Overcoming False Premises Using Abductive Reasoning in Question Answering with LLMs

by

Varun Singh

to obtain the degree of Master of Science in Computer Science
at the Delft University of Technology,
to be defended publicly on Thursday July 11, 2024 at 15:00.

Student number:	5441935	
Project duration:	November 1, 2023 – July 11, 2024	
Thesis committee:	Prof. Ujwal Gadiraju,	TU Delft, Thesis advisor
	Gaole He,	TU Delft, Daily co-supervisor
	Prof. Thomas Höllt,	TU Delft, Computer Graphics & Visualisation

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

It feels unreal to write this preface which I had been putting off till every other part of the thesis was complete. I can hardly believe I have reached the end. I sincerely hope the work presented in this thesis is useful in developing reliable, trustworthy, and honest AI systems.

I would first like to thank and acknowledge all those who have contributed towards the completion of this thesis. The work presented in this document would not have been possible without my supervisors Prof. Ujwal Gadiraju and Gaole He. I would like to thank Prof. Gadiraju for providing me with the wonderful opportunity to work on a topic that I found fascinating and for giving me the freedom to explore it in my own way. I would especially like to thank my daily supervisor, Gaole He, who has stuck with me since the beginning and witnessed in every step how the thesis has progressed. His patience when it would come to discuss ideas presented by me that were clearly not up to the mark and still having a meaningful inquiry was very much appreciated. He has always taken the time to discuss and answer any questions I might have had, no matter how trivial it might have seemed. I would also like to thank Prof. Thomas Höllt for agreeing to be a part of my thesis committee.

My dear friends Aitor, Sepehr, Eva, Nicolas, Gabriela, and Jonah, your kindness, understanding, and joyfulness made my transition to the Netherlands not only bearable but truly enjoyable. Rahul and Deepali have had my back throughout this entire journey, constantly encouraging me to become stronger and better. They have helped me maintain my motivation, especially during tough times. I couldn't have wished for better friends. Lastly, I would like to express my heartful thanks to my parents and family for their unconditional support and blessings. I would not be in the position I am today without their unconditional support and love.

Varun Singh
Den Haag, July 2024

Abstract

Language models (LLMs) have demonstrated impressive performance on knowledge-intensive tasks like question answering when supported by external knowledge. However, their success relies not only on their reasoning capabilities and the accuracy of the external knowledge but also on the truthfulness of the prompts provided. False premises in prompts can lead to ”hallucinations,” where the generated content appears plausible but is factually incorrect. This issue is common in online questions, particularly when users search for information on unfamiliar topics, leading to confirmation bias in information retrieval.

Existing methods for detecting hallucinations may not effectively handle false premises, as they can be misled by coherent responses that align with the false premises. Fact-checking methods may also be unsuitable, as LLMs can exhibit sycophantic behavior in attempting to satisfy user requirements.

To address this challenge, we propose a **F**alse **P**remise **D**etection with **A**bductive **R**easoning **(FPDAR)** method for question answering with LLMs. Abductive reasoning enables backward thinking, minimizing less plausible assumptions and working towards the correct answer through a bottom-up approach. FPDAR is designed as a plug-and-play module that can be integrated after the question-answering process.

FPDAR employs a two-stage abductive reasoning process. First, it infers the most plausible question intent based on factual context and generated response without considering the potentially problematic question as input. This allows for the identification of false premises by comparing the inferred intent with the original question. Second, abductive reasoning helps generate a more plausible explanation aligned with factual context, increasing the likelihood of correctness and ruling out less plausible, potentially hallucinated responses.

To the best of our knowledge, this is the first study introducing abductive reasoning for identifying and diagnosing false premises. We conduct extensive experiments on two question-answering benchmarks containing false premises to validate the effectiveness of FPDAR. The results show that FPDAR can achieve high accuracy in terms of response correctness, although it may struggle to effectively detect false premises. Nevertheless, it achieves substantial accuracy improvements over state-of-the-art methods.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
CoT	Chain of Thought
FPDAR	False Premise Detection with Abductive Reasoning
LLM	Large Language Model
NLP	Natural Language Processing
QA	Question Answering
RAG	Retrieval Augmented Generation

Symbols

Symbol	Definition
P_1	Base QA Prompt
P_2	Infer Question Q' Prompt
P_3	Generate Alternate Explanation Z Prompt
Q	Original Question
Q'	Inferred Question
X	Additional Context/External Information
X'	Additional Context from Q'
Y	Initial Answer/Conclusion
Z	Alternate Explanation
s	Intent Similarity Function
τ	Semantic Similarity Threshold

1

Introduction

The brain is an abduction machine, continuously trying to prove abductively that the observables in its environment constitute a coherent situation.

– Jerry Hobbs, ACL 2013 Lifetime Achievement Award[1]

The past few years have witnessed tremendous advancements in Computing and Artificial Intelligence (AI). One of the most notable developments is the advent of Large Language Models (LLMs) [143, 105, 55] such as OpenAI's GPT-3 [69] and GPT-4 [1]. LLMs have demonstrated remarkable performance on various downstream tasks without requiring explicit training on specific use-case datasets [55, 129]. This capability has enabled the field of Natural Language Processing (NLP) to shift from a task-centric approach, where specific models needed to be trained for individual tasks like machine translation and summarization, to a more generalized use-case setting. In this new paradigm, LLMs can be adapted to perform various tasks as required.

Due to their advent, popularity, and potential, LLMs are incorporated into various domains, including medicine [117], law [18], and finance [61]. However, as with any AI system, there are reliability concerns [142, 132, 107], especially in the case of LLMs, since they can generate content that appears plausible but is factually false or illogical [144, 66]. The quality of an LLM-generated output also significantly depends on the instructions provided in the prompt [129, 21]. In most practical applications, it is generally assumed that the input to the LLM is accurate and logical. However, this is not always the case, as prompts may contain incorrect or illogical information, degrading the model's output [54]. As the old saying goes, "Garbage in, garbage out." Incorrect information or assumptions in the prompts can be unintentional and may stem from a lack of relevant knowledge when creating the prompt [45]. In such instances, the LLM must be robust enough to identify these faulty assumptions and provide a reasonably accurate response based on the correct information available in the prompt.

