QA	0.90	0.01	0.03	0.50
Dialogue	0.94	0.01	0.03	0.50

D. Chain-of-Thought (CoT)

Chain-of-Thought (CoT) prompting encourages the model to reason step-by-step before answering, using cues like *"Let's think step by step"*. This helps generate more logically consistent and interpretable outputs.

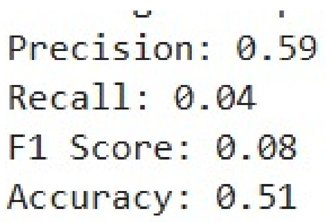
a) Prompt Design.: In QA, prompts are modified to include a reasoning cue before the answer. For dialogue, the model is similarly guided to reason through its reply rather than respond directly, encouraging more structured thinking.

b) Generation.: The model generates two responses per input using the reasoning-enhanced prompt. These outputs reflect a more deliberate reasoning process compared to the baseline.

c) Filtering.: The model compares the two generated responses and selects the one that is more accurate or coherent. This helps retain answers with better logical flow.

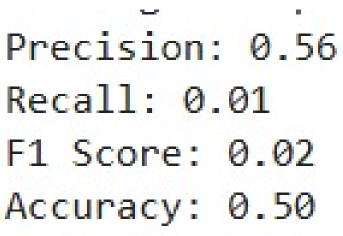
d) Evaluation.: The selected response is evaluated for factual accuracy. The model determines whether it is factual or hallucinated, allowing us to assess how CoT impacts factual consistency.

e) Observations: Chain-of-Thought shows modest gains over the baseline, especially in QA, with improved F1 (0.08) and recall (0.04). Precision remains moderate (0.59), and performance in dialogue is weak, with very low recall (0.01). While CoT aids reasoning, its impact on factual accuracy is limited, particularly in dialogue.



```
Precision: 0.59
Recall: 0.04
F1 Score: 0.08
Accuracy: 0.51
```

Fig. 6. QA: CoT Evaluation



```
Precision: 0.56
Recall: 0.01
F1 Score: 0.02
Accuracy: 0.50
```

Fig. 7. Dialogue: CoT Evaluation

E. Hybrid: CoVe + CoT

This hybrid strategy combines Chain-of-Thought (CoT) and Chain-of-Verification (CoVe) to improve both reasoning and factual accuracy in model-generated outputs.

a) Prompt Design.: The prompt first instructs the model to reason step-by-step using a Chain-of-Thought format (e.g., "Let's think step by step"). After generating a response, the model is given an additional prompt asking it to verify the factual accuracy of its own answer based on the original input. This two-part prompt design encourages both logical reasoning and post-response verification.

b) Generation.: Using the reasoning-enhanced prompt, the model generates two candidate responses for each input. These responses benefit from the CoT scaffold, which guides the model through intermediate reasoning steps before arriving at a final answer.

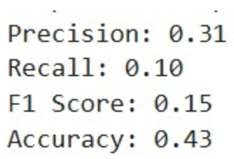
c) Filtering.: The two responses are then passed through a verification stage where the model is asked to compare them and select the one that is more factually consistent with the input. This process helps retain responses that are both logically coherent and grounded in factual context.

d) Evaluation.: The selected response is finally assessed for factuality. The model is asked to determine whether the response is factual or hallucinated. This final evaluation reflects the combined effect of reasoning and self-verification on improving overall output reliability.

e) Observations: The Hybrid strategy achieves the best overall results. In QA, it shows the highest F1 (0.15) and recall

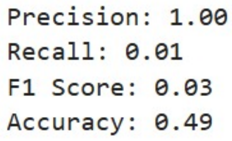
TABLE IV
EVALUATION METRICS FOR COT

Task	Precision	Recall	F1	Accuracy
QA	0.59	0.04	0.08	0.51
Dialogue	0.56	0.01	0.02	0.50



```
Precision: 0.31
Recall: 0.10
F1 Score: 0.15
Accuracy: 0.43
```

Fig. 8. QA: Hybrid Evaluation



```
Precision: 1.00
Recall: 0.01
F1 Score: 0.03
Accuracy: 0.49
```

Fig. 9. Dialogue: Hybrid Evaluation

(0.10), with balanced precision. In dialogue, it reaches perfect precision (1.00) but low recall (0.01), indicating that many hallucinations are still missed. Overall, combining reasoning and verification offers a more reliable approach to reducing hallucinations.

TABLE V
HYBRID CoVE + CoT EVALUATION METRICS

Task	Precision	Recall	F1	Accuracy
QA	0.31	0.10	0.15	0.43
Dialogue	1.00	0.01	0.03	0.49

V. Evaluation and Metrics

We employ a blind classification setup to evaluate each prompting strategy. Here, the model is shown a response—either hallucinated or factual—along with its input context, and must judge its correctness without knowing the label. This mimics real-world inference where the model verifies outputs autonomously.

