Measuring and Mitigating Hallucinations in Large Language Models: A Multifaceted Approach

Xavier Amatriain
xavier@amatriain.net

March 4, 2024

Abstract

The advent of Large Language Models (LLMs) has ushered in a new era of possibilities in artificial intelligence, yet it has also introduced the challenge of hallucinations—instances where models generate misleading or unfounded content. This paper delves into the multifaceted nature of hallucinations within LLMs, exploring their origins, manifestations, and the underlying mechanics that contribute to their occurrence. We present a comprehensive overview of current strategies and methodologies for mitigating hallucinations, ranging from advanced prompting techniques and model selection to configuration adjustments and alignment with human preferences. Through a synthesis of recent research and innovative practices, we highlight the effectiveness of these approaches in reducing the prevalence and impact of hallucinations. Despite the inherent challenges, the paper underscores the dynamic landscape of AI research and the potential for significant advancements in minimizing hallucinations in LLMs, thereby enhancing their reliability and applicability across diverse domains. Our discussion aims to provide researchers, practitioners, and stakeholders with insights and tools to navigate the complexities of hallucinations in LLMs, contributing to the ongoing development of more accurate and trustworthy AI systems.

1 Introduction

The phenomenon of ”hallucinations” in Large Language Models (LLMs) [1] represents a significant challenge in the field of artificial intelligence, with implications for both theoretical understanding and practical applications. Despite the recent surge in research, including comprehensive surveys and analyses [2, 3, 4], a gap remains in translating these insights into actionable strategies for real-world scenarios. Drawing from extensive literature review and firsthand experience in addressing hallucinations across diverse applications, this paper aims to bridge this gap, offering a nuanced exploration of the mechanisms, implications, and mitigation strategies for hallucinations in LLMs.

2 Hallucinations in Large Language Models: Definition and Context

A hallucination in an LLM is defined as ”the generation of content that is nonsensical or unfaithful to the provided source” (see ”Survey of Hallucination in Natural Language Generation”[5]). This terminology, although rooted in psychological parlance, has been appropriated within the field of artificial intelligence, albeit not without controversy.

The term ”hallucination,” traditionally associated with sensory perceptions in the absence of external stimuli, as per the Wikipedia definition[1], has been repurposed in AI to describe a distinct phenomenon. This linguistic shift has sparked debates centered around three primary concerns. On the one hand, the usage of ’hallucination’ might erroneously imply a semblance of consciousness or perception in LLMs, which operate purely on data-driven patterns rather than sentient perception or imagination. Furthermore, according to critics, the term could obscure the understanding of LLM operations, which are based on statistical patterns and do not involve cognitive processes like ’seeing’ or ’imagining’. Finally, describing AI outputs as ’hallucinations’ may undermine the seriousness of potential risks posed

[1] https://en.wikipedia.org/wiki/Hallucination

by incorrect or misleading information generated by LLMs, particularly in scenarios where users excessively rely on these models without adequate verification. For all these reasons, some researchers have proposed alternatives such as ”confabulations” or ”fabrications” .

The use of ’hallucination’ in AI though predates its current popularity. Early instances can be traced back to a 1996 paper, ”Text Databases and Information Retrieval”[6], and subsequent references in the literature, including works from 2009 and 2014, which discussed ’hallucinating’ topics and translations in AI contexts[7, 8]. This historical exploration underscores the evolution of the term from its psychological origins to a niche meaning within AI.

Dictionaries such as Merriam-Webster have begun to include AI-specific definitions of ’hallucination’, indicating a broader recognition of this terminology within the field. For all these reasons, and for consistency with current literature, we will keep this term in this paper.

2.1 Classification of Hallucinations in LLMs

We distinguish between two primary types of hallucinations in LLMs: intrinsic and extrinsic. Intrinsic hallucinations are characterized by outputs that directly contradict factual information from the source material. For example, an LLM summarizing a Wikipedia page on Paris[2] might erroneously state that its population is 1 million, a clear deviation from verifiable facts in the source. On the other hand, extrinsic hallucinations involve statements that, while not directly refutable by the source material, introduce unverifiable or speculative content. An example would be an LLM suggesting that Paris is home to the most successful soccer team in France, a claim that, although potentially true, cannot be directly verified against the specific source material provided.

The definition of ’source’ in LLM contexts varies with the task. In dialogue-based tasks, it refers to ’world knowledge’, whereas in text summarization, it pertains to the input text itself. This distinction plays a crucial role in evaluating and interpreting hallucinations.

Similarly, the impact of hallucinations is highly context-dependent. For instance, in creative endeavors like poem writing, hallucinations might be deemed acceptable or even beneficial.

2.2 Mechanics of Hallucination in LLMs

LLMs, trained on diverse datasets including the internet, books, and Wikipedia, generate text based on probabilistic models without an inherent understanding of truth or falsity. Recent advancements like instruct tuning and Reinforcement Learning from Human Feedback (RLHF) have attempted to steer LLMs towards more factual outputs, but the fundamental probabilistic nature and its inherent limitations remain. In particular, LLMs are generally trained to predict tokens probabilistically and therefore have no notion of ground truth.

A recent study, “Sources of Hallucination by Large Language Models on Inference Tasks”[9], highlights two key aspects contributing to hallucinations in LLMs: the veracity prior and the relative frequency heuristic, underscoring the complexities inherent in LLM training and output generation.

3 Assessment of Hallucinations in Large Language Models

Understanding and mitigating hallucinations in LLMs necessitate a structured and quantitative approach to their quantification. This section delineates a robust methodology comprising five interconnected steps, designed to measure hallucinations effectively.

The process begins with the selection of grounding data, which serves as a critical reference point for evaluating LLM outputs. The nature of this grounding data is inherently diverse, reflecting the specific requirements of different applications, from the use of resumes in employment-related tasks to the incorporation of search engine results for web-based queries.

Following the establishment of grounding data, the next step involves the development or selection of measurement test sets. These test sets are composed of input-output pairs, potentially including human-LLM interactions, and are crafted to capture a wide range of scenarios. A comprehensive test set includes both a general subset, representing typical use cases, and an adversarial subset, designed from red-teaming exercises to probe the model’s vulnerabilities in high-risk situations.

[2] https://en.wikipedia.org/wiki/Paris

Subsequently, the assessment process requires the extraction of claims made by the LLM. This critical step employs a variety of methods, from manual analysis to rule-based and machine learning algorithms, each offering unique advantages and challenges in identifying and cataloging the assertions generated by the model.

The validation of these claims against the grounding data constitutes the fourth step, wherein the LLM's outputs are meticulously compared with the reference material to evaluate their accuracy and fidelity. This comparison can be executed through both automated systems and manual review, ensuring a thorough assessment of the model's alignment with factual information.

Finally, the assessment culminates in the reporting of metrics, with the "Grounding Defect Rate" being a common measure. This metric quantifies the proportion of responses from the LLM that lack proper grounding in the reference data, offering a clear indicator of the model's propensity for hallucination. Alongside this, additional metrics are explored below to provide a more nuanced analysis of the model's performance and the effectiveness of interventions designed to mitigate hallucinatory outputs.

3.1 Automatic Evaluation of Hallucinations: Metrics and Methodologies

The automatic evaluation of hallucinations in LLMs necessitates both statistical and model-based metrics to capture the multifaceted nature of hallucinated content.

Statistical metrics, such as ROUGE[10] and BLEU[11], serve as foundational tools for assessing text similarity, primarily targeting intrinsic hallucinations. These metrics, by comparing the overlap between the generated text and reference materials, provide an initial gauge of the model's adherence to the provided source. However, their utility extends further when paired with advanced metrics like PARENT[12], PARENT-T[13], and Knowledge F1[14], which are particularly valuable in contexts where structured knowledge bases underpin the source material. Despite their effectiveness, these statistical approaches have inherent limitations, notably in their capacity to fully grasp the syntactic and semantic subtleties inherent in natural language.

To address these limitations and provide a more granular analysis, model-based metrics come into play, offering a diverse array of methodologies.

Information Extraction (IE)-based metrics stand out by distilling complex knowledge into relational tuples for a straightforward comparison with the source content, thus enabling a detailed examination of the model's output in relation to factual accuracy.

Complementing IE-based metrics, QA-based methodologies offer a dynamic framework for evaluating the alignment between generated content and source material. By formulating questions derived from the generated text and assessing the model's responses against the source, these metrics, exemplified by studies such as "Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering"[15], provide insights into the model's understanding and representation of source knowledge.

Further enriching the toolkit, NLI-based metrics employ Natural Language Inference datasets to scrutinize the veracity of hypotheses generated by LLMs, grounded on specific premises drawn from the source. This approach, as highlighted in "Evaluating Groundedness in Dialogue Systems: The BEGIN Benchmark"[16], offers a method for assessing the truthfulness and logical consistency of model outputs.

Lastly, faithfulness classification metrics introduce a specialized layer of evaluation, focusing on the creation of task-specific datasets that facilitate a nuanced and context-sensitive analysis of model outputs. This category of metrics, illustrated by studies such as "Rome was built in 1776: A Case Study on Factual Correctness in Knowledge-Grounded Response Generation"[17], underscores the importance of tailored evaluations in discerning the subtleties of model-generated hallucinations.