This thesis aims to improve the reliability of LLM-generated textual content in scenarios where the LLM might be requested to provide a response based on flawed underlying assumptions. Identifying and overcoming these flawed assumptions can be challenging for models [124], especially since they are typically trained and fine-tuned on ideal and somewhat perfect information and data formats [143]. Our approach to improving reliability operates at the inference level, utilizing the LLM's inherent reasoning capabilities more effectively through a specific reasoning strategy. We propose that a backward reasoning methodology based on logical reasoning will be superior in addressing scenarios involving faulty assumptions. Our solution is implemented in a Knowledge Intensive Question Answering setting where external knowledge can be obtained to support question answering. Since our questions may contain false premises, we use external knowledge to assist the LLM in answering the question and demonstrate how our backward reasoning strategy can further improve the model-generated output by identifying false premises. This thesis will detail our rationale and implementation for this strategy,

[1]The full transcript of his award speech is available at `https://www.mitpressjournals.org/doi/full/10.1162/COLI_a_00171`

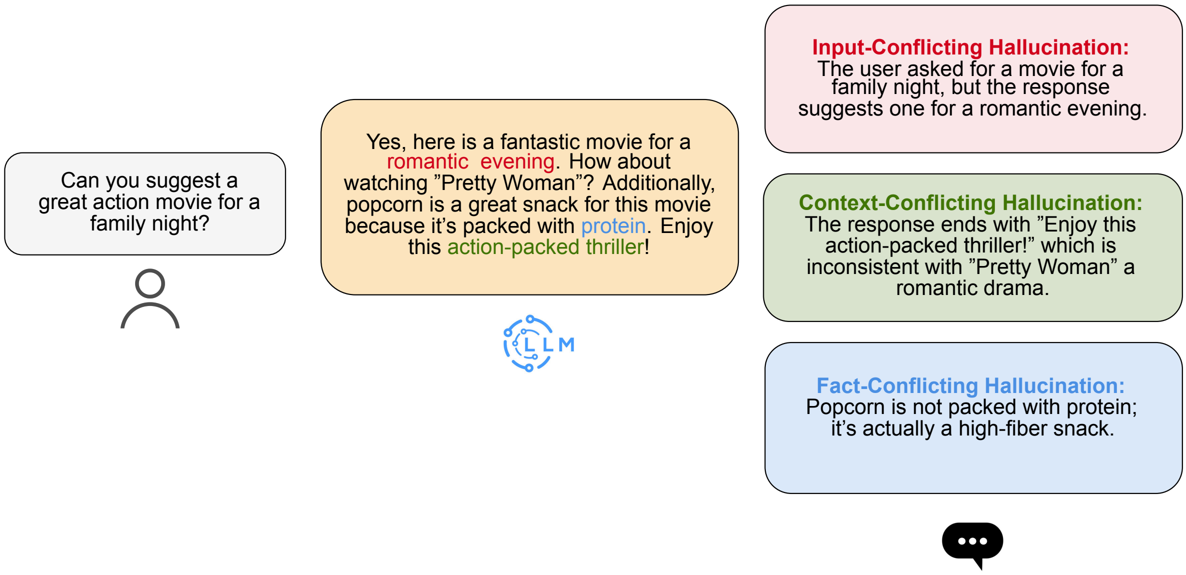


Figure 1.1: An illustration of Hallucination and its categories according to [144]

along with benchmarks comparing various methods and models. But first, let us understand the specific problem this thesis aims to tackle.

1.1. Challenges

This section will focus on challenges that undermine the reliability of LLMs, namely hallucinations. The remainder of this thesis will focus on a specific type of hallucination: factual hallucinations and certain challenges often associated with factuality, namely false premise.

1.1.1. Hallucination

In the realm of LLMs, hallucination is an instance where the model creates content that isn't grounded in factual or precise information. This phenomenon occurs when the model generates text that includes invented, deceptive, or completely made-up details, facts, or assertions instead of delivering dependable and truthful data [46]. This issue stems from the model's capacity to produce seemingly plausible text based on patterns it has learned from its training data [56, 105], even if the generated content is not factually correct [31, 104]. Hallucination can be inadvertent and may be caused by various factors, such as prejudices in the training data, the model's inability to access real-time or updated information, or inherent limitations in the model's understanding and generation of contextually accurate responses.

Hallucinations can be categorized into various groups, and some surveys [46, 42] have done so to suit their specific focus. We will follow the categorization by [144], as it allows us to delve into the specific type of hallucination of interest.

An example highlighting the different types is presented in figure 1.1 and is discussed below:

- **Input-conflicting hallucination:** Occurs when language models produce content that diverges from the user-provided input.
- **Context-conflicting hallucination:** Happens when language models generate content that contradicts information previously generated by themselves.
- **Fact-conflicting hallucination:** Arises when language models produce content that is inaccurate or not consistent with established knowledge.

Input-conflicting hallucinations This type of hallucination occurs when the content generated by language models strays from the user's input. Generally, user input for language models consists of the system prompt (describing the task and its instruction) and the user prompt (the task input). Sometimes, there can be a contradiction between the task instruction and the input. The task instruction and input are combined in the example seen in figure 1.1. The user is querying the model to recommend a movie for a specific genre. Over here, asking the model to act as a recommender is the system prompt, and the subsequent specific recommendation requested is the user prompt. However, there is a conflict between the prompts and the LLM's response. The prompts request a movie from the action genre, but the response suggests a romantic movie.

Context-conflicting hallucination LLMs may exhibit self-contradictions when generating lengthy or multi-turn responses. This type of hallucination occurs when they lose track of the context or fail to maintain consistency throughout the conversation. This can be due to limitations in maintaining long-term memory or in identifying relevant context [108]. As seen in figure 1.1, the LLM starts by suggesting how "Pretty Woman" is a romantic genre movie but then contradicts the context by mentioning how the suggested movie is an action genre.

Fact-conflicting hallucination When LLMs generate inconsistent content or conflict with established world knowledge, it is termed factual hallucination. From the example in figure 1.1, the claim that popcorn is high in protein is inconsistent with world knowledge and is therefore termed a factual hallucination. Fact-conflicting hallucinations are distinct from the other two types since they can be verified through external knowledge sources and corrected with feedback. This might give the impression that factual hallucinations are easier to solve, but they come with their own set of challenges. Firstly, due to the absence of an authoritative knowledge source, deciding which source should be considered as a reference can be difficult. Second, factual information is often requested in the form of questions. These questions are assumed to be correct and consistent with established world knowledge, but that is not always or rather never a guarantee in practical systems. Questions based on false assumptions can be challenging for LLMs to overcome because the type of answer requested is often the wrong answer without further elaboration.

1.1.2. False premises

The input provided to LLMs is often presumed to be accurate; however, this isn't always the case. Naturally occurring information-seeking questions often contain false premises or assumptions that can induce hallucinations, as they may prompt language models to respond based on incorrect or misleading information [54]. These false assumptions can also be present separate from the primary message being conveyed through language and are typically assumed to be true by all participants in a conversation. For example, if someone says, "Let's reschedule our meeting," it assumes an initial meeting was scheduled, implying prior knowledge or agreement on the matter, but this assumption could be false. In the context of questions, the accuracy of these false premises is crucial because it determines whether the question can be answered using the requested answer type. Providing a matching answer type might inherently support the false assumption and will inevitably lead to a hallucinated response [45]. An example of this can be seen in table 1.1 with the third question. The question is requesting a numeric answer, but any attempts to do so will support the false premise that Mark Zuckerberg founded Google[2]. Instead, the proper strategy would be to identify such invalid premises and, if possible, assume a valid question and provide the true answer. Questions containing faulty assumptions are not necessarily formed with malicious intent; they can arise from a genuine desire to understand a topic better. Typically, the person asking the question has some basis for their inquiry, although this basis might be incorrect. For example, consider the second question in Table 1.1. The question might seem like a genuine factual inquiry to someone unfamiliar with this topic, even though it is based on a faulty assumption. This issue is further exacerbated when searching online, leading to confirmation bias in information retrieval. False premise questions can also inquire about events that may occur in the future, as seen in the fourth question. This can prove to be difficult for LLMs to tackle since if they attempt to answer it; they might provide an answer that could be outdated, limited by their training knowledge cut-off [132]. Questions like these contain temporal aspects and are termed as dynamically changing. The particular example is a question that is dynamically changing and contains a false premise because there is no person who fits that question, but it is also something that could change in the future. LLMs, in general, still struggle when being asked to answer questions containing false premise and temporal aspects [124]. It is important to note that these false assumptions are not necessarily malicious but rather genuine misconceptions or flawed beliefs that the user might hold. When these assumptions are

[2]Perhaps an answer stating zero years and further elaborating how Mark Zuckerberg did not found Google would technically be valid but the point here is that a standard numeric answer without any explanation would not work.

