IJHSR
ijhighschoolresearch.org
REVIEW ARTICLE

Prompt Engineering and Context Engineering: Reducing AI Hallucinations in Large Language Models

Abhinav Tammana

Los Altos High School, 201 Almond Avenue, Los Altos, California, 94022, USA; abhinav27.tammana@gmail.com

ABSTRACT: Large Language Models (LLMs) are widely used in today's world for a variety of tasks. They are known for generating confident and convincing content; however, these outputs are not always accurate. Such inaccuracies, commonly referred to as *hallucinations*, present a growing concern as individuals and organizations are increasingly relying on LLMs for content creation and augmentation. To address this concern, researchers and AI scholars have been exploring methods to design prompts and structure the context to improve the accuracy of the LLM responses. This research paper examines how prompt engineering and context engineering are being used to reduce hallucinations in LLMs, which techniques are effective, and whether all hallucinations should be considered detrimental. Most studies examined different approaches to prompt formulation, context structuring, and the use of tools such as Retrieval-Augmented Generation (RAG) to enhance the output. This paper posits that no single solution is universally applicable; there are specific techniques that work best for particular fields. However, context structure will be a key factor in reducing hallucinations and ensuring the accuracy of the output. Context engineering is expected to be as important as prompt engineering, and leveraging the two approaches together helps reduce hallucinations. This research is very important because LLMs are deeply integrated into our daily lives. Humans are becoming more reliant on and attached to AI. It is incredibly important to get accurate results from them, particularly as LLMs are increasingly utilized in critical fields such as medicine and finance.

KEYWORDS: Artificial Intelligence, Large Language Models (LLMs), Prompt Engineering, Context Engineering, AI Hallucinations.

Introduction

Large Language Models (LLMs) are advanced neural networks, trained on vast text datasets, and are capable of generating human-like output. In recent years, humans have become more reliant on LLMs such as GPT, Google's Gemini, Anthropic's Claude, X's Grok, and Meta's LLaMa models. These models perform a large variety of natural language processing (NLP) tasks - from summarization and question answering to writing code and making decisions. However, there is one serious issue with these LLMs: *hallucinations*. LLM hallucinations are generated responses that are not supported by factual truth, and are something the LLM, in other words, guesses. These inaccurate outputs can mislead users, especially in high-stakes domains.[1] Hallucinations in LLMs raise important questions regarding trust, safety, and responsibility. For example, an incorrect analysis of a medical report could endanger a patient, or an incorrect citation in a student's paper could undermine academic integrity. There are many studies that document LLM hallucinations in casual chats, and professional tools like customer service bots, writing assistants, and platforms used for drafting legal documents.[2] As LLMs continue to evolve and individuals and organizations become more reliant on them, the need to reduce LLM hallucinations becomes more urgent. To address this underlying challenge, researchers have turned to two approaches: 1) Prompt Engineering, which modifies the phrasing, structure, or reasoning steps of the input to improve its reliability using linguistic tuning, Chain-of-Thought reasoning, and rewriting tools such as DecoPrompt. 2) Context Engineering, which supplies the model with additional background information it needs using RAG and graph-based inputs. Prompt Engineering revolves around the structuring, phrasing, and crafting of the user input.[3] Context engineering focuses on supplying the model with relevant, structured, and task-specific background information, often through the context provided in the input.[4] These techniques aim to reduce hallucinations by providing context and clarifying the user's intent, so the model's prediction is more accurate.

This paper explores the effectiveness of prompt engineering and context engineering in reducing LLM hallucinations across various tasks. It is structured into 4 sections: Section 1: Prompt Engineering and Context Engineering: Distinctions and Definitions; Section 2: Prompt Engineering Strategies for Hallucination Mitigation; Section 3: Context Engineering Strategies for Hallucination Mitigation; Section 4: Hallucinations: Causes and Mitigations.