Together, these statistical and model-based metrics form a comprehensive framework for the automatic evaluation of hallucinations in LLMs, each contributing unique insights and capabilities to the overarching goal of understanding and mitigating hallucinatory outputs in LLM applications.

3.2 Hallucination Datasets and Benchmarks

The field of hallucination measurement in Large Language Models (LLMs) is burgeoning, characterized by rapid advancements and the increasing availability of specialized datasets and benchmarks. These resources are pivotal for establishing baselines against which the performance and progress of LLMs in mitigating hallucinations can be gauged.

A cornerstone in this developing landscape is the HAllucination DEtection dataset (HADES)[18], which represents an early but significant effort to systematically quantify hallucinations. HADES was constructed by introducing perturbations into raw text sourced from the web, followed by an evaluation process where human judges determined whether these alterations resulted in hallucinations. This methodology not only provides a concrete framework for identifying hallucinations but also sets a precedent for subsequent datasets in this domain.

Building on the foundation laid by HADES, the introduction of HaluEval[19] marks a significant expansion of available resources. HaluEval encompasses a comprehensive dataset featuring 35,000 samples, each classified as either hallucinated or normal, specifically curated for the nuanced evaluation of LLMs. This dataset is instrumental in facilitating a deeper understanding of hallucinations, offering a rich resource for researchers and practitioners alike to test and refine their models. Finally, ”The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations”[20], recently published the HallucInation eLiciTation (HILT), a dataset comprising of 75,000 samples generated using 15 LLMs along with human annotations for six different categories of hallucination proposed by the authors.

Complementing these datasets, initiatives like the Hallucination Leaderboard[3] and the Galileo Hallucination Index[4] provide platforms for comparing the hallucination rates of state-of-the-art LLMs. These efforts not only foster a competitive environment that drives improvements in model performance but also offer transparency and insight into the capabilities and limitations of current LLMs in handling hallucinations.

Together, these datasets and benchmarks form an essential infrastructure for the systematic study of hallucinations in LLMs, enabling the community to track progress, identify challenges, and drive forward the development of more reliable and accurate models.

3.3 Human Evaluation in Hallucination Assessment

Despite the sophistication of automated metrics, the nuanced nature of language and hallucinations necessitates the inclusion of human judgment in the assessment process. Human evaluators bring a level of understanding and interpretation that is currently unmatched by automated systems, making their involvement crucial for a comprehensive evaluation of LLM outputs. This section explores three primary methodologies employed in human evaluation: scoring, comparative analysis, and red teaming.

Scoring involves human evaluators assigning a level of hallucination to LLM outputs on a predefined scale. This method provides a quantifiable measure of the degree to which the generated content deviates from expected norms or factual accuracy. Through scoring, evaluators contribute to a granular understanding of the model's performance, offering insights into the subtleties of its output that might be overlooked by automated systems.

Comparative analysis extends the evaluative process by positioning the LLM-generated content against a set of baseline or ground-truth references. This approach not only facilitates a direct comparison of the LLM's outputs with established standards but also introduces an essential layer of subjective assessment. Evaluators are tasked with discerning the fidelity of the content to the source material, thereby gauging the model's ability to generate coherent and contextually appropriate responses.

It is important to highlight that publicly available datasets such as the ones described in section 3.2.

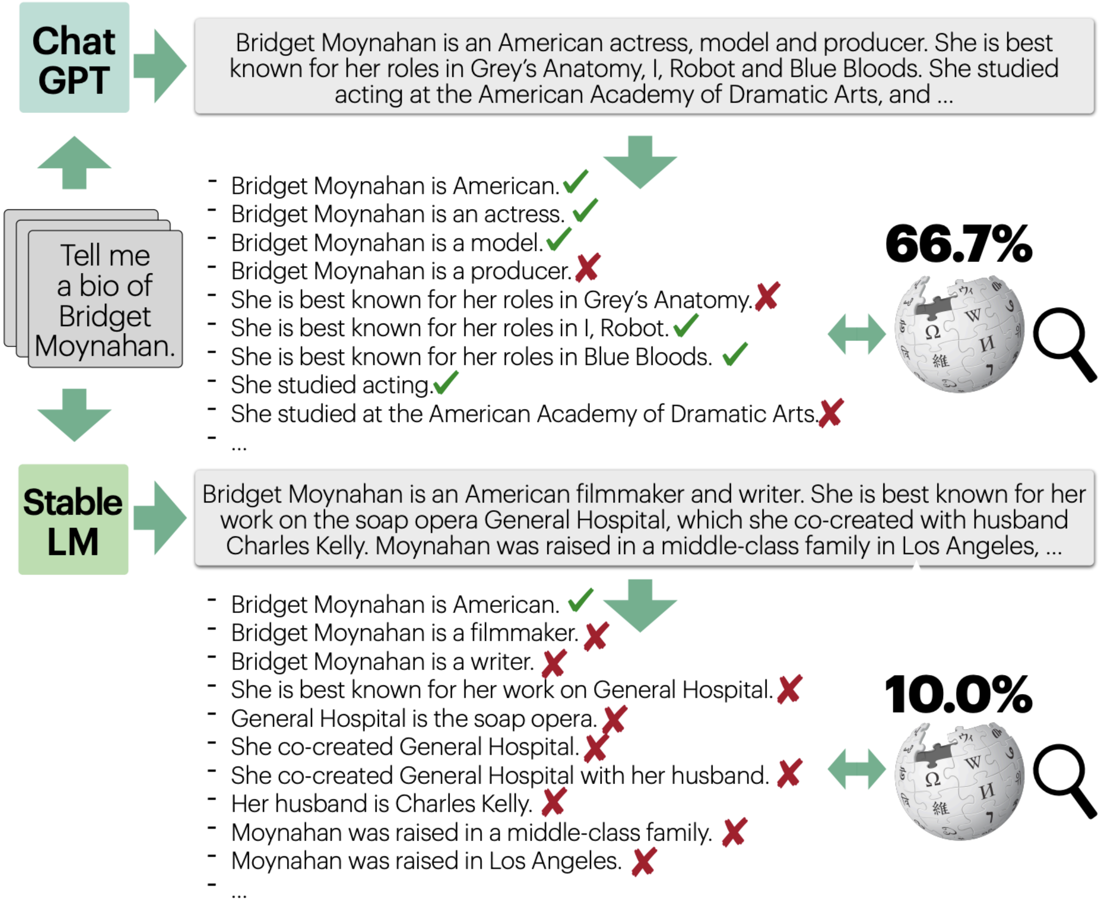
3.3.1 The Example of FActScore

A notable advancement in the domain of human evaluation is the FActScore[21], a metric designed to bridge the gap between human and automated assessments. FActScore deconstructs LLM-generated content into discrete ”atomic facts,” each evaluated for its accuracy against the source material. The metric assigns a binary accuracy value to each atomic fact, which is then aggregated to derive the final score. This approach ensures that each fact is weighted equally, providing a balanced and comprehensive assessment of the content's fidelity to the source.

The implementation of FActScore involves various automation strategies, leveraging LLMs to approximate the human evaluative process. This integration of human judgment with model-based assessments exemplifies the collaborative potential between human evaluators and LLMs, aiming to enhance the reliability and accuracy of hallucination detection.

[3] https://github.com/vectara/hallucination-leaderboard/

[4] https://www.rungalileo.io/hallucinationindex



Tell me a bio of Bridget Moynahan.

Bridget Moynahan is an American actress, model and producer. She is best known for her roles in Grey's Anatomy, I, Robot and Blue Bloods. She studied acting at the American Academy of Dramatic Arts, and ...

- Bridget Moynahan is American. ✓
- Bridget Moynahan is an actress. ✓
- Bridget Moynahan is a model. ✓
- Bridget Moynahan is a producer. ✗
- She is best known for her roles in Grey's Anatomy. ✗
- She is best known for her roles in I, Robot. ✓
- She is best known for her roles in Blue Bloods. ✓
- She studied acting. ✓
- She studied at the American Academy of Dramatic Arts. ✗
- ...

Bridget Moynahan is an American filmmaker and writer. She is best known for her work on the soap opera General Hospital, which she co-created with husband Charles Kelly. Moynahan was raised in a middle-class family in Los Angeles, ...

- Bridget Moynahan is American. ✓
- Bridget Moynahan is a filmmaker. ✗
- Bridget Moynahan is a writer. ✗
- She is best known for her work on General Hospital. ✗
- General Hospital is the soap opera. ✗
- She co-created General Hospital. ✗
- She co-created General Hospital with her husband. ✗
- Her husband is Charles Kelly. ✗
- Moynahan was raised in a middle-class family. ✗
- Moynahan was raised in Los Angeles. ✗
- ...

Figure 1: The FActScore metric

3.3.2 Red Teaming

Red teaming stands as a pivotal adversarial testing strategy, where trained human evaluators rigorously challenge LLMs to expose potential vulnerabilities, including susceptibility to generating hallucinations. This method, while applicable across various objectives such as minimizing jailbreaking attempts, is particularly valuable in providing a critical layer of scrutiny beyond systematic measurements in hallucination assessment.