Question	Answer	Comment
What is the capital of France?	Paris	True Premise question that was answered correctly.
When did Madam Curie discover Uranium?	Madam Curie discovered Uranium in 1898	Question containing false premise that was answered incorrectly. The faulty premise is that Madam Curie discovered Uranium. Her work on radiation centered on uranium but she did not discover the element, it was Henri Becquerel.
How old was Mark Zuckerberg when he founded Google?	Mark Zuckerberg did not found Google. He is the founder of Facebook.	Question containing false premise that was answered correctly. The response is able to identify the false premise and provide the correct answer.
Who is the first female president of the United States of America?	This question contains a false premise because the United States has not yet had a female president.	Question containing false premise and also a temporal aspect that was answered appropriately. The response is able to identify the false premise and also acknowledge that this is an event that has not yet occurred.

Table 1.1: This table illustrates examples of true and false premises, along with dynamically changing questions, showcasing the types of questions Large Language Models (LLMs) can be asked and their potential responses, which may be correct or incorrect.

malicious in nature, they could be attributed to spreading misinformation such as fake news [126]. This is a separate topic with some overlapping but mostly different challenges requiring distinct approaches. The focus of this thesis is solely on false premises.

1.2. Motivation and Contributions

While having showcased remarkable performance in various downstream tasks, LLMs are plagued by various issues, such as hallucinations, which hampers the reliability of such models in practical high-stakes application fields such as medicine, finance, and law. Still, there is no denying that these models possess incredible comprehension and reasoning abilities, which have been proven through impressive performance in various benchmarks [136, 134, 91, 113, 8] even though there is a debate regarding the truth behind these claims and whether LLMs are truly capable of understanding. This is discussed in detail in appendix A.

Drawing from the impressive reasoning capabilities exhibited by LLMs, our objective is to leverage these strengths to enhance the reliability of LLMs in challenging contexts, especially those entailing false premises, thereby mitigating potential hallucinations. While a variety of datasets and benchmarks exist for false premises, there have not been many attempts to propose an approach to overcome such scenarios. Most of the research on false premises has evaluated models on different benchmarks [124, 54, 14, 141]. A few methods [33, 123] that have been proposed are from the LLM reasoning and prompting literature which often involve querying a knowledge base for external knowledge to verify the veracity of factual claims. However, these methods may struggle with false premises since they rely on checking factual claims based on the premise provided in the question, and if the question itself is faulty, then the verification would likely result in faulty claims being presented as accurate. In addition, most of these approaches apply a forward reasoning approach, which is also the most intuitive since it essentially involves reaching a solution to a given problem. We wanted to explore whether employing a backward reasoning approach would perform better in identifying false premises and hallucinations. A backward reasoning approach would involve providing a solution and investigating whether the initial premise (problem) can be reached. This can be considered a form of verification and is often used as a strategy for solving and verifying mathematical proofs. Based on research in proof verification through AI systems [52] and evaluations involving reasoning with LLMs [134], there is evidence to suggest that backward reasoning as an approach might provide comparable or superior performance in terms of the responses being generated by activating the models to reason more effectively.

The novelty in our work is found in the implementation of backward reasoning. We propose a plug-and-play backward reasoning approach grounded in a specific logical reasoning theory, i.e., Abductive Reasoning. Abductive reasoning involves generating hypotheses to explain a present situation in the best possible manner. The abduction process works naturally with our backward reasoning approach since they both require reasoning in a reverse direction. We incorporate this approach into a Knowledge Intensive Question Answering setting [89] where the LLM's inherent reasoning capabilities are used to detect false premises and reduce hallucinations. Research into using abduction for task accomplishment has shown promising results, such as generating more reasonable and sound explanations [115], predicting future events based on past information [111], and enhancing the generation of personalized healthcare recommendations for patients [27]. However, as far as we have investigated, this is the first study on utilizing abduction to detect false premises and hallucinations in LLMs.

1.3. Research Question and Approach

This thesis aims to investigate whether a backward reasoning approach based on abductive reasoning can be utilized to detect false premises and hallucinations in LLM-generated content in a QA setting. Based on this, the following research question is formulated:

> *Can abductive reasoning be used to detect and mitigate false premises in Large Language Models (LLMs) within a Question Answering (QA) framework?*

Our approach to answering this research question will be split into three distinct steps:

1. Develop a solution that integrates a backward reasoning methodology through abductive reasoning for question answering (QA).
2. Benchmark the proposed solution against other comparable methods on false premise QA datasets to demonstrate the viability of a backward reasoning methodology within a QA framework.
3. Discuss the benefits, limitations, and insights derived from the benchmarks and the feasibility of extending a backward reasoning strategy beyond false premises.

The remainder of this thesis is organized as follows:

In Chapter 2, we provide relevant knowledge regarding large language models, hallucinations and false premises, building upon the explanation in Chapter 1. This chapter focuses on approaches to addressing hallucinations and false premises, discusses logical reasoning, and explores how and why large language models (LLMs) are used for reasoning in practical applications. It also examines different methodologies for achieving reasoning through LLMs. Chapter 3 details the methodology of the abductive reasoning approach, its incorporation into a knowledge-intensive question-answering framework, and our rationale for its effectiveness. This is followed by organizing the methodology into a dedicated workflow called **False Premise Detection with Abductive Reasoning (FPDAR)**. Chapter 4 outlines our experimental setup and demonstrates how **FPDAR** compares against other methods across different benchmarks. Finally, in Chapter 5, we discuss key findings, limitations, and potential promising research directions for future work.

2 Background Knowledge and Related Work

In the previous chapter, we explored the issues of hallucinations and false premises, focusing primarily on why they pose significant problems. This chapter aims to build on that discussion by first walking through the development of large language models, examining existing methods for detecting hallucinations, and identifying false premises. Given that our research question seeks to evaluate the effectiveness of a backward reasoning approach, we will also introduce logical reasoning and its three main types, focusing primarily on abductive reasoning.