This paper uses a review-based approach, comparing findings from recent LLM hallucination studies. It analyzes secondary data drawn from benchmark experiments, prompt design evaluations, and task-specific performance reports. It reviews prompting strategies such as Chain-of-Thought, DecoPrompt, Linguistic tuning, and context engineering strategies like Retrieval-Augment Generation (RAG) and graph-based input structuring. It also provides an insight into common hallucination causes. Ultimately, this paper argues that neither prompting nor context engineering alone is sufficient in reducing hallucinations. Rather, there should be a hybrid approach

DOI: 10.36838/IJHSR811.16

that is tailored to every field and task. This is the most effective way of reducing hallucinations. Moreover, the paper suggests that not all hallucinations are bad. It suggests that some hallucinations may be guided or filtered[5] paving the path for new directions for future research into productive and controlled hallucinations.

Discussion

1. Prompt Engineering and Context Engineering: Distinctions and Definitions:

1.1. Distinguishing Prompt Engineering and Context Engineering:

Prompt engineering and context engineering are related, but separate fields. Clarifying the boundaries of each field aids in evaluating their specific contributions toward reducing AI hallucinations. When users interact with an LLM, two key factors determine the quality of the output: how the input is phrased and what information is in the input for the LLM to use. Prompt engineering focuses on phrasing, structuring, and formatting inputs. This includes length, tone, and word choice. Meanwhile, context engineering concerns the information provided to the model for analysis and integration into its output. This information ranges from providing word definitions all the way to providing information graphs. These two fields are frequently employed in conjunction, leading to common misinterpretations of their distinct roles. However, recent findings suggest that they have their own benefits when it comes to reducing hallucinations. By clearly delineating them, the effectiveness of various approaches can be evaluated by examining different use cases.

1.2. Prompt Engineering:

Prompt engineering is essentially the process of optimizing interactions with an LLM to elicit the most accurate and useful output. You guide the model using carefully written prompts.[6] Researchers have found that the wording, length, and structure of a prompt can make a difference in the quality of the response.[1,7] Prompt engineering is not a recent topic; however, it has been evolving since 2015, when attention mechanisms were introduced to bring more importance to the context of a prompt.[8] There are many prompt engineering approaches, some being instruction-based prompts, metaphorical prompts, and even information-based prompts.[7] However, the goal remains the same throughout these different strategies: improve the model's output on a plethora of tasks like translation, summarization, question answering, and overall text generation.

1.3. Context Engineering:

Context engineering involves providing an LLM with appropriate context or background information to enable improved and more accurate predictions. The quality of the output depends a lot on the context it is given in.[4] The optimal method for obtaining context involves this process: 1) Retrieve information from a database. 2) Generate supporting details. 3) Process the information. 4) Feed it to the LLM via user input.[4] This approach is known as Retrieval-Augmented Generation, otherwise known as RAG. The approach previously outlined resembles human information processing, and RAG indeed follows the same approach. RAG interacts with external data available online and works with other models through a multi-agent system. Essentially, the purpose of context engineering is to create a more context-aware LLM through the inputs, to ensure the LLM possesses relevant, accurate information during response generation.

1.4. Distinctions and Effectiveness towards Hallucination Reduction:

Although prompt engineering and context engineering are related through their shared involvement in prompt modification, they constitute distinct fields with differing objectives. However, researchers have found that context engineering often outperforms prompt engineering due to its provision of a structured framework for the model to operate within.[9] This may sound confusing because prompt engineering is all about providing a structured input to the LLM. Prompt engineering focuses on refining instructions, so the model is less likely to hallucinate.[1,10] For example, adjusting prompts for ChatGPT, Gemini, and Claude makes the answers from the LLMs more reliable.[11] Some techniques, like Chain-of-Thought reasoning patterns and grammar-aware prompts, improve consistency and reduce the need for the LLM to hallucinate.[12,13] Context engineering focuses on providing the models with relevant outside information to use, really effective when using methods like RAG.[14,15] The purpose of this is to fill in the model's knowledge gap, so it would not hallucinate. Research shows that adding structured tags to this context reduces hallucinations by a staggering 99.88%.[16] The subsequent section will delve into specific prompt engineering strategies.