Z. Ji et al. /citeacmsurvey2023 provide an extensive survey on hallucination in natural language generation, categorizing its types, causes, and evaluation methods—laying a foundational understanding that motivates our work.

Evaluation protocol:

- **Task:** Given a context (question or dialogue) and a response, the model classifies it as hallucinated (“Yes”) or factual (“No”).
- **Sampling:** Responses are randomly drawn from hallucinated and ground-truth sets for balanced, unbiased evaluation.
- **Judgment:** Model predictions are compared to true labels; in CoVe and Hybrid, verification feedback further informs accuracy.
- **Metrics:** Precision, Recall, Accuracy, F1-Score

This framework enables fair and structured comparison across all four prompting strategies—Baseline, CoVe, CoT, and Hybrid—on both QA and dialogue tasks.

TABLE VI
EVALUATION METRICS FOR QA PROMPTING STRATEGIES

Strategy	Precision	Recall	F1 Score	Accuracy
Initial Code	0.51	0.03	0.06	0.50
CoVe	0.90	0.01	0.03	0.50
CoT	0.59	0.04	0.08	0.51
CoVe + CoT	0.31	0.10	0.15	0.43

TABLE VII
EVALUATION METRICS FOR DIALOGUE PROMPTING STRATEGIES

Strategy	Precision	Recall	F1 Score	Accuracy
Initial Code	0.54	0.01	0.01	0.50
CoVe	0.94	0.01	0.03	0.50
CoT	0.56	0.01	0.02	0.50
CoVe + CoT	1.00	0.01	0.03	0.

VI. Results

Summary: The Hybrid (CoVe + CoT) strategy achieved the best overall performance, especially in QA, with the highest F1 score (0.15) and recall (0.10).

CoVe showed the highest precision in both QA and Dialogue, but suffered from low recall. CoT moderately improved reasoning and F1 over the baseline. The Baseline (Initial Code) performed the weakest across all metrics.

These results clearly demonstrate that prompt engineering—particularly structured approaches like CoT, CoVe, and their combination—is an effective method for reducing hallucinations in large language models.

A. QA Metric Comparison Visualization

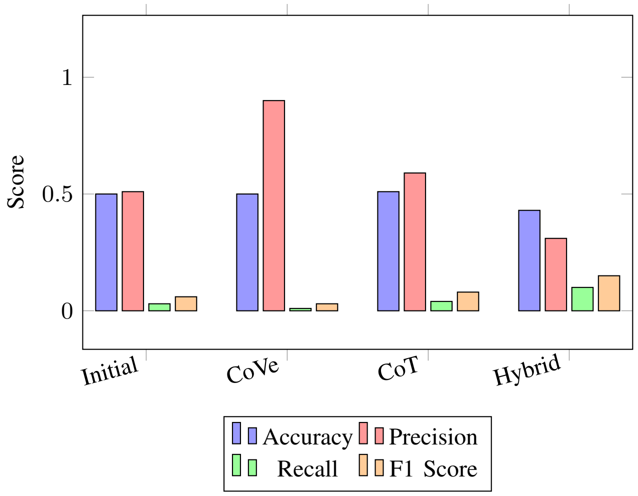


Fig. 10. QA Comparison of evaluation metrics across strategies.

B. Dialogue Metric Comparison Visualization

VII. Formulas

We use four standard classification metrics—Accuracy, Precision, Recall, and F1 Score—to evaluate hallucination detection across prompting strategies, offering insight into both correctness and robustness.

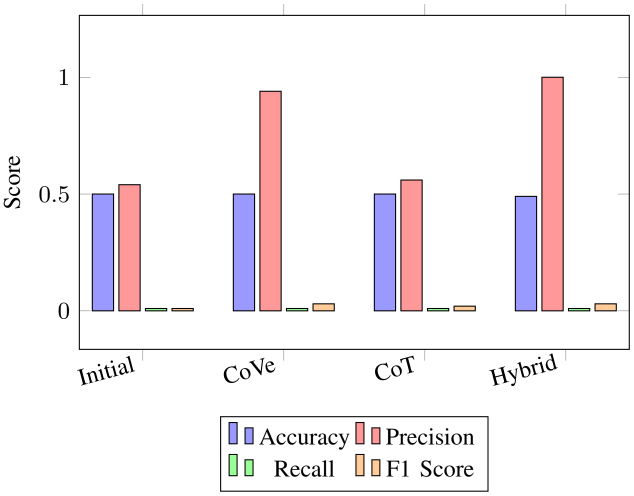


Fig. 11. Dialogue Comparison of evaluation metrics across strategies.