Following this, we will delve into natural language reasoning, discussing how large language models (LLMs) are typically employed for such tasks and why they are essential. This discussion will provide a comprehensive background for the proposed abductive reasoning approach explained in Chapter 3.

2.1. Large Language Models

Language is a significant human capability for expression and communication that begins to develop in early childhood and continues to evolve throughout life [35]. In contrast, machines cannot inherently understand and communicate in human language. Achieving this has been a persistent research challenge, aiming to enable machines to read, write, and communicate in a human-like manner [119]. Through certain Artificial Intelligence (AI) techniques, there have been attempts to overcome this research challenge. From a technical standpoint, language modeling (LM) stands out as a primary method for enhancing the linguistic intelligence of machines. Essentially, LM seeks to construct models that predict the probability of word sequences (tokens) being generated, thereby enabling predictions of the probabilities associated with future tokens.

Based on this setup, research into different strategies for developing models has existed, the earliest relying on statistical methods. Statistical language models (SLMs) [28, 98] primary focus is on word prediction by harnessing the Markov assumption, which involves predicting the next word based on the most recent context. SLMs with a fixed context length of ’n’ are known as ’n-gram language models’, exemplified by bigram and trigram models. These models have found extensive application in improving the performance of tasks related to information retrieval (IR) [64] and natural language processing (NLP) [116]. However, a significant challenge encountered with high-order language models is the curse of dimensionality, which makes it difficult to estimate transition probabilities accurately as the number of these probabilities grows exponentially.

Moving from SLMs, which were task-specific helper models, Pre-trained language models (PLMs) have revolutionized the field of NLP. ELMo was an early pioneer [102], using a bidirectional LSTM network to learn context-aware word representations that could be fine-tuned for specific tasks. Later, BERT [20] introduced the Transformer architecture [122] with self-attention mechanisms and pre-trained on large-scale unlabeled datasets, achieving state-of-the-art results. This ”pre-training and fine-tuning” approach has since become a standard paradigm in NLP, with many subsequent studies proposing new architectures (e.g., GPT-2 and BART) [92, 58] and pre-training strategies [101, 125]. These PLMs have raised the bar for NLP tasks but often require fine-tuning to adapt to different downstream tasks.

As researchers scale up pre-trained language models (PLMs) by increasing the model or training data size, they often observe improved performance on downstream tasks, following the scaling laws [50]. Several studies have pushed the limits by training ever-larger PLMs, such as GPT-3 with 175 billion parameters and PaLM with 540 billion parameters. These massive language models exhibit different behaviors and surprising capabilities, which have been termed ”emergent abilities” [21] compared to smaller models like BERT (330 million parameters) and GPT-2 (1.5 billion parameters). For instance, GPT-3 can perform few-shot learning [69] tasks, which GPT-2 struggles with. As a result, the research community has coined the term ”large language models” (LLMs) to refer to these large models, drawing increasing research interest. A notable application of LLMs is ChatGPT [1], which adapts the GPT series for dialogue and demonstrates remarkable conversational abilities with humans.

The rapid progress of LLMs is revolutionizing various AI research areas. In Natural Language Processing (NLP), LLMs can function as general-purpose language task solvers to some extent, and the research paradigm is shifting towards using LLMs. In Information Retrieval (IR), traditional search engines are being challenged by the new information-seeking method through AI chatbots like ChatGPT, and Bing Chat [2] presents an initial attempt to enhance search results based on LLMs. In the field of Computer Vision (CV), researchers are developing ChatGPT-like vision-language models to support multimodal dialogues better [43, 11], and GPT-4 [1] has introduced multimodal input by integrating visual information. This new wave of technology could potentially lead to a prosperous ecosystem of real-world applications based on LLMs. Most recently, in May of 2024, OpenAI released its most capable model, GPT-4o[3]. GPT-4o's performance on various benchmarks is an improvement, but the most impressive capability of the model is its ability to accept input in various combinations of text, audio, and image and generate an output in various combinations of text, audio, and image. Before GPT-4o, a pipeline of different models had to be set up first to transcribe audio to text, process it, and output it again as audio. Advancements like this in LLMs have profoundly impacted the Artificial Intelligence (AI) community, spurring a reevaluation of the possibilities for artificial general intelligence (AGI).

While the success of LLMs is quite promising for future research, they are not without their problems. It is largely unclear why the emergent abilities have appeared in LLMs and not in smaller language models [130, 26]. This might also be compounded by the difficulty of training and researching LLMs (especially commercial models) due to the large amount of computing and training resources needed to conduct investigative studies. Adding to this, many LLMs are developed by the industry where a significant amount of crucial information is not revealed, i.e., they are close-sourced models. With AI systems, there is also the risk of biases being present due to perhaps being trained on biased or unbalanced data. In LLMs, the risk of generating toxic, harmful, and false output is also prevalent. LLMs need to be better aligned with human values such as the 3H (Honesty, Harmless, Helpfulness) [149].

As discussed in the introduction, our focus in this thesis is on identifying false premises and mitigating any potential hallucinations that could be caused by such faulty assumptions. The next section will provide a brief discussion of existing research and strategies proposed to achieve this goal.

2.2. Hallucination Detection in Large Language Models

Hallucination detection involves identifying potential hallucinations within the responses generated by LLMs. Various benchmarks have been proposed to evaluate hallucination detection in LLMs. However, since our focus is specifically on factual hallucinations, as highlighted in Section 1.1, this discussion will be limited to a few of the most influential datasets primarily used for detecting factual hallucinations.

[1] `https://openai.com/chatgpt/`
[2] `https://www.bing.com/chat`
[3] `https://openai.com/index/hello-gpt-4o/`