2. Prompt Engineering Strategies for Hallucination Mitigation:

2.1. Overview of Prompt Engineering for Hallucination Reduction:

Prompt engineering can reduce hallucinations by clarifying the input's intent, controlling the formality, and restructuring misleading or broad tasks. Many studies show that minor alterations in how a prompt is structured can lead to significant differences in the LLM output.[17] These subtle changes range all the way from small word changes to complete prompt-rewriting tools like DecoPrompt.[18] Hallucinations have remained a major barrier in ensuring accuracy in AI. As a result, prompt engineering has become its own field in trying to put a stop to this spreading concern. Prompt engineering solely relies on designing the instructions given to the LLM, making sure not to alter any of the LLM's core features.[19]

2.2. Linguistic Tuning:

Word choice and sentence structure influence hallucination rates in LLMs. Research shows that prompts characterized by formal word choice and concreteness reduce hallucinations.[18] This aligns with effective prompt design, which relies on principles such as clarity, contextual framing, and instructional phrasing; all of which contribute towards guiding LLMs in the right direction, away from hallucinations.[20] These im-

portant input elements help LLMs output more accurate and factual responses. Vague or ambiguous prompts increase the likelihood of hallucinations occurring because they prompt the LLM to extract information from the biases in its training data.[1] This makes the crafting of prompts important because it requires not only attention to vocabulary, but also the sentence structure, readability, and phrasing. All of these factors contribute to the accuracy of the LLM output.[18] Prompts with lower readability, concreteness, and formality cause challenges for LLMs, leading them to hallucinate.[18]

2.3. DecoPrompt:

One way to mitigate hallucinations is to completely rewrite the prompt.[21] Reformulated prompts provide more context and modify a model's answer to a more accurate and less confusing one.[17] One of these rewriting tools is called DecoPrompt. DecoPrompt is designed to mitigate hallucinations.[3] DecoPrompt, as suggested in the title, gives LLMs the ability to decode false prompts and encode them to be specific, concrete, and straightforward prompts.[3] This approach is inspired by the observation of the randomness of false-premise prompts being related to the likelihood of producing hallucinations.[3] Furthermore, the two experiments conducted on behalf of DecoPrompt have exemplified that DecoPrompt can effectively reduce hallucinations. DecoPrompt also allows for cross-model transferability, making it more usable for applications involving LLMs on a large scale.[3]

2.4. Chain-of-Thought:

Chain-of-Thought (CoT) prompting improves the output reliability by returning its output in step-by-step instructions. The reason for the step-by-step instructions is to guide the model through a logical sequence, allowing the model to correct its own mistakes.[22] While having well-structured inputs, with formal word choices, does reduce hallucinations, guiding the LLM through reasoning steps with CoT can further enhance mitigation against hallucinations.[22] But even well-designed prompts will fail if they are not matched with relevant context by the model. Therefore, prompt engineering must be partnered with a resource useful for retaining contextual information from the prompt given to maximize the accuracy of the LLM's output.[19]

2.5. Section Summary:

Prompt engineering offers many branches for reducing hallucinations in LLMs. Techniques like linguistic tuning, DecoPrompt, and Chain-of-Thought reasoning each address different causes of hallucinations: ambiguity to faulty processing. However, prompt engineering alone would not mitigate hallucinations. Hallucination mitigation will be most effective when prompt engineering is combined with context retrieval mechanisms (ie, Context Engineering) to generate the most accurate and hallucination-free response.

3. Context Engineering Strategies for Hallucination Mitigation:

3.1. Overview of Context Engineering for Hallucination Reduction:

Context engineering reduces hallucinations by providing relevant and organized information to the LLM, thereby supplying valid factual bases. Many studies conclude that LLMs struggle with accuracy when relying solely on their original training data, which, as previously discussed, can become outdated. Context engineering solves this problem by using methods like Retrieval Augmented Generation (RAG) and graph-based inputs. By grounding the LLM's output with reliable information, context engineering makes it more accurate.[23]

3.2. Retrieval Augmented Generation (RAG):

RAG is the most common context engineering method. It works by retrieving information from external documents and letting LLMs use that information while they are generating responses. Instead of LLMs relying solely on their training data, RAG enables access to up-to-date external sources, which collectively lowers the incidence of hallucinations.[1] However, RAG depends on the quality of the documents it finds, which can be a slight problem. If the retrieval step yields inaccurate information, RAG itself cannot discern its veracity, leading the LLM to incorporate it and consequently generate fabricated responses.[24] To fix this, there are methods like Corrective Retrieval-Augmented Generation (CRAG), which check the quality of the retrieved documents and decide whether they are accurate or not.[24] RAG is multimodal, meaning it applies to texts, images, audio, and video, but these create unnecessary challenges as well.[23] RAG has long supported LLMs with accurate information.