To maximize the effectiveness of red teaming, adhering to a set of best practices is essential. It is crucial to perceive red teaming and stress-testing as complementary tools that enhance, rather than replace, systematic measurements. These practices are designed to augment the robustness of LLM evaluations by introducing an adversarial perspective, ensuring a comprehensive understanding of model performance and resilience.

Conducting tests in conditions that closely mimic real-world scenarios, particularly on production endpoints, offers invaluable insights into the model's behavior in actual use cases. This approach ensures that the findings are relevant and applicable to the environments in which the LLMs will operate.

The definition of potential harms and the establishment of clear testing guidelines are fundamental steps in preparing for red teaming. These guidelines ensure that all participants share a common understanding of the objectives and ethical boundaries of the testing process, thereby aligning efforts and focusing on predefined areas of interest.

Prioritization of focus areas within the red teaming exercises allows for targeted investigations, directing resources and attention towards the most critical features, harms, and scenarios. This focused approach not only enhances efficiency but also increases the likelihood of uncovering significant insights.

The diversity and expertise of the testers play a crucial role in the depth and breadth of the evaluation. A team comprising individuals from varied backgrounds and with diverse skill sets can approach the task from multiple perspectives, enriching the assessment with a wide range of insights and uncovering biases that might otherwise go unnoticed.

Documentation throughout the red teaming process is indispensable, providing a structured framework for capturing findings and facilitating subsequent analysis. Clear, detailed records of the testing methodologies, scenarios, and outcomes form the backbone of an effective evaluation, enabling the systematic review and replication of tests.

Moreover, managing the time and well-being of testers is of paramount importance. Establishing reasonable time commitments and being mindful of the potential for burnout are critical in maintaining the quality and creativity of the testing efforts. Adequate planning and support ensure that testers remain engaged and productive throughout the process.

Emerging strategies in red teaming, such as leveraging LLMs to test other LLMs, represent innovative approaches to enhancing the rigor and efficiency of these assessments. For instance, initiatives like DeepMind's exploration of "Red Teaming Language Models with Language Models"[22] underscore the potential of using AI to refine and strengthen the robustness of LLMs against hallucinations and other challenges.

In sum, red teaming is an indispensable component of a comprehensive LLM evaluation strategy, offering unique insights and fostering improvements in model resilience and reliability. By adhering to these best practices, researchers and practitioners can harness the full potential of red teaming to advance the state of the art in LLM development and deployment.

4 Strategies for Mitigating Hallucinations in Large Language Models

The inherent complexity of LLMs and their training processes means that hallucinations, to some extent, are an inevitable occurrence[23]. Recognizing this, the focus shifts from attempting to eliminate hallucinations entirely to effectively mitigating their frequency and impact. Mitigation requires a nuanced, multifaceted approach, tailored to the specific needs and contexts of various LLM applications. This section outlines several practical strategies aimed at reducing the prevalence of hallucinations and their potential to mislead or cause harm.

The figure above illustrates the complexity of the mitigation process, emphasizing that no single strategy suffices; rather, a combination of approaches is necessary to address the diverse manifestations of hallucinations in LLMs. These strategies range from technical interventions in the model's training process to procedural safeguards implemented during its deployment and use. The following subsections will delve into these strategies in detail, highlighting how they can be applied individually and in concert to minimize the adverse effects of hallucinations.

4.1 Product Design and User Interaction Strategies

Mitigating hallucinations in Large Language Models (LLMs) commences at the foundational stage of use case and product design, necessitating a strategic approach that integrates user interaction considerations to minimize the occurrence and impact of hallucinations.

At the use case design level, the focus is on configuring applications in a manner that intrinsically diminishes the risk of hallucinations. This might involve, for example, orienting the application towards generating opinions rather than factual content in scenarios where subjective interpretations are less prone to inaccuracies.

Transitioning to product-level strategies, several recommendations have emerged as effective in enhancing the reliability and accountability of AI-generated content. Enabling user editability stands out as a critical feature, allowing users to directly modify AI-generated content, thereby instilling a layer of human scrutiny and intervention that can catch and correct hallucinations.

Emphasizing user responsibility is another pivotal strategy, where users are made aware of their role in reviewing and ensuring the accuracy of the content they generate and disseminate using the LLM. This approach fosters a culture of accountability and vigilance among users.

Incorporating citations and references within the generated content offers a tangible means for users to verify information, thereby promoting transparency and trust in the AI's outputs. Operational modes, such as a "precision" mode, present a trade-off between computational efficiency and accuracy, giving users the choice to prioritize depth and correctness of content when needed.

User feedback mechanisms play an essential role in the continuous improvement of LLMs. By enabling users to flag inaccuracies or hallucinations, developers can gather valuable insights to refine the model and enhance its reliability.

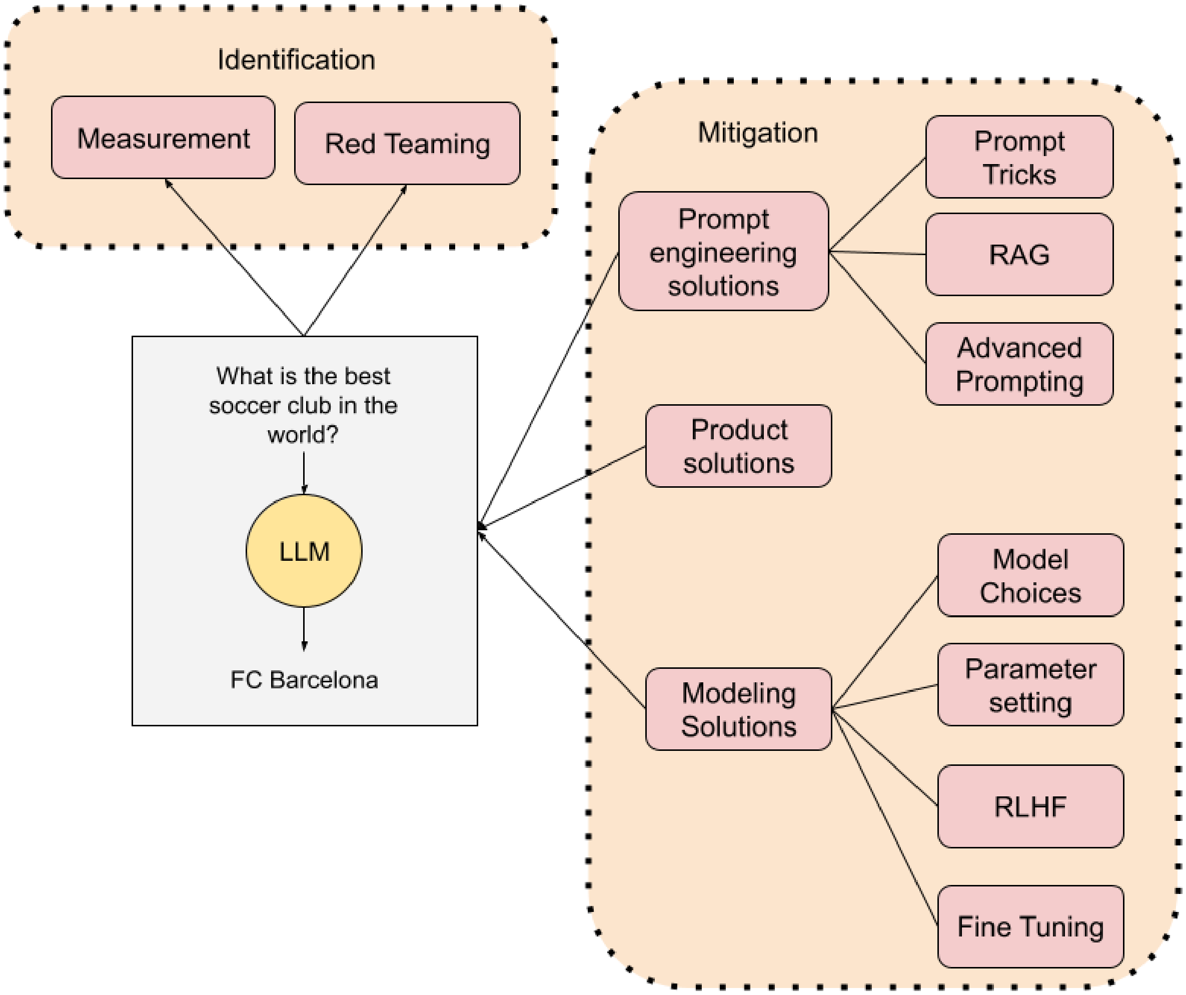


Figure 2: Mitigating hallucinations requires a multifaceted approach

Controlling the length and complexity of AI-generated responses can also serve as a preventive measure against hallucinations, as shorter and simpler outputs are typically less susceptible to generating erroneous content.

Finally, employing structured input and output formats, especially in applications like resume generation, can significantly mitigate the risk of hallucinations by confining the AI's responses to predefined parameters and reducing the scope for inaccuracies.