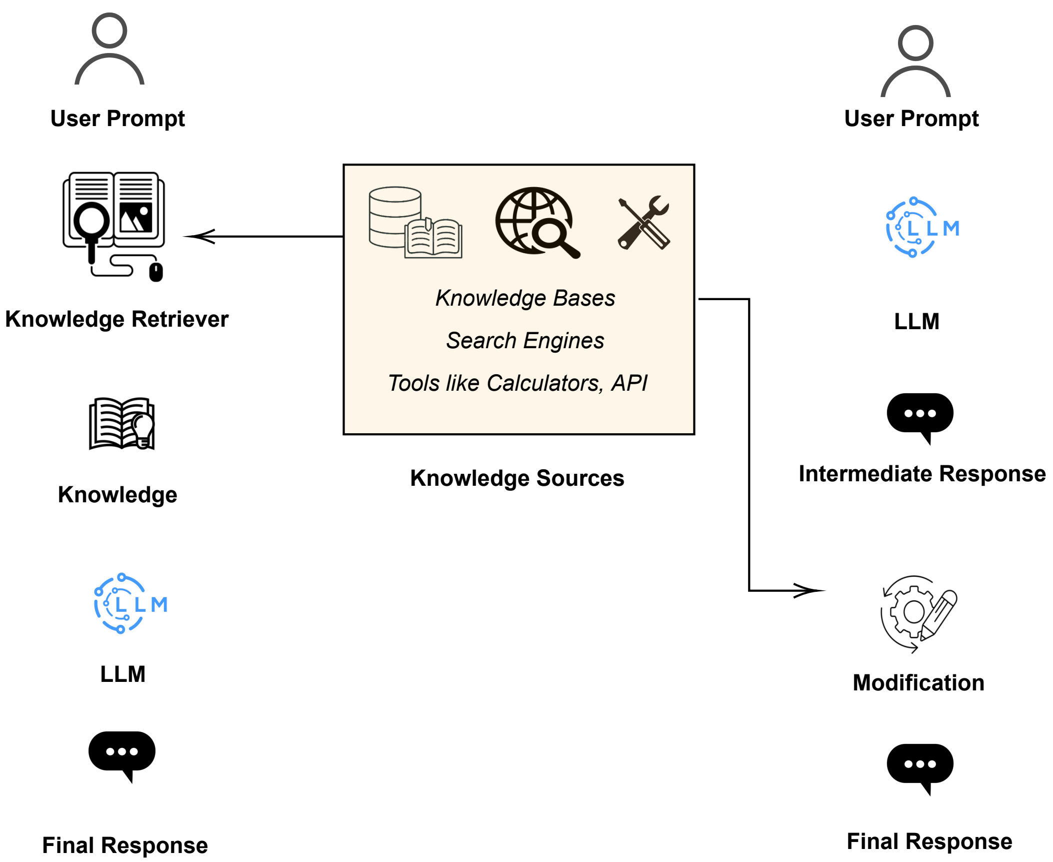


Figure 2.1: A high-level overview of the different processes in Retrieval Augmented Generation (RAG) at inference level. External knowledge is obtained through various knowledge sources and utilized at different stages of the RAG process, depending on the methodology followed. The knowledge can be utilized during the generation phase by simply concatenating it with the user prompt, as seen on the left iteration. Alternatively, as seen on the right, a find-fix-verify strategy could be adopted where the knowledge is used to modify the LLM response.

2.2.1. Evaluation Benchmarks

Detection datasets evaluate the LLM's ability to distinguish hallucinated and truthful statements. Detection does not explicitly require the LLM to indicate where the hallucination has occurred as long as the LLM can choose the most logical response, but these kinds of requirements depend on the tasks defined in the datasets. TruthfulQA [62] is a hallucination detection benchmark that evaluates detection by presenting a Question Answering setting with a multiple-choice format where the model has to identify truthful statements. HaluEval [59], similar to TruthfulQA, requires the model to detect hallucinations in a Question Answering setting but, in addition, also provides instruction tasks such as summarisation and conversation dialogue where the model has to identify the hallucinations. FACTOR [77] evaluates the model's ability to detect hallucination by assigning higher likelihood scores to factual statements than non-factual statements in a multiple-choice setting. The benchmarks discussed above are not the extent to which hallucination detection is evaluated. However, these are representative of the various benchmarks available that tackle a different facet of detection specific to a domain or follow a particular task format. For an exhaustive list of different hallucination benchmarks, the reader is encouraged to view this GitHub repo [75], which lists the various benchmarks and metrics and any specific frameworks that might have been created.

2.2.2. Inference Methods

Hallucination detection methods, much like evaluation benchmarks, are extensive and diverse. This discussion will concentrate on methods applied during inference, as our proposed reasoning approach in Chapter 3 operates at this stage and is based on these strategies. For a comprehensive examination of the various strategies used to detect and mitigate hallucinations, particularly during the training phase, please refer to the surveys [144, 66].

Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation is a promising approach to improving LLM's factuality [73, 96]. It broadly involves two main steps. First is the acquisition of relevant knowledge from external sources. The second step involves leveraging this knowledge in an effective manner to generate responses. Frameworks that incorporate external knowledge to supplement the LLM's knowledge are called knowledge augmented models.

LLMs possess a vast amount of world knowledge through extensive training and fine-tuning in their parameters, which is referred to as parametric knowledge [97]. However, there is no surety that this knowledge is correct, and moreover, it is likely to be outdated, which can lead to hallucinations [132]. To combat this, knowledge acquisition of up-to-date information through credible sources has been proposed by researchers. Knowledge acquisition is typically made by relying on knowledge bases such as MassiveText [93], specific websites like Wikipedia [87], or the Internet itself [57]. Search engines like Google can be incorporated by utilizing a search engine API for knowledge acquisition for various factual downstream applications involving up-to-date and relevant information.

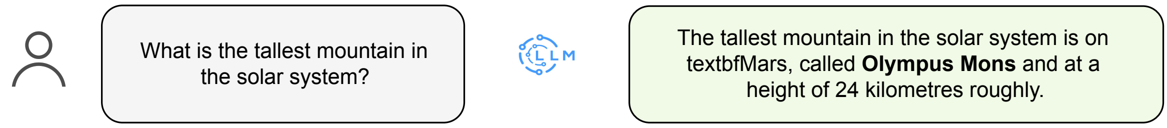
Once knowledge has been acquired, it must be effectively utilized to generate a non-hallucinated response. This knowledge can either be injected during the response generation time by simply concatenating it with the user's instruction prompt to the LLM [109] as seen in figure 2.1 (left). The incorporation of such knowledge or additional information is known as in-context learning [110], and LLMs have been shown to possess strong capabilities for in-context learning by identifying the relevant information from external knowledge required to provide a non-hallucinated response [21]. Another approach to utilizing external knowledge is through a find-fix-verify strategy. First, generate the model's response and then use the external knowledge to compare and identify potential hallucinations that need to be fixed, as seen in figure 2.1 (right). Various methods [10, 30] employing this strategy have been proposed, and they usually involve a few loops to identify, correct, and review the final response.

Uncertainity

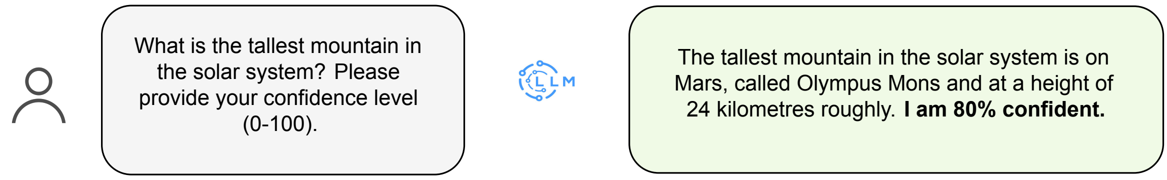
Detecting and managing hallucinations during inference can benefit from considering uncertainty levels, as suggested in [68]. This typically pertains to the confidence level in the model's outputs [47]. Recognizing uncertainty can help users determine when to trust language models (LLMs). If the uncertainty of LLM responses can be accurately characterized, users can identify and address highly uncertain claims, which are more likely to be erroneous. The three main approaches can be seen in figure 2.2 and are as follows:

- **Logits:** The first method, called the logit-based approach, relies on accessing the model's logits and usually measures uncertainty by calculating token-level probability or entropy [32].
- **Verbalise Confidence:** The second method, known as the verbalize-based approach, entails directly asking language models to articulate their uncertainty, often through prompts like, "Please answer and provide your confidence score (from 0 to 100)." This method is effective because of the language models' remarkable comprehension skills and ability to follow instructions [133].
- **Consistency:** Consistency-based approaches [127] rely on the premise that when language models are unsure or making up information, they tend to provide inconsistent responses to the same question. This involves generating multiple candidate responses using a sampling strategy such as temperature sampling, i.e., varying the temperature and generating responses. The core idea is that if the model truly possesses knowledge of a given concept, the candidate responses will likely be similar and consistent. This similar and consistent response is chosen as the final response.