3.3. Graph-Based and Structured Input:

Graph-based input methods guide LLMs by organizing information into a clear format.[25,26] Knowledge graphs store facts and help reduce hallucinations through a method called Knowledge Graph-based Retrofitting (KGR).[27] KGR compares the knowledge graph with the initial LLM output and modifies the output to match the facts from the knowledge graph.[27] Graphs can also be used to detect hallucinations by looking at and comparing the relationship between several generated answers. Graph Attention Networks (GANs) identify patterns that separate facts from hallucinations.[28] Graph-based methods are used for tasks that require structured data. These tasks include writing code, research, and even government tasks like planning events, showing how versatile graph-based prompts and graphs are in reducing errors.[29]

3.4. Section Summary:

Context engineering provides LLMs with essential background information, enabling them to answer user prompts with greater accuracy. Techniques like RAG and graph-based inputs each address different problems. RAG addresses the outdated information and provides up-to-date information, while graph-based inputs focus on making sure the data is

DOI: 10.36838/IJHSR811.16

structured and understandable, and use it to improve the accuracy of the output generated by the LLM. The subsequent section will discuss hallucinations and their causative factors.

4. Hallucinations: Causes and Mitigations:

4.1. Overview of Hallucination Causes:

Hallucinations occur when an LLM tries to generate an output not grounded in evidence. But there are many ways for hallucinations to happen, including missing or biased data within the training set, internal model reasoning errors, or a misinterpretation of the user's intent.[30] By examining these causes, we can gain a deeper understanding of why hallucinations occur and maybe even how to prevent them. Hallucinations generally pose a significant threat, impeding the trustworthiness of LLMs as their integration expands into critical fields such as healthcare, law, and education.

4.2. Training Data Limitations:

Many hallucination cases originate from problems within the data used to train LLMs.[29] If bias exists in the training data or if it is incomplete, the model will produce incorrect answers.[29] A study found that the standard knowledge conversation benchmark consisted of 60% hallucinated responses, which allowed LLMs to train on them and carry on the misinformation.[31] The quality of the training data directly impacts hallucination rates in LLMs, with issues like incompleteness or misinformation contributing to the problem. Furthermore, if the training data is outdated, the model's response will lack factual support, as it relies on old rather than current information. This was exemplified by earlier versions of ChatGPT, whose training data had not been updated since 2021.[32] Any biases in the training data are carried over to the LLM's output, affecting its factual reliability. These limitations highlight the need for updated, accurate, and unbiased training data to make LLM outputs more accurate and reduce the rate of hallucinations.[32]

4.3. Prompt Misinterpretation:

Hallucinations often occur when the LLM has trouble understanding the user query, leading to confident but inaccurate answers.[33] This typically occurs when the user provides ambiguous input to the LLM, prompting it to independently infer missing information.[34] LLMs may also prioritize the aesthetic presentation.[34] Ambiguity is definitely an issue, with more people thinking LLMs are smart enough to understand what the user is saying. This is why there are tools like DecoPrompt that exist, to remove the ambiguity and lack of clarity from the user inputs. When LLMs are presented with false information in a user query, they often assume the information is correct and proceed to generate misleading output that reinforces the false input. This is especially applicable in fields like biomedicine, where LLMs struggle to generate accurate responses, as many users simply paste reports or other material directly into the input.[34]

4.4. Valuable Hallucinations:

Valuable hallucinations in LLMs can be formally defined and analyzed using systematic approaches.[5] Rather than dismissing all hallucinations as flaws, recent work provides clear representations and manual judgments for hallucinations that are realizable and constructive under particular conditions. These propositions are not factual, but could be achieved in the future or in specific scenarios, distinguishing them from otherwise unfaithful, fabricated content. Prompt engineering techniques, such as ReAct prompting with reasoning, self-confidence assessment, and answer verification, make it possible to control and optimize the frequency and type of hallucinations produced by LLMs. Experiments using the Qwen2.5 model and the HalluQA dataset show that systematic control can reduce overall hallucinations by 5.12% and increase the proportion of valuable hallucinations from 6.45% to 7.92%.5 This demonstrates that direct intervention in the model's generative process leads to more outputs that have creative or practical utility, such as ideas for scientific or technological innovation, without introducing unhelpful or misleading information.