Together, these product design and user interaction strategies form a comprehensive approach to mitigating hallucinations in LLMs, emphasizing the synergy between technological sophistication and user-centric design principles.

4.2 Data Management and Continuous Improvement

A cornerstone of effectively mitigating hallucinations in Large Language Models (LLMs) lies in the meticulous management of data and the commitment to continuous improvement of the models. Establishing and rigorously maintaining a tracking set dedicated to hallucinations plays a pivotal role in this process. Such a tracking set not only serves as a repository of instances where the LLM has generated hallucinated content but also functions as a crucial resource for analyzing patterns, identifying recurring issues, and guiding targeted interventions to enhance the model's performance.

The management of this data, however, brings to the forefront the critical importance of adhering to stringent data privacy and security best practices. Given the potentially sensitive nature of the content within the tracking set, ensuring the confidentiality, integrity, and availability of this data is paramount. This involves implementing robust security measures, from encryption and access controls to regular audits and compliance checks, to safeguard against unauthorized access and potential data breaches.

Moreover, the ethos of continuous improvement underpins the entire process of managing hallucinations in LLMs. By leveraging the insights gleaned from the tracking set, developers and researchers can iterate on the model's design, training methodologies, and operational parameters. This iterative process, grounded in empirical evidence and guided by best practices in data management, fosters a cycle of ongoing refinement and enhancement, ensuring that LLMs become progressively more adept at minimizing hallucinations while maintaining high standards of data privacy and security.

4.3 Prompt Engineering and Metaprompt Design

The nuanced practice of prompt engineering and the strategic construction of metaprompts are critical in optimizing the functionality of Large Language Models (LLMs)[24], particularly in anchoring their responses and diminishing the occurrence of hallucinations. Insights from recent studies illuminate the profound influence that well-crafted directives within metaprompts exert on mitigating hallucinatory outputs. By delineating the bounds of acceptable responses and channeling the LLM towards more accurate and relevant outputs, we can significantly curtail the rates of hallucinations.

Central to the efficacy of metaprompts is their capacity to convey explicit instructions to the LLMs. An adeptly designed metaprompt not only delineates what the model should avoid but also propels it towards viable alternatives, thereby enriching the fidelity and pertinence of its responses. This approach of simultaneously imposing constraints and offering direction proves to be a potent method in securing the LLM's adherence to the grounded reality.

4.3.1 General Guidelines to Curb Hallucinations

In the realm of prompt engineering and metaprompt design, several guiding principles emerge as instrumental in reducing hallucinations:

Simplifying complex tasks into more digestible components enables LLMs to better comprehend and execute instructions, leading to outputs that are both coherent and closely aligned with the intended context. Employing the innate functionalities and features embedded within metaprompts can significantly enhance the model's grasp and fulfillment of tasks. Incorporating examples within prompts, a technique known as few-shot learning, provides the model with concrete references, clarifying the expected format and substance of the response.

Iterative refinement of the metaprompt, based on the model's performance, allows for continuous enhancement of the quality and relevance of the generated content (see next prompt fine-tuning below). It is imperative to recognize that while these strategies are pivotal in bolstering the grounding of LLM outputs, they invariably introduce computational demands. Thus, when integrating LLMs into product designs, it becomes crucial to judiciously navigate the balance between augmenting grounding accuracy and maintaining computational efficiency. Striking this balance is essential for the effective and sustainable deployment of LLMs in various applications.

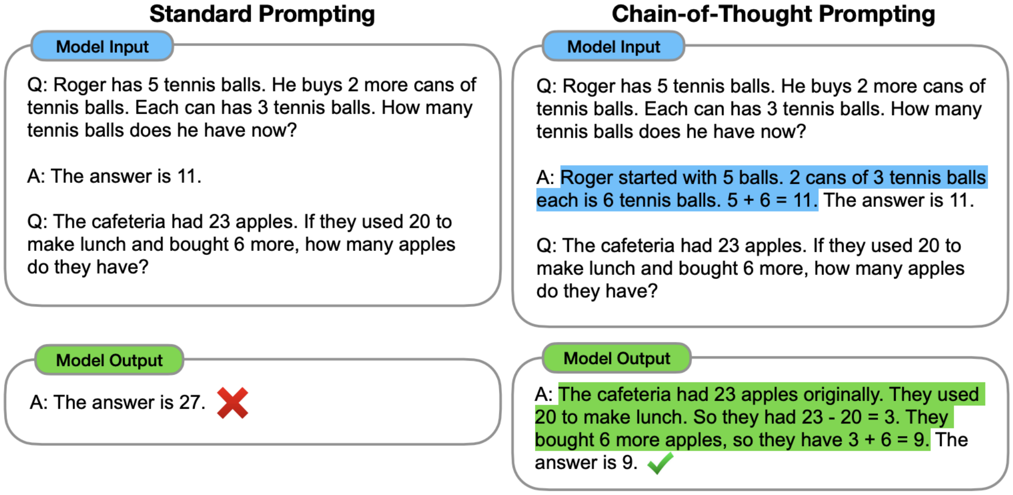
4.3.2 Fine-Tuning Your Prompts

Fine-tuning prompts involves an iterative approach to enhancing the Large Language Model's (LLM's) adherence to directives and improving the grounding of its responses. Adopting an assertive tone, for instance, can significantly influence model compliance. Emphasizing certain directives, possibly through the use of ALL CAPS or highlighting, can draw the model's attention to key instructions, thereby improving its performance.

Understanding that context is paramount, providing the LLM with ample background information can serve to better anchor its responses in the relevant factual framework. This foundational context enables the model to generate outputs that are more aligned with the given scenario.

The process of refinement is iterative, necessitating a reevaluation of the model's initial outputs and the subsequent adjustment of prompts to hone the desired outcome. Inline citations represent a strategic tool in this process, prompting the model to substantiate its claims and thereby fostering a layer of accountability and verifiability in its responses.

Framing the task at hand can also have a profound impact on the results. Tasks framed as summarization activities, for example, tend to yield more grounded outputs than those approached from a question-answering perspective. This distinction underscores the importance of carefully considering how tasks are presented to the model.



Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27. ✗

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 3: Chain of Thought compared to standard prompting

Selective grounding is another critical consideration, requiring a discernment of scenarios where grounding is imperative versus those where it may be less critical. This selective approach ensures that computational resources are allocated efficiently, focusing on grounding where it most significantly enhances the output's value.

Reiterating key points towards the end of the prompt can serve to reinforce their importance, ensuring that essential instructions are not overlooked. Echoing vital details from the input within the prompt itself can further ensure that the model's output remains closely aligned with the source data.

Algorithmic filtering can be employed to navigate through the vast information processed by the model, prioritizing data that is most relevant to the task at hand. This targeted approach aids in focusing the model's attention and resources on information that most significantly influences the quality of its output.

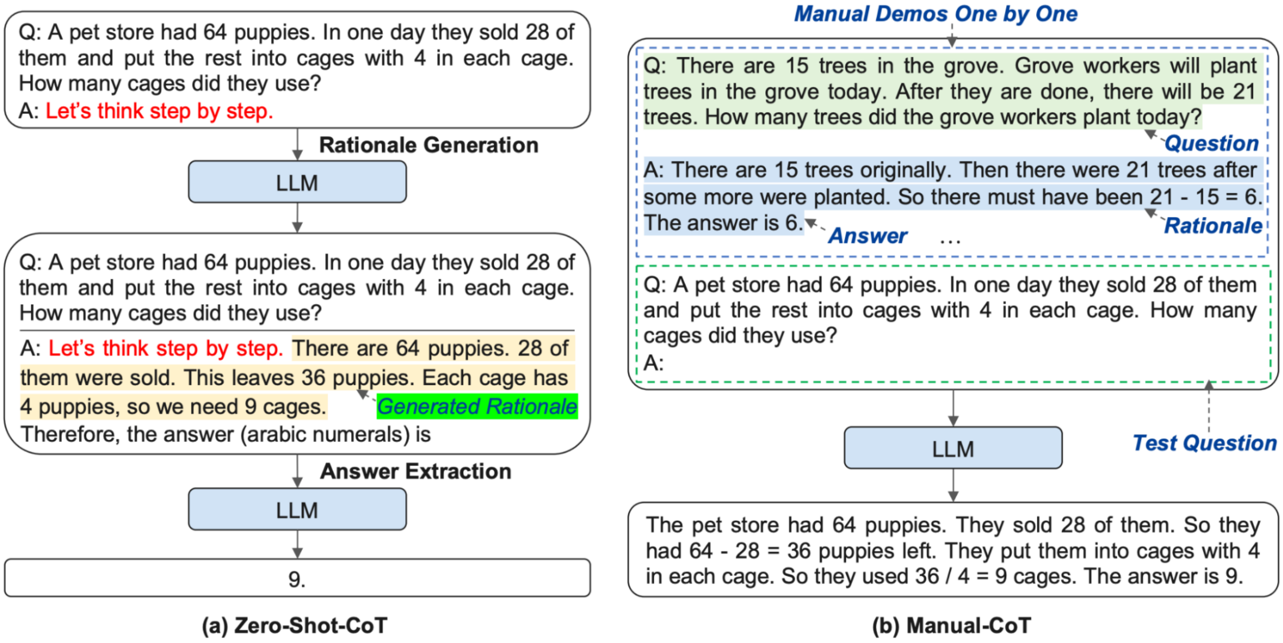
In the subsequent sections, we will explore advanced prompting techniques, including the 'chain of thought' approach, and examine how Retrieval-Augmented Generation (RAG) can be utilized to achieve more effectively grounded results. These advanced strategies offer a glimpse into the evolving landscape of prompt engineering, promising new avenues for enhancing the fidelity and utility of LLM outputs.