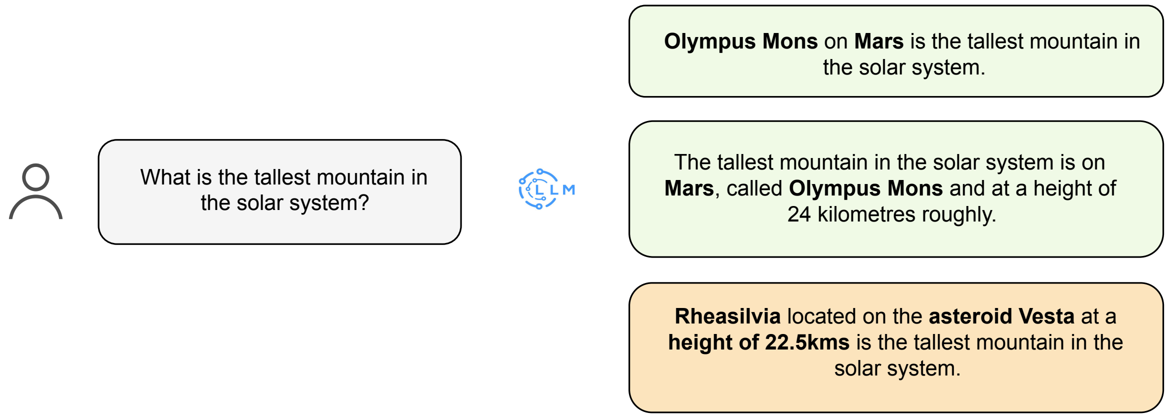
Several recent research works have utilized uncertainty estimation to identify and reduce hallucinations in language models. SelfCheckGPT [68] is the pioneering framework for detecting LLM hallucinations by measuring uncertainty in a zero-resource and black-box scenario. They utilize a consistency-based method for estimating uncertainty. In [2], the authors extend the use of the verbalize-based approach to assessing the hallucination rate of LLMs in generating references. In contrast, [121] employs the logit-based technique to identify false concepts in LLM responses exhibiting high uncertainty. They subsequently address such erroneous content using auxiliary retrieval-augmented LLMs. Authors of



(a) Logit based method



(b) Verbalise Confidence based method



(c) Self-Consistency based method

Figure 2.2: Uncertainity-based methods **(a)** Logit-based methods rely on the probability of the tokens to decide whether a response or part of it is hallucinated or not. If the probability values for certain tokens are lower than a predefined threshold, then those can be flagged as hallucinations. In this example, the bold text has lower probabilities than the other words, indicating that the LLM is uncertain about this part of the response. In this particular case, the overall response is correct. **(b)** Verbalise-based methods rely on the LLM's ability to follow instructions and express their confidence in the response when explicitly prompted. **(c)** Self-consistency requires the model to output several candidate responses and choose the most consistent response, essentially employing a form of majority voting. In this particular example, the first two responses are consistent with each other in terms of the information they convey, and the last response provides a different answer, therefore through majority voting either of the first two responses can be selected as the final response.

[78] employ an extra language model to determine if two LLM responses are logically contradictory within the same context, indicating that at least one of them may be hallucinated. They then use another LLM to correct such self-contradictory hallucinations present in both responses.

Prompt Engineering

Prompt engineering involves crafting a set of instructions that a generative AI model can comprehend and act upon. A prompt consists of natural language that outlines the specific task for the AI to accomplish [112, 148]. Prompt engineering relies on in-context learning, which is characterized by a model's capability to gain knowledge from prompts temporarily. This capacity for in-context learning is an emergent ability from LLMs and is influenced by the scale of the model, with larger models exhibiting increased effectiveness at a different rate compared to smaller ones [21].

Prompt engineering by itself is not a dedicated hallucination detection technique. Rather, it involves constructing appropriate instructions to guide the LLM's response, ensuring accuracy, coherence, and reliability. By generating a proper response, the chances of encountering hallucinations are reduced. Since prompt engineering is the process of constructing task instructions, to an extent, the aforementioned RAG approach and, in general, all methods that operate at inference can be categorized under

Q: Sarah has 3 boxes of chocolates. Each box contains 12 chocolates. She receives 2 more boxes as a gift. How many chocolates does she have now?
Answer:

LLM Response: Sarah has 42 chocolates.

(a) Zero-Shot

Q: In a jar, there are 24 candies. Half of them are peppermints, and half of the peppermints are red. How many red peppermints are there?
Answer: There are 6 red peppermints.

Q: Sarah has 3 boxes of chocolates. Each box contains 12 chocolates. She receives 2 more boxes as a gift. How many chocolates does she have now?
Answer:

LLM Response: The answer is 42.

(b) Few-Shot

Q: In a jar, there are 24 candies. Half of them are peppermints, and half of the peppermints are red. How many red peppermints are there?
Answer: There are 24 candies in total. Half of them, 24 / 2 = 12, are peppermints. Half of the peppermints, 12 / 2 = 6, are red peppermints. So, there are 6 red peppermints in the jar.

Q: Sarah has 3 boxes of chocolates. Each box contains 12 chocolates. She receives 2 more boxes as a gift. How many chocolates does she have now?
Answer:

LLM Response: Sarah initially has 3 boxes of chocolates, each with 12 chocolates. That's a total of 3 * 12 = 36 chocolates. She then receives 2 more boxes as a gift, still with 12 chocolates each. That's an additional 2 * 12 = 24 chocolates. So, she now has a total of 36 + 24 = 60 chocolates.

(c) Few-Shot CoT

Q: Sarah has 3 boxes of chocolates. Each box contains 12 chocolates. She receives 2 more boxes as a gift. How many chocolates does she have now?
Answer: Let's think step by step

LLM Response: Here's how we can solve this step by step:
a) Start with the initial amount: Sarah begins with 3 boxes * 12 chocolates/box = 36 chocolates.
b) Calculate the chocolates from the gift: She receives 2 boxes * 12 chocolates/box = 24 chocolates.
c) Add the chocolates together: Sarah now has a total of 36 chocolates + 24 chocolates = 60 chocolates.