4.5. Section Summary:

Hallucinations in LLMs do not stem from a single cause, but rather from a group of factors, including training data limitations and prompt misinformation. By splitting these factors up into their own categories, researchers and developers can design multiple solutions towards reducing hallucinations, instead of only relying on one approach.[1] A deeper dive into these specific factors will provide a better understanding of their contributions towards hallucinations and will also provide a roadmap for developing reliable LLMs that only deliver factual and trustworthy outputs.[35]

■ Synthesis and Recommendations

This review demonstrates that neither prompt engineering nor context engineering alone is sufficient to consistently minimize hallucinations in LLMs. Evidence across recent studies suggests that prompt engineering, especially techniques such as linguistic tuning, DecoPrompt, and Chain-of-Thought, effectively reduces hallucinations caused by ambiguous or misleading inputs. However, prompt engineering is limited by the LMM's base knowledge and may not address gaps or inaccuracies within the model's training data. Context engineering complements prompt-engineering strategies by providing relevant, up-to-date external information through RAG and graph-based data. This enables LLMs to ground their responses in factual data, addressing issues of data bias and knowledge obsolescence. A hybrid technique leveraging the strengths of both approaches is recommended. The most robust framework combines well-crafted prompts with context retrieval mechanisms that supply the LLM with accurate data.

■ Conclusion

No single approach eliminates hallucinations in Large Language Models. However, the combination of prompt engineering and context engineering provides an effective and reliable strategy in mitigating them. This paper has demon-

strated how prompt engineering – focusing on the phrasing, structure, and word choice of input – guides the LLM more effectively. Specifically, Section 2 examined prompt engineering strategies such as linguistic tuning, DecoPrompt, and Chain-of-Thought reasoning, which reduce ambiguity, rewrite problematic prompts, and enable step-by-step logical processing to enhance output reliability. Complementing this, Section 3 delved into context engineering techniques, including Retrieval Augmented Generation (RAG) and graph-based inputs. These methods address knowledge gaps and provide up-to-date, structured information, thereby grounding the LLM's responses in factual data. Section 4 provided a detailed analysis of the multifaceted causes of hallucinations, attributing them primarily to training data limitations and prompt misinterpretation. While a singular solution for mitigating hallucinations remains elusive, the comprehensive review presented here underscores the necessity of a hybrid approach tailored to specific fields and tasks. Moving forward, critical questions for future research include the real-time self-detection of hallucinations by models and the development of a classification framework to distinguish between safe and dangerous forms of hallucination.

■ Acknowledgments

I want to thank my research mentor, Dr. Siddarth Krishnan, and my teaching assistant, Plinio Zanini, for their guidance and support throughout this invaluable journey.