4.3.3 Chain of Thought

The concept of Chain of Thought (CoT) was introduced in the paper "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" by researchers at Google[25]. This innovative approach is predicated on the understanding that Large Language Models (LLMs) are fundamentally designed to predict the next sequence of tokens rather than to engage in explicit reasoning processes. By delineating the necessary reasoning steps within the prompt, however, it becomes possible to steer LLMs towards a more reasoned and logical output, aligning them closer to the cognitive processes they aim to emulate (see Figure 3).

A pivotal aspect of CoT is the distinction between its implementation methods. The so-called "Manual CoT" involves explicitly providing examples of the reasoning steps, highlighted in blue in the illustrative examples in Figure 4, guiding the LLM through the thought process. This method contrasts with the "zero-shot CoT," where the model is simply instructed to "think step by step" without concrete examples. While the manual approach tends to yield more effective results by offering clear reasoning templates, it faces challenges in scalability and maintenance due to the need for tailored examples.

The CoT methodology embodies a structured extension of the general recommendation to "simplify complex tasks." By breaking down the reasoning process into discrete, manageable steps, CoT not only facilitates a more logical and coherent output from LLMs but also significantly mitigates the incidence of hallucinations across various applications. This approach underscores the potential of prompt engineering to enhance the reasoning capabilities of LLMs, making it a valuable tool in the broader strategy to minimize hallucinatory content.



Q: A pet store had 64 puppies. In one day they sold 28 of them and put the rest into cages with 4 in each cage. How many cages did they use?
A: Let's think step by step.

Q: A pet store had 64 puppies. In one day they sold 28 of them and put the rest into cages with 4 in each cage. How many cages did they use?
A: Let's think step by step. There are 64 puppies. 28 of them were sold. This leaves 36 puppies. Each cage has 4 puppies, so we need 9 cages.
Therefore, the answer (arabic numerals) is

(a) Zero-Shot-CoT

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Q: A pet store had 64 puppies. In one day they sold 28 of them and put the rest into cages with 4 in each cage. How many cages did they use?
A:

The pet store had 64 puppies. They sold 28 of them. So they had 64 - 28 = 36 puppies left. They put them into cages with 4 in each cage. So they used 36 / 4 = 9 cages. The answer is 9.

(b) Manual-CoT

Figure 1: Zero-Shot-CoT [Kojima et al., 2022] (using the "Let's think step by step" prompt) and Manual-CoT [Wei et al., 2022a] (using manually designed demonstrations one by one) with example inputs and outputs of an LLM.

Figure 4: Zero-shot vs. Manual Chain of Thought

4.3.4 Grounding with RAG

Retrieval-Augmented Generation (RAG) represents a pivotal advancement in enhancing the capabilities of Large Language Models (LLMs). Introduced by Facebook in 2020, specifically in relation to their BART model[26], RAG has been adopted widely, including its integration into the Hugging Face library, marking a significant milestone in the evolution of LLMs.

The Core Concept At its core, RAG is designed to synergize a retrieval component with a generative model, allowing for a harmonious interplay between the two. This innovative approach is depicted in Figure 5. Note though that current approaches to RAG operate in a zero-shot setting, typically without the need for additional training or fine-tuning illustrated in that early instance of RAG.

By integrating retrieval capabilities, RAG empowers LLMs to draw upon external sources of information, significantly grounding the generated content in verifiable data. The retrieval component sources pertinent information, which is then artfully woven into the model's output, ensuring responses are not only coherent but also contextually grounded.

As RAG continues to mature, it has become a cornerstone technique for prompt engineers, particularly in its application to more complex scenarios. It stands as a testament to the potential of combining simple retrieval mechanisms with the generative prowess of LLMs to enhance content reliability and mitigate the risk of hallucinations.

RAG known caveats and guardrails Despite its efficacy, RAG is not without its challenges, particularly the risk of over-reliance on retrieved results, which can inadvertently lead to inaccuracies or hallucinations. The key to navigating these pitfalls lies in recognizing and strategically addressing potential shortcomings.

For instances where retrieval yields no results, prompt designs must incorporate fallback strategies, such as politely declining to answer while suggesting alternative query formulations. This approach helps maintain user trust by transparently acknowledging the model's current limitations.

Ambiguous queries require a different tactic, emphasizing the need for clarification. In scenarios where a query could be interpreted in multiple ways, it's prudent to engage the user in refining their request, thereby reducing ambiguity and enhancing the relevance of the response.

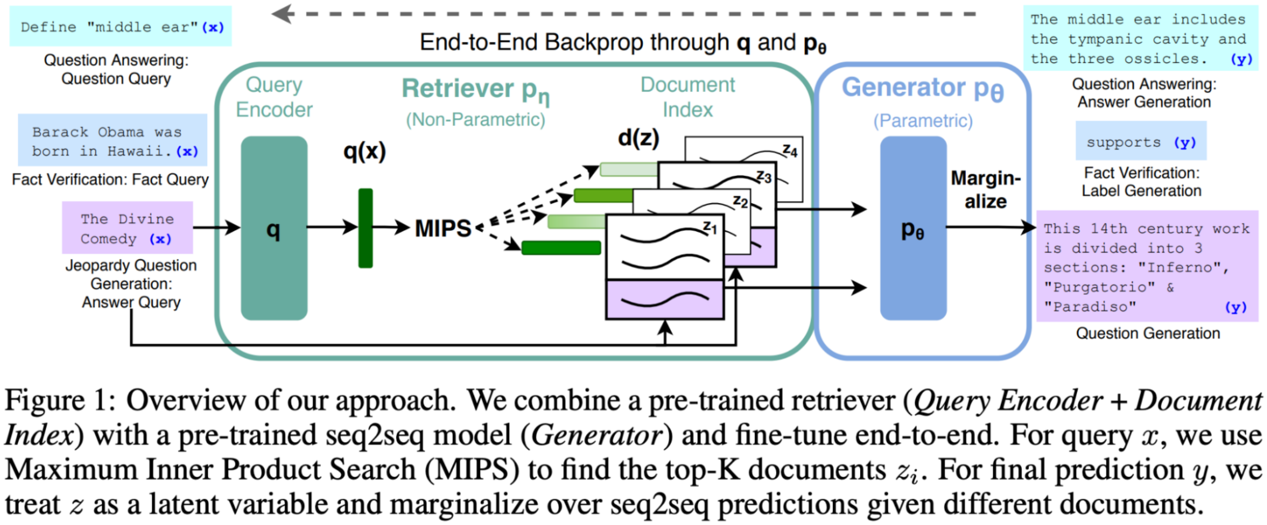

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and p_θ
Query Encoder
Retriever p_η
(Non-Parametric)
Document Index
q
q(x)
MIPS
d(z)
z_4
z_3
z_2
z_1
Generator p_θ
(Parametric)
p_θ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

Figure 5: Retrieval Augmented Generation can be used for mitigating hallucinations

Addressing incorrect retrieval results poses a more complex challenge, as it demands a nuanced understanding of the retrieval engine's performance within specific contexts. Continuous analysis and prompt optimization in areas known for retrieval inaccuracies are crucial for minimizing the impact of erroneous information on the model's outputs.

By implementing these strategies, RAG can be effectively utilized to ground LLM responses, reducing the likelihood of hallucinations while ensuring the model remains responsive and informative, even in the face of imperfect retrieval outcomes.

4.4 Advanced Prompt Engineering Techniques

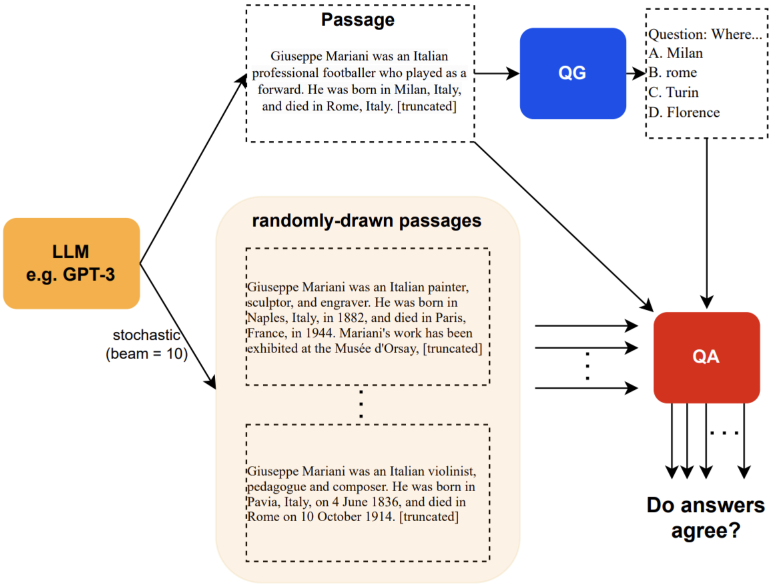
In recent months, the quest to address hallucinations and improve grounding in Large Language Models (LLMs) has spurred a wave of innovation, yielding a suite of advanced prompt engineering techniques. These methodologies represent a significant departure from the foundational strategies previously discussed, delving deeper into the nuanced interplay between prompt design and model output quality. For those seeking a deeper dive into the realm of prompt engineering, I recommend exploring my earlier work, ”Prompt Engineering 201: Advanced methods and toolkits,” which offers a comprehensive overview of these sophisticated approaches.