(d) Zero-Shot CoT

Figure 2.3: Prompt Engineering-based methods **(a)** Zero-shot prompting involves instructing the LLM on the task it needs to accomplish without providing any example demonstrations. In this case, the LLM's response is incorrect. **(b)** Few-shot prompting relies on the LLM's ability for in-context learning and involves providing a few demonstrations to guide the model toward the type of response desired. **(c)** Chain of Thought prompting relies on guiding the model to express its reasoning before providing the final response. The examples described how the final answer was reached, which should guide the model in responding similarly. The model can reach the correct answer in this example by explicitly generating its reasoning. **(d)** Chain of Thought can also be achieved in a zero-shot setting by simply appending the phrase "Let's think step by step." This has the advantage of not requiring any examples in the prompt.

prompt engineering. There are explicit techniques that can be incorporated in most workflows like chain-of-thought (CoT) prompt [129] (figure 2.3c) to prompt LLMs to generate reasoning steps before offering final answers have demonstrated the LLM's abilities to generate reasoning steps and avoid hallucinations. Tree-of-thought prompting [65] extends the concept of chain-of-thought by instructing the model to generate one or more "potential next steps" and then executing the model on each of these options using breadth-first, beam, or another tree search method. Zero-shot prompting [92] (figure 2.3a) represents a significant change in how we utilize large language models. This method eliminates the need for vast training data by relying on well-designed prompts to direct the model toward new tasks. The model is provided with a task description in the prompt but does not have access to labeled data for training on precise input-output relationships. Instead, it uses its existing knowledge to predict the new task based on the prompt. Zero-shot prompting can be combined with CoT to generate reasoning steps without relying on in-context learning [55] (figure 2.3d). Few-shot prompting [69] (figure 2.3b) allows for in-context learning by including demonstrations in the prompt. These demonstrations help steer the model towards better performance when generating responses in similar scenarios. The strategies described here work with varying degrees of success, as seen in Figure 2.3. Users are typically encouraged to experiment with different approaches to determine which strategy best suits their specific needs. Based on these approaches, various advancements and new techniques have emerged; for a comprehensive taxonomy of prompting techniques, the reader is encouraged to refer to survey [99].

2.3. False Premises in QA

Datasets and Benchmarks

Questions seeking information often contain assumptions that may be false or impossible to verify. Dealing with such questions requires a different approach than usual, as the answers must address these assumptions directly. Recently, there has been a surge of interest in this topic, focusing primarily on false premises and the temporal aspects of questions in general since there can be an overlapping element. A majority of research has focused on creating datasets to evaluate LLM's reasoning capabilities on dynamically changing (temporal) questions [14, 51, 54, 63]. [14] created *TimeQA* by extracting evolving facts from *Wikidata* along with aligned Wikipedia passages to synthesize 20K timestamped question-answer pairs. [141] constructed *SituatedQA* by annotating 9K realistic questions from existing open-domain QA datasets with temporal context (i.e., timestamps). *StreamingQA* [63] consists of both LLM-generated and human-written questions (146K total questions) answerable from a corpus of timestamped news articles. Also related is the dynamic *RealTimeQA* benchmark [51], which evaluates models weekly on a set of around 30 multiple-choice questions about new events extracted from news websites. The aforementioned works exclusively focus on the questions' dynamic nature (temporal aspect) and the LLM's ability to reason about them. These benchmarks do not necessarily contain false premises. The works done in [138, 54, 124] focus primarily on false premises. [138] consists of 8400 Reddit questions (of which 25% questions contain false premises annotated by human workers) split into train/dev/test sets. [54] constructed *(QA)*^{2}, an evaluation set of 570 questions based on frequent search engine queries, which are annotated by expert annotators and crowd workers, and evenly divided between those with and without questionable premises. The *FreshQA* dataset created in [124] is different compared to the previous works mentioned because it contains a fixed set of 500 human-written open-ended questions whose answers by nature can change based on new developments in the world. The questions created can either contain false premises, have a temporal aspect to them, or both. The authors of [124] have grouped questions based on the degree of any possible change to the answer in the future, which means the dataset is dynamic. The authors have committed to weekly updating the *FreshQA* dataset through community efforts.

Knowledge augmented LLMs for false premises

Detection of false premises in question answering with LLMs is still relatively under-explored. Most of the solutions that have been proposed focus on augmenting the LLM with external knowledge to assist it in generating reliable responses. These methods often rely on iterative fact-checking mechanisms to assess the validity of a response. These solutions are not specifically designed to handle false premises, but they share a similar focus and could be adapted to address such issues. *CRITIC* [30] is a framework that allows LLMs to validate their output by interacting with external tools in an iterative *verify-fix-verify* cycle. In this case, the external tool is often a web search engine that allows access to encyclopedic sites like Wikipedia to verify information. This iteration serves as a fact check and aims to improve the reliability of LLMs. The authors of [57] propose a similar framework where an LLM is augmented with information from the internet. Their main takeaway is that utilizing information in an effective manner through efficient prompting can be more beneficial even for models with significantly fewer parameters than simply selecting the largest language model. However, a major disadvantage of this method, at least in the implementation given in [57], is that it requires around fifty API inference calls to the Google search engine, which makes it quite expensive. There are other similar methods [71, 80] as well in terms of their implementation and focus.

In addition to the above methods, there are a few dedicated methods to overcome false premises. The most prominent is *FreshPrompt* [124], which takes advantage of the in-context learning abilities of LLMs to have a few-shot prompting approach that is augmented with knowledge retrieved from the internet to identify false premises. The authors of *FreshPrompt* also created *FreshQA*, which has been used in the creation and fine-tuning process of various commercial systems such as Perplexity AI[4], You.com[5] and Contextual AI's RAG 2.0[6]. It has also inspired a new metric dubbed "Freshness". A model is fresh if it can answer with the most up-to-date information. The authors of [54] evaluated various prompting and reasoning strategies such as CoT and few-shot and found that a *few-shot approach* works best. [139] is one of the few works that focus on tackling false premises by analyzing the internal mechanism of an LLM. The authors find that a small subset of attention heads greatly influences the factual knowledge that is outputted in the form of an LLM response. By constraining such heads during the inference process, the model is able to identify and mitigate false premise hallucinations. With the rise of multimodal models, there has also been an interest in identifying false premises in images [131].

Since our focus is primarily on overcoming false premises, we choose dedicated methods such as *FreshPrompt* [124] and the *few-shot approach* [54] as comparison methods in our evaluation. We also choose self-consistency due to its remarkable performance on various hallucination and reasoning benchmarks [68, 127]. Next, the upcoming sections will introduce logical reasoning briefly, followed by a discussion on natural language reasoning, where LLMs are prominently applied.

2.4. Logical Reasoning

Reasoning is the process of inferring or drawing conclusions based on past experience and available evidence. This usually involves thinking in a systematic and logical manner, utilizing the information from available evidence and past experiences to make a decision [34, 70, 40, 136]. Being able to reason is considered a key distinguishing ability possessed by humans [72]. According to [86, 24], reasoning can be categorized into Deductive, Inductive, and Abductive Reasoning, which is the categorization this thesis follows.

Before delving into the specific reasoning types, the terms *Premise*, *Arguments*, and *Conclusion* are defined.