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score:**

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- **TP**: Hallucinations correctly identified.
- **TN**: Factual responses correctly identified.
- **FP**: Factual responses incorrectly flagged as hallucinated.
- **FN**: Hallucinations missed by the model.

Metrics are computed separately for QA and Dialogue tasks across all prompting strategies to highlight trade-offs between sensitivity (recall) and specificity (precision) in hallucination detection.

VIII. Conclusion

It is shown that for artificial-generated hallucinations structured prompting reduces them in large language models such as Mistral-7B Instruct. Of the strategies tested, the Hybrid approach (CoVe + CoT) produced the best results in QA, having the highest F1 score (0.15) and recall (0.10), indicating that combining reasoning with verification is beneficial.

CoVe achieved the highest precision by itself in both QA and Dialogue, which suggests CoVe possessed the highest accuracy in detecting hallucinations (but it also exhibited low recall). CoT yielded mild improvements over the baseline, mainly by architecture enhancement. There, the Baseline approach achieved the lowest performance, advocating the necessity of guided prompting.

In summary, we verify that prompt engineering, especially the combination of CoT and CoVe, 16 is a practical and effective method to enhance the factual reliability of LLMs (without retraining).

IX. Future Work and Limitations

A. Future Work

We plan to extend our framework to summarization, a task especially prone to hallucinations due to its abstract nature. Applying CoT and CoVe here will help assess their generalizability.

We also aim to explore other prompting strategies—such as Self-Ask, Zero-shot CoT, and ReAct—to broaden hallucination mitigation techniques. Evaluating on a range of LLMs (e.g., LLaMA-2, Falcon, GPT-3.5) may uncover model-specific behavior.

While our current evaluation relies on standard metrics like accuracy, precision, recall, and F1 score, M. Cao et al. [19] show that some hallucinations—though not grounded in the source—can still be factually correct, challenging the idea that all hallucinations are inherently harmful. Taking inspiration from this, future work could explore evaluation methods that go beyond surface metrics to assess the factual validity of hallucinated content more directly.

Recent studies have also shown that contrastive learning techniques can help reduce hallucination in conversational agents by improving attribution and factual grounding.

Additionally, integrating external knowledge sources, such as knowledge graphs or retrieval-augmented generation, may further improve factual grounding during both generation and verification.

B. Limitations

Our study focused only on Mistral-7B Instruct, so this could limit the applicability to other models. More generalization across architectures is required.

Note that we only focused on QA and dialogue tasks, excluding other tasks such as summarization and storytelling, which potentially pose different hallucination patterns.

The accuracy of CoT and CoVe is dependent on the quality of the prompts and reasoning ability of the model, and the model can still make mistakes in making confident predictions.

Lastly, the binary scoring method (factual vs hallucinated) might discard partial correct answers. Deeper insights might be possible with more nuanced or probabilistic scoring.

References

[1] H. Rashkin, V. Nikolaev, M. Lamm, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, and D. Reitter, “Measuring attribution in natural language generation models,” *CoRR, abs/2112.12870*, 2021.

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” *arXiv preprint arXiv:2201.11903*, 2022.

[3] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, “Chain-of-verification reduces hallucination in large language models,” *arXiv preprint arXiv:2309.11495*, 2023.

[4] S. Vatsal and H. Dubey, “A survey of prompt engineering methods in large language models for different nlp tasks,” *arXiv preprint arXiv:2407.12994*, 2024.

[5] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, and W. Peng, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv preprint arXiv:2311.05232*, 2023.

[6] T. Hu and X.-H. Zhou, "Unveiling llm evaluation focused on metrics: Challenges and solutions," *arXiv preprint arXiv:2404.09135*, 2024.

[7] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "Halueval: A large-scale hallucination evaluation benchmark for large language models," *arXiv preprint arXiv:2305.11747*, 2023.

[8] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[9] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei, "Challenging big-bench tasks and whether chain-of-thought can solve them," *arXiv preprint arXiv:2210.09261*, 2022.

[10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2023.

[11] W. Yu, H. Zhang, X. Pan, K. Ma, H. Wang, and D. Yu, "Chain-of-note: Enhancing robustness in retrieval-augmented language models," *arXiv preprint arXiv:2311.09210*, 2023.

[12] J. Wang, Q. Sun, X. Li, and M. Gao, "Boosting language models reasoning with chain-of-knowledge prompting," *arXiv preprint arXiv:2306.06427*, 2024.

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[14] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, and et al., "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[15] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, and et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[16] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[17] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, and et al., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, and et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[19] M. Cao, Y. Dong, and J. C. K. Cheung, "Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization," *arXiv preprint arXiv:2109.09784*, 2021.