One of the critical considerations when employing advanced prompt engineering techniques is the inherent trade-offs they present, notably in terms of complexity, latency, and computational cost. These methods often necessitate multiple interactions with the LLM, which can introduce delays and elevate expenses. Despite these challenges, the potential benefits these techniques offer in terms of enhanced grounding and diminished hallucinations can be compelling, particularly in applications where accuracy and reliability are paramount.

Furthermore, the evolving landscape of LLMs presents opportunities to leverage these advanced techniques in conjunction with smaller, more efficient models. This synergy can strike a balance between achieving high-quality outputs and maintaining manageable operational costs. By integrating these sophisticated prompt engineering strategies into your toolkit, you can navigate the complexities of LLM applications more effectively, making informed decisions about when and how to apply these methods to optimize performance without incurring prohibitive costs.

4.4.1 Self-consistency

The concept of self-consistency, as delineated in ”SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models”[27], introduces a novel approach whereby a Large Language Model (LLM) is utilized to cross-verify its own outputs. This ensemble-based strategy involves prompting the LLM to produce multiple responses to the same query, with the underlying premise that a higher degree of consistency among these responses serves as an indicator of their reliability.



Giuseppe Mariani was an Italian professional footballer who played as a forward. He was born in Milan, Italy, and died in Rome, Italy. [truncated]

Question: Where...
A. Milan
B. rome
C. Turin
D. Florence

Giuseppe Mariani was an Italian painter, sculptor, and engraver. He was born in Naples, Italy, in 1882, and died in Paris, France, in 1944. Mariani's work has been exhibited at the Musée d'Orsay, [truncated]

Giuseppe Mariani was an Italian violinist, pedagogue and composer. He was born in Pavia, Italy, on 4 June 1836, and died in Rome on 10 October 1914. [truncated]

Figure 6: Illustration of the self-consistency approach in a question-answering scenario

As illustrated in Figure 6, this method is particularly effective in question-answering contexts, where ”consistency” is quantified by the alignment of answers across different instances. The original study expands on this by introducing additional metrics for consistency assessment, including BERT-scores and n-gram overlaps, alongside a composite measure that amalgamates these individual metrics, offering a multifaceted view of response accuracy.

4.4.2 Reason and Act (ReAct)

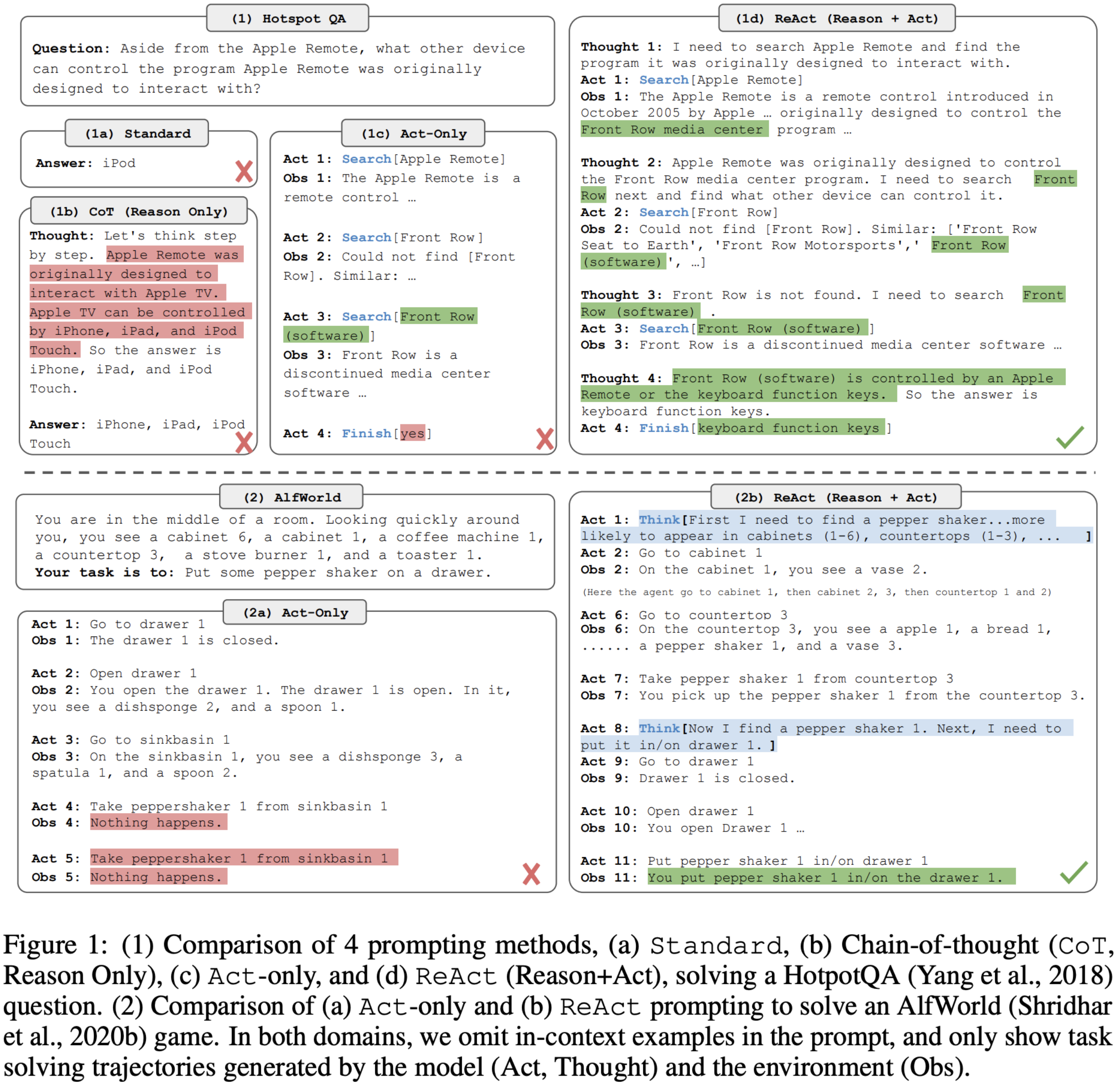
The ReAct methodology, detailed in ”ReAct: Synergizing Reasoning and Acting in Language Models”[28], proposes a structured framework for LLMs that interleaves verbal reasoning with actionable steps. This approach not only fosters dynamic reasoning within the model but also enhances the coherence and groundedness of its outputs, as depicted in Figure 7.

A significant advantage of ReAct is its potential to curtail hallucinatory content, particularly when compared to earlier methodologies such as Chain of Thought (CoT). The integration of reasoning and action sequences not only contributes to a more grounded and trustworthy model output but also introduces a structured format that the LLM can follow. It’s noteworthy, however, that this structured approach may slightly constrain the model’s reasoning flexibility, a trade-off that underscores the intricate balance between enhancing model reliability and preserving its adaptability.

4.4.3 Reflection

Building on the concept of self-consistency, where the similarity of responses to the same query suggests confidence, the technique of Reflection delves into a more direct interrogation of an LLM’s confidence in its own outputs. As highlighted by Eric Jang, emerging evidence suggests that models like GPT-4 may possess the capability to critically assess and revise their outputs based on an internal evaluation of their logical soundness.

The approach, as detailed in the ”Reflection” paper[29], is characterized as ”reinforcement via verbal reflection.” This method involves an LLM, acting as both the actor and the evaluator, generating a hypothesis and then assessing its validity. A self-reflection component synthesizes a summary of this assessment, which is iteratively refined until the evaluator deems the response sufficiently accurate. This reflective process has been shown to significantly enhance the model’s ability to identify and rectify hallucinations, even outperforming the ReAct framework in certain instances.



(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod ✗

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch ✗

(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control …

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: …

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Act 4: Finish[yes] ✗

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys] ✓

(2) AlfWorld

You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 1, a coffee machine 1, a countertop 3, a stove burner 1, and a toaster 1.
Your task is to: Put some pepper shaker on a drawer.

(2a) Act-Only

Act 1: Go to drawer 1
Obs 1: The drawer 1 is closed.