■ References

1. Tonmoy, S. M. T. I.; Zaman, S. M. M.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; Das, A. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. arXiv January 3, 2024. https://doi.org/10.48550/arXiv.2401.01313.
2. Barkley, L.; Merwe, B. van der. Investigating the Role of Prompting and External Tools in Hallucination Rates of Large Language Models. arXiv October 25, 2024. https://doi.org/10.48550/arXiv.2410.19385.
3. Xu, N.; Ma, X. DecoPrompt : Decoding Prompts Reduces Hallucinations When Large Language Models Meet False Premises. arXiv November 12, 2024. https://doi.org/10.48550/arXiv.2411.07457.
4. Mei, L.; Yao, J.; Ge, Y.; Wang, Y.; Bi, B.; Cai, Y.; Liu, J.; Li, M.; Li, Z.-Z.; Zhang, D.; Zhou, C.; Mao, J.; Xia, T.; Guo, J.; Liu, S. A Survey of Context Engineering for Large Language Models. arXiv July 21, 2025. https://doi.org/10.48550/arXiv.2507.13334.
5. Chen, Q.; Wang, B. Valuable Hallucinations: Realizable Non-Realistic Propositions. arXiv February 18, 2025. https://doi.org/10.48550/arXiv.2502.11113.
6. White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D. C. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. arXiv February 21, 2023. https://doi.org/10.48550/arXiv.2302.11382.
7. Rathod, J. D. Systematic Study of Prompt Engineering. *Int. J. Res. Appl. Sci. Eng. Technol.* 2024, *12* (6), 597–613. https://doi.org/10.22214/ijraset.2024.63182.
8. Muktadir, G. M. A Brief History of Prompt: Leveraging Language Models. (Through Advanced Prompting). arXiv November 28, 2023. https://doi.org/10.48550/arXiv.2310.04438.
9. Pajo, P. *(PDF) Context Engineering: Enhancing Large Language Model Performance Through Comprehensive Contextual Management*. https://www.researchgate.net/publication/393511218_Context_Engineering_Enhancing_Large_Language_Model_Performance_Through_Comprehensive_Contextual_Management (accessed 2025-08-07).
10. Geroimenko, V. Generative AI Hallucinations in Healthcare: A Challenge for Prompt Engineering and Creativity. In *Human-Computer Creativity: Generative AI in Education, Art, and Healthcare*; Geroimenko, V., Ed.; Springer Nature Switzerland: Cham, 2025; pp 321–335. https://doi.org/10.1007/978-3-031-86551-0_17.
11. Sato, K.; Kaneko, H.; Fujimura, M. Reducing Cultural Hallucination in Non-English Languages Via Prompt Engineering for Large Language Models. OSF May 6, 2024. https://doi.org/10.31219/osf.io/4hzya.
12. Joseph, T.; Keneth, M. H. Exploring the Synergy of Grammar-Aware Prompt Engineering and Formal Methods for Mitigating Hallucinations in LLMs. *East Afr. J. Inf. Technol.* **2024**, *7* (1), 188–201. https://doi.org/10.37284/eajit.7.1.2111.
13. Velásquez-Henao, J. D.; Franco-Cardona, C. J.; Cadavid-Higuita, L. Prompt Engineering: A Methodology for Optimizing Interactions with AI-Language Models in the Field of Engineering. *DYNA* **2023**, *90* (230), 9–17.
14. Jha, S.; Jha, S. K.; Lincoln, P.; Bastian, N. D.; Velasquez, A.; Neema, S. Dehallucinating Large Language Models Using Formal Methods Guided Iterative Prompting.
15. Longpre, S.; Perisetla, K.; Chen, A.; Ramesh, N.; DuBois, C.; Singh, S. Entity-Based Knowledge Conflicts in Question Answering. arXiv January 12, 2022. https://doi.org/10.48550/arXiv.2109.05052.
16. Feldman, P.; Foulds, J. R.; Pan, S. Trapping LLM Hallucinations Using Tagged Context Prompts. arXiv June 9, 2023. https://doi.org/10.48550/arXiv.2306.06085.
17. Sarkar, R.; Sarrafzadeh, B.; Chandrasekaran, N.; Rangan, N.; Resnik, P.; Yang, L.; Jauhar, S. K. Conversational User-AI Intervention: A Study on Prompt Rewriting for Improved LLM Response Generation. arXiv June 25, 2025. https://doi.org/10.48550/arXiv.2503.16789.
18. Rawte, V.; Priya, P.; Tonmoy, S. M. T. I.; Zaman, S. M. M.; Sheth, A.; Das, A. Exploring the Relationship between LLM Hallucinations and Prompt Linguistic Nuances: Readability, Formality, and Concreteness. arXiv September 20, 2023. https://doi.org/10.48550/arXiv.2309.11064.
19. Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv March 16, 2025. https://doi.org/10.48550/arXiv.2402.07927.
20. Cleti, M.; Jano, P. Hallucinations in LLMs: Types, Causes, and Approaches for Enhanced Reliability. Open Science Framework, October 21, 2024. https://doi.org/10.31219/osf.io/tj93u.
21. Zhang, H.; Deng, H.; Ou, J.; Feng, C. Mitigating Spatial Hallucination in Large Language Models for Path Planning via Prompt Engineering. *Sci. Rep.* **2025**, *15* (1), 8881. https://doi.org/10.1038/s41598-025-93601-5.
22. Sun, Y.; Liu, Z.; Liu, C.; Pu, B.; Zhang, Z.; Xie, H. Hallucination Mitigation Prompts Long-Term Video Understanding. arXiv June 17, 2024. https://doi.org/10.48550/arXiv.2406.11333.
23. Abootorabi, M. M.; Zobeiri, A.; Dehghani, M.; Mohammadkhani, M.; Mohammadi, B.; Ghahroodi, O.; Baghshah, M. S.; Asgari, E. Ask in Any Modality: A Comprehensive Survey on Multimodal Retrieval-Augmented Generation. arXiv June 2, 2025. https://doi.org/10.48550/arXiv.2502.08826.
24. Yan, S.-Q.; Gu, J.-C.; Zhu, Y.; Ling, Z.-H. Corrective Retrieval Augmented Generation. arXiv October 7, 2024. https://doi.org/10.48550/arXiv.2401.15884.