Act 2: Open drawer 1
Obs 2: You open the drawer 1. The drawer 1 is open. In it, you see a dishsponge 2, and a spoon 1.

Act 3: Go to sinkbasin 1
Obs 3: On the sinkbasin 1, you see a dishsponge 3, a spatula 1, and a spoon 2.

Act 4: Take peppershaker 1 from sinkbasin 1
Obs 4: Nothing happens.

Act 5: Take peppershaker 1 from sinkbasin 1
Obs 5: Nothing happens. ✗

(2b) ReAct (Reason + Act)

Act 1: Think[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.

(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)

Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, ... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

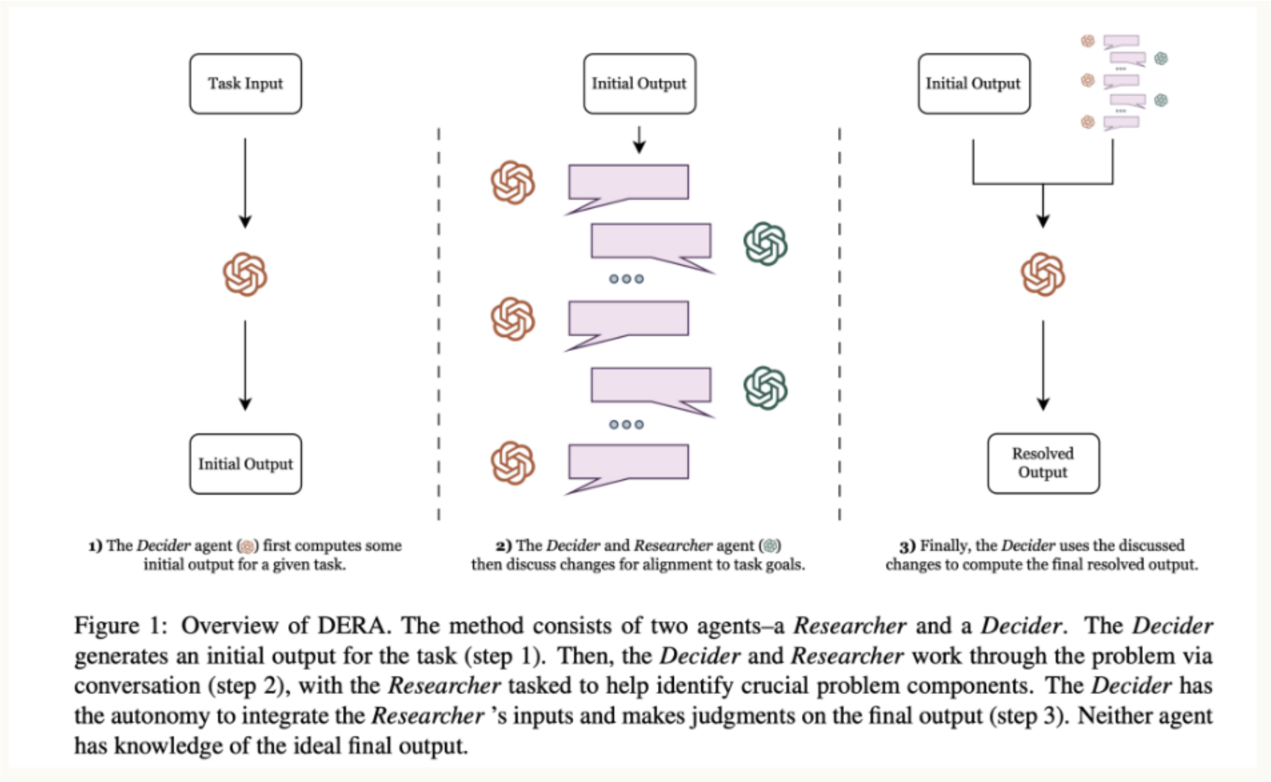
Act 8: Think[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1. ✓

Figure 1: (1) Comparison of 4 prompting methods, (a) Standard, (b) Chain-of-thought (CoT, Reason Only), (c) Act-only, and (d) ReAct (Reason+Act), solving a HotpotQA (Yang et al., 2018) question. (2) Comparison of (a) Act-only and (b) ReAct prompting to solve an AlfWorld (Shridhar et al., 2020b) game. In both domains, we omit in-context examples in the prompt, and only show task solving trajectories generated by the model (Act, Thought) and the environment (Obs).

Figure 7: The ReAct approach to integrating reasoning and action in LLM outputs



1) The *Decider* agent (●) first computes some initial output for a given task.

2) The *Decider* and *Researcher* agent (●) then discuss changes for alignment to task goals.

3) Finally, the *Decider* uses the discussed changes to compute the final resolved output.

Figure 1: Overview of DERA. The method consists of two agents–a *Researcher* and a *Decider*. The *Decider* generates an initial output for the task (step 1). Then, the *Decider* and *Researcher* work through the problem via conversation (step 2), with the *Researcher* tasked to help identify crucial problem components. The *Decider* has the autonomy to integrate the *Researcher* 's inputs and makes judgments on the final output (step 3). Neither agent has knowledge of the ideal final output.

Figure 8: Illustration of Dialog-Enabled Resolving Agents (DERA) in action

4.4.4 Dialog-Enabled Resolving Agents (DERA)

The DERA framework[30] introduces a novel dialog-based approach tailored for high-stakes domains such as healthcare. It delineates distinct agent roles within a dialog, including "Researchers" who explore various facets of a query in parallel and a "Decider" who synthesizes their findings into a conclusive response. This parallel processing contrasts with the sequential nature of reflection in the Reflexion model, offering a dynamic and collaborative approach to problem-solving and decision-making, as illustrated in Figure 8.

4.4.5 Chain-of-Verification (COVE)

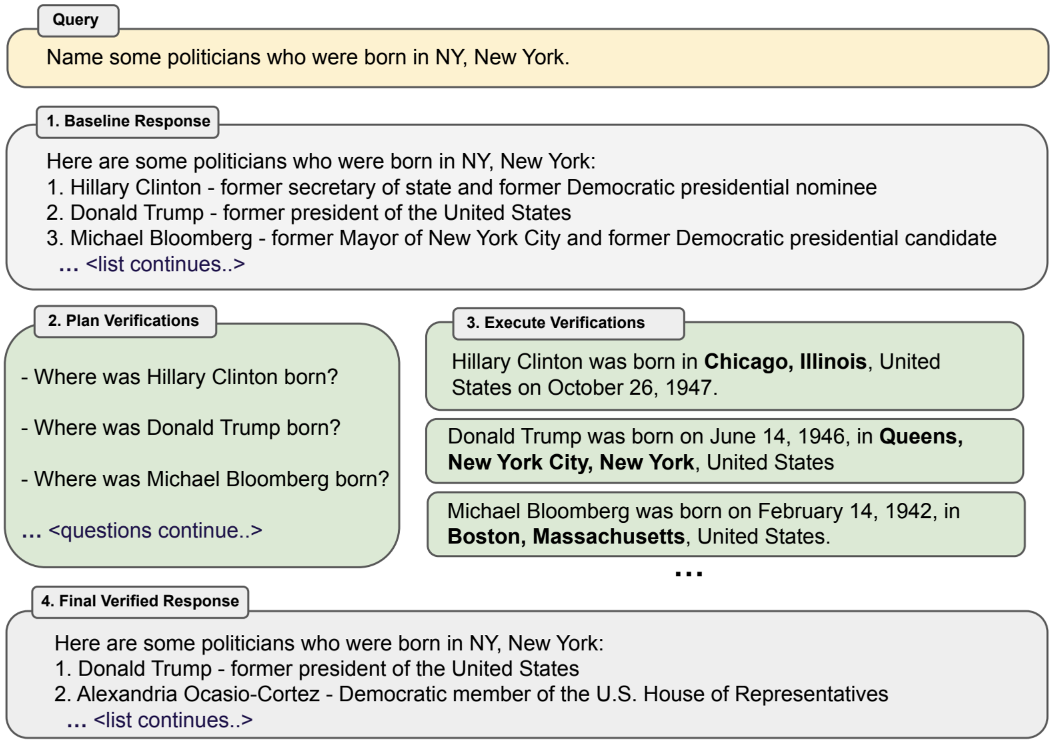
COVE[31], introduced by Meta, presents an iterative self-verification method wherein an LLM drafts an initial response, formulates verification queries to fact-check this draft, independently answers these queries to avoid bias, and finally, synthesizes a verified response. This process, depicted in Figure 9, embodies a comprehensive approach to enhancing the reliability and factual accuracy of LLM outputs by integrating a self-contained verification loop.

4.4.6 Rails

Rails represent a sophisticated mechanism for directing the output of a Large Language Model (LLM) through programmable constraints. Defined using Colang, a straightforward modeling language, and Canonical Forms that serve to standardize expressions in natural language, rails offer a dynamic framework for modulating LLM behaviors[5].

Among the various applications of rails, the Fact Checking rail stands out for its relevance to our discussion on minimizing hallucinations. This specific rail is engineered to enhance the factual accuracy of LLM outputs, serving as a guardrail against the generation of unfounded or erroneous content.

[5] https://github.com/NVIDIA/NeMo-Guardrails/blob/main/docs/README.md



Name some politicians who were born in NY, New York.

Here are some politicians who were born in NY, New York:
1. Hillary Clinton - former secretary of state and former Democratic presidential nominee
2. Donald Trump - former president of the United States
3. Michael Bloomberg - former Mayor of New York City and former Democratic presidential candidate
... <list continues..>

- Where was Hillary Clinton born?
- Where was Donald Trump born?
- Where was Michael Bloomberg born?

... <questions continue..>

Hillary Clinton was born in **Chicago, Illinois**, United States on October 26, 1947.

Donald Trump was born on June 14, 1946, in **Queens, New York City, New York**, United States

Michael Bloomberg was born on February 14, 1942, in **Boston, Massachusetts**, United States.

...

Here are some politicians who were born in NY, New York:
1. Donald Trump - former president of the United States
2. Alexandria Ocasio-Cortez - Democratic member of the U.S. House of Representatives
... <list continues..>

Figure 1: Chain-of-Verification (CoVe) method. Given a user query, a large language model generates a baseline response that may contain inaccuracies, e.g. factual hallucinations. We show a query here which failed for ChatGPT (see section 9 for more details). To improve this, CoVe first generates a plan of a set of verification questions to ask, and then executes that plan by answering them and hence checking for agreement. We find that individual verification questions are typically answered with higher accuracy than the original accuracy of the facts in the original longform generation. Finally, the revised response takes into account the verifications. The factored version of CoVe answers verification questions such that they cannot condition on the original response, avoiding repetition and improving performance.

Figure 9: The Chain-of-Verification (COVE) process

4.4.7 Guidance (Constrained Prompting)

The concept of ”Constrained Prompting,” introduced by Andrej Karpathy, encapsulates methodologies that integrate generation, prompting, and logical control within the operational flow of LLMs. This approach aims to provide a structured framework that guides the LLM’s output more precisely.

Guidance represents a pioneering example of this concept, functioning not merely as a prompting technique but as a full-fledged ”prompting language”[6]. It leverages templates based on Handlebars syntax to facilitate a seamless integration of prompting and generation, along with the management of logical control flow and variables. This structured approach ensures that, at any point in the process, the LLM can be utilized for text generation or to make logical decisions, thereby offering a comprehensive toolkit for implementing a wide array of strategies discussed in this post.

Through the application of Guidance, developers can harness the full potential of constrained prompting, enabling the implementation of sophisticated logic and control mechanisms within LLM workflows. This advanced technique

[6] https://github.com/guidance-ai/guidance

underscores the evolving landscape of LLM interaction, where structured, language-based frameworks empower users to shape and steer model outputs with unprecedented precision.

4.5 Model Selection and Configuration for Hallucination Mitigation

Mitigating hallucinations in Large Language Models (LLMs) extends beyond prompt engineering to encompass strategic model selection and nuanced configuration settings. The inherent characteristics of the chosen model, along with how it's configured, play pivotal roles in its propensity to produce grounded, accurate responses.

The size and complexity of an LLM are fundamental determinants of its grounding capabilities. For example, advancements from GPT-3.5 to GPT-4 have demonstrated a notable improvement in reducing hallucinations, attributable to the larger model's enhanced data processing and learning depth. This suggests a correlation where more extensive and sophisticated models tend to be better equipped at understanding context and maintaining factual consistency. Note though that ”size is not all that matters”. It is advisable to refer to some of the leaderboards described in section 3.2 for an up-to-date, independent, assessment of how different models fare.

Another critical aspect of configuration involves the model's temperature setting, which governs the randomness of its outputs. Opting for a lower temperature can steer the model towards more predictable and conservative outputs, thereby reducing the risk of hallucinatory content. This setting acts as a dial to balance the model's creativity against the need for precision and reliability, allowing for tailored adjustments based on the application's specific demands.

In essence, the deliberate selection of an LLM and thoughtful adjustments to its operating parameters can significantly influence its effectiveness in avoiding hallucinations. This approach underscores the importance of a holistic strategy that combines model selection with optimal configuration to enhance the reliability of LLM outputs.

4.6 Mitigating Hallucination Through Alignment Techniques

Alignment techniques, designed to harmonize Large Language Models (LLMs) with human preferences, also hold potential for mitigating hallucinations when applied judiciously.

Instruction tuning, for instance, has been shown to reduce hallucinations when the model is fine-tuned with a carefully curated instruction dataset. This approach, as discussed in recent studies[32], leverages targeted instructions to guide the model towards generating more accurate and grounded responses.

Tailoring models through domain-specific fine-tuning represents another effective strategy. By focusing the model's training on a specific field or subject area, it becomes more adept at producing relevant and factually consistent outputs, particularly in specialized applications where accuracy is paramount.

Reinforcement Learning from Human Feedback (RLHF) has emerged as a powerful method for refining model outputs based on human preferences[33]. In contexts where domain-specificity is key, RLHF can significantly contribute to reducing instances of hallucination by steering the model towards responses that align more closely with human expectations and factual accuracy.

However, the success of these alignment techniques in curbing hallucinations hinges on the quality of the data and the nature of the feedback provided. Alignment aims primarily at resonating with human preferences, which may not always prioritize the reduction of hallucinations. There have been instances where the application of RLHF, without careful consideration of the feedback's grounding, has inadvertently led to an increase in hallucinatory content. Thus, the implementation of these techniques necessitates a nuanced approach, ensuring that the feedback and data used for alignment are well-grounded and reflective of factual accuracy.

5 Conclusions

The phenomenon of hallucinations in Large Language Models (LLMs) has surged to the forefront of AI research as these models become increasingly integrated into various applications. With the growing reliance on LLMs, the imperative to mitigate hallucinations has never been more pronounced. This paper has outlined that while hallucinations are an inherent aspect of current LLMs, their mitigation necessitates a multifaceted approach, encompassing advanced prompting techniques, careful model selection and configuration, and alignment with human preferences.

Advanced prompting techniques such as Self-consistency and Reflection offer promising avenues for reducing hallucinations by enhancing the self-awareness and reasoning capabilities of LLMs. Similarly, strategic model selection and configuration, particularly in terms of model size and temperature settings, play a crucial role in grounding model outputs. Alignment techniques like instruction tuning and RLHF further underscore the importance of tailoring LLM behavior to closely align with human feedback and preferences, albeit with the caveat that the effectiveness of these methods is contingent upon the quality of the underlying data and feedback.

As the field continues to evolve at a rapid pace, there is a justified optimism that ongoing research and innovation will lead to significant advancements in reducing the incidence and impact of hallucinations in LLMs. The collective efforts across various domains of AI research are poised to enhance the reliability and trustworthiness of LLMs, ensuring their beneficial integration into society.

References

[1] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024.

[2] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean: A survey on hallucination in large language models, 2023.

[3] S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models, 2024.

[4] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023.

[5] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023.

[6] Ellen Riloff and Lee Hollaar. Text databases and information retrieval. *ACM Comput. Surv.*, 28(1):133–135, mar 1996.

[7] Ann Irvine and Chris Callison-Burch. Hallucinating phrase translations for low resource MT. In Roser Morante and Scott Wen-tau Yih, editors, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 160–170, Ann Arbor, Michigan, June 2014. Association for Computational Linguistics.

[8] Jingjing Liu and Stephanie Seneff. Review sentiment scoring via a parse-and-paraphrase paradigm. In Philipp Koehn and Rada Mihalcea, editors, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 161–169, Singapore, August 2009. Association for Computational Linguistics.

[9] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks, 2023.

[10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[12] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. Handling divergent reference texts when evaluating table-to-text generation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy, July 2019. Association for Computational Linguistics.

[13] Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. Towards faithful neural table-to-text generation with content-matching constraints. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online, July 2020. Association for Computational Linguistics.

[14] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[15] Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[16] Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. Evaluating attribution in dialogue systems: The BEGIN benchmark. *Transactions of the Association for Computational Linguistics*, 10:1066–1083, 2022.

[17] Sashank Santhanam, Behnam Hedayatnia, Spandana Gella, Aishwarya Padmakumar, Seokhwan Kim, Yang Liu, and Dilek Z. Hakkani-Tür. Rome was built in 1776: A case study on factual correctness in knowledge-grounded response generation. *ArXiv*, abs/2110.05456, 2021.

[18] Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. A token-level reference-free hallucination detection benchmark for free-form text generation. *arXiv preprint arXiv:2104.08704*, 2021.

[19] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models, 2023.

[20] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore, December 2023. Association for Computational Linguistics.

[21] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation, 2023.

[22] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022.

[23] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2024.

[24] Xavier Amatriain. Prompt design and engineering: Introduction and advanced methods, 2024.

[25] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[26] Meta. Retrieval augmented generation: Streamlining the creation of intelligent natural language processing models, 220.

[27] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023.

[28] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

[29] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

[30] Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. Dera: Enhancing large language model completions with dialog-enabled resolving agents, 2023.

[31] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models, 2023.

[32] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.