DOI: 10.36838/IJHSR811.16

25. Chen, K.; Chen, Q.; Zhou, J.; Tao, X.; Ding, B.; Xie, J.; Xie, M.; Li, P.; Zheng, F.; He, L. Enhancing Uncertainty Modeling with Semantic Graph for Hallucination Detection. arXiv April 5, 2025. https://doi.org/10.48550/arXiv.2501.02020.
26. Zhou, J.; Ghaddar, A.; Zhang, G.; Ma, L.; Hu, Y.; Pal, S.; Coates, M.; Wang, B.; Zhang, Y.; Hao, J. Enhancing Logical Reasoning in Large Language Models through Graph-Based Synthetic Data. arXiv December 16, 2024. https://doi.org/10.48550/arXiv.2409.12437.
27. Guan, X.; Liu, Y.; Lin, H.; Lu, Y.; He, B.; Han, X.; Sun, L. Mitigating Large Language Model Hallucinations via Autonomous Knowledge Graph-Based Retrofitting. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38* (16), 18126–18134. https://doi.org/10.1609/aaai.v38i16.29770.
28. Nonkes, N.; Agaronian, S.; Kanoulas, E.; Petcu, R. Leveraging Graph Structures to Detect Hallucinations in Large Language Models. arXiv July 5, 2024. https://doi.org/10.48550/arXiv.2407.04485.
29. Liu, X. A Survey of Hallucination Problems Based on Large Language Models. *Appl. Comput. Eng.* **2024**, *97* (1), 24–30. https://doi.org/10.54254/2755-2721/2024.17851.
30. Reddy, G. P.; Pavan Kumar, Y. V.; Prakash, K. P. Hallucinations in Large Language Models (LLMs).
31. Dziri, N.; Milton, S.; Yu, M.; Zaiane, O.; Reddy, S. On the Origin of Hallucinations in Conversational Models: Is It the Datasets or the Models? arXiv April 17, 2022. https://doi.org/10.48550/arXiv.2204.07931.
32. Ang, T. L.; Choolani, M.; See, K. C.; Poh, K. K. The Rise of Artificial Intelligence: Addressing the Impact of Large Language Models Such as ChatGPT on Scientific Publications. *Singapore Med. J.* **2023**, *64* (4), 219. https://doi.org/10.4103/singaporemedj.SMJ-2023-055.
33. Bruno, A.; Mazzeo, P. L.; Chetouani, A.; Tliba, M.; Kerkouri, M. A. Insights into Classifying and Mitigating LLMs' Hallucinations. arXiv November 14, 2023. https://doi.org/10.48550/arXiv.2311.08117.
34. Aftab, W.; Apostolou, Z.; Bouazoune, K.; Straub, T. Optimizing Biomedical Information Retrieval with a Keyword Frequency-Driven Prompt Enhancement Strategy. bioRxiv April 28, 2024, p 2024.04.23.590746. https://doi.org/10.1101/2024.04.23.590746.
35. Ye, H.; Liu, T.; Zhang, A.; Hua, W.; Jia, W. Cognitive Mirage: A Review of Hallucinations in Large Language Models. arXiv September 13, 2023. https://doi.org/10.48550/arXiv.2309.06794.

Author

Abhinav Tammana is a senior at Los Altos High School. He wants to pursue Computer Science and Business in college. He has a strong love for vibecoding AI apps and a deep passion for AI.