- "A premise is a proposition—a true or false declarative statement—used in an argument to prove the truth of another proposition called the conclusion." [7]
- "An argument consists of a set of statements called premises that serve as grounds for affirming another statement called the conclusion." [8]

The above primarily applies to deductive reasoning. In inductive and abductive reasoning, the terms *Observation* and *Explanation* are more prevalent and used analogously to *Premise* and *Conclusion*, respectively. These terms are used interchangeably when necessary to explain certain concepts and aspects. The rationale behind the use of these terms will become clearer in the discussions on induction and abduction.

2.4.1. Deductive Reasoning

Deductive reasoning involves reaching a conclusion based solely on the truth of the premises [48, 103]. In deductive reasoning, it is necessary that the conclusion formed follows the premises, and if the

[4] https://www.perplexity.ai/hub/blog/introducing-pplx-online-llms
[5] https://about.you.com/introducing-the-you-api-web-scale-search-for-llms/
[6] https://contextual.ai/introducing-rag2/
[7] https://en.wikipedia.org/wiki/Premise
[8] https://iep.utm.edu/deductive-inductive-arguments/

premises are true, then it is impossible for the conclusion formed to be false. In other words, if the premises are true, then the conclusion will also be true. An example is given below.

Premise 1: All men are mortal.
Premise 2: Socrates is a man.
Conclusion: Therefore, Socrates is mortal.

Rules of Inference

Deductive reasoning is usually achieved by applying certain inference rules [23]. These rules define a schema for formulating conclusions from a set of observations. These observations are often categorized as premises. A few of the most common rules are described below.

Modus ponens

Modus ponens applies when the first premise is a conditional statement (if P, then Q), and the second premise is the antecedent (P) of that conditional statement. The rule allows us to deduce the consequent (Q) of the conditional statement as the conclusion.

Premise 1: All living organisms require water to survive. *(First premise is a conditional statement)*
Premise 2: Roses are living organisms. *(Second premise is the antecedent)*
Conclusion: Therefore, roses require water to survive. *(Conclusion deduced is the consequent)*

Modus tollens

Modus Tollens is another fundamental rule of deductive inference. It applies when the premises include a conditional statement (if P then Q) and the negation of the consequent (¬Q). The conclusion drawn is the negation of the antecedent (¬P).

Premise 1: If the alarm system is armed, then the doors are locked. *(First premise is a conditional statement)*
Premise 2: The doors are not locked. *(Second premise is the negation of the consequent)*
Conclusion: Therefore, the alarm system is not armed. *(Conclusion deduced is the negation of the antecedent)*

Hypothetical syllogism

A hypothetical syllogism is a deductive inference that involves combining two conditional statements to form a conclusion.

Premise 1: If it rains, then the streets get wet. *(First premise is a conditional statement)*
Premise 2: If the streets get wet, then people use umbrellas. *(Second premise is a conditional statement)*
Conclusion: Therefore, if it rains, then people use umbrellas. *(Conclusion deduced is the hypothesis of the first premise with the conclusion of the second premise)*

Validity and soundness

In deductive reasoning, arguments are evaluated based on validity and soundness [24, 22].

Validity: An argument is valid if it's impossible for the premises to be true while the conclusion is false. In other words, if the premises are true, the conclusion must also be true. Validity focuses on the logical structure of the argument, irrespective of the truth of the premises. An argument can be valid even if one or more of its premises are false.

Soundness: An argument is sound if it's both valid and all of its premises are true. Soundness combines validity with the truth of the premises. A sound argument provides a strong justification for accepting its conclusion as true.

Premise 1: If it's raining, then the streets are wet. *(First premise is a conditional statement)*
Premise 2: The streets are wet. *(Second premise is consequent)*
Conclusion: Therefore, it's raining. *(Conclusion deduced is the antecedent)*

This argument is valid because the conclusion logically follows from the premises. However, it may not be sound if the second premise (the streets are wet) is false, such as if the streets were wet due to a water leak rather than rain.

It is important to understand that deductive arguments must follow rules of inference and be sound in their argumentation [24]. If any additional information is included as a new premise, the previously formed conclusion should remain consistent and largely unchanged. If there is a change in the conclusion, it would indicate that the original premise was incorrect or incomplete or that the reasoning process was flawed. If the conclusion does not logically follow from the premises or if it can be affected by new information, the argument may become invalid.

In contrast, inductive and abductive reasoning are less constrained in their evaluation and acknowledge that even if the premise is true and valid, the conclusion may still be false.

2.4.2. Inductive Reasoning

Inductive reasoning involves reaching a hypothesis to explain the observations based on evidence gathered since the observations alone are insufficient to support the hypothesis [100] conclusively. Inductive reasoning is ampliative, meaning that the hypothesis formed is more than simple reformulation [81]. There are debates over the many definitions of inductive reasoning [136], but this thesis follows the viewpoint defined in [25].

- The premise cannot provide conclusive support to the conclusion since the conclusion can simplify or go beyond the information present in the premise.
- The conclusion can be generalized using its premise in a way that can allow the conclusion to be applied to more instances than mentioned in the premise.

An example of inductive reasoning is provided below:

Premise: Every swam seen till now has been white. *(First premise is an observation)*
Conclusion: Therefore, all swans are white. *(Conclusion induced is generalised based on the premise)*

The above example highlights how the conclusions generalised from the premise, even if the premise is true (the observer could have only seen white swans so their belief is valid) can be wrong. There are also black swans, but since the observations contained incomplete or limited information, a completely correct conclusion could not be formed. An example of where the conclusion could be correct is if the observation is that the sun rises from the east, and hence, the conclusion could be that the sun will always rise from the east.

2.4.3. Abductive Reasoning

Abduction can best be described by comparing it with deduction and induction. In deduction, conclusions are formed solely based on the premise. In induction, the number of observations and other considerations in the premise are used to generalize a conclusion to explain the observations. In abduction, given an observation that is either derived or extracted from a conclusion, the conclusion is accepted, assuming it explains the observations (Q and P is the best explanation for Q) [37]. Abductive reasoning can also be considered an inference to the most plausible explanation or best explanation [83]. In other words, the conclusion is assumed because it best explains the observations. Humans perform abduction in everyday scenarios involving dealing with incomplete data to provide a plausible explanation for their situation [4], reading between the lines [16] and counterfactual reasoning [84].

In the example seen in figure 2.4, the conclusion is formed based on the reasoner's understanding of the situation resulting from the premises. Information regarding the possibility of a thief is not provided on the premises, but based on those premises and the general understanding of the world, it is most logical to assume that Jenny's house was robbed by a thief.

Naturally, many explanations can seek to provide the most plausible explanation, and it is possible for these explanations to be contradictory with each other [37]. In such cases, an explanation that can explain the observations comprehensively relative to the other explanations should be chosen. This can done based on probabilities assigned to different explanations. The abduction process is not constrained to its premises when formulating its conclusion. The conclusion formed can introduce new ideas outside of those described in the premise [85, 7]. In the above example, the conclusion that a thief broke into the house is a new idea because there is no direct evidence presented in the observations.

An important distinction from deductive reasoning is that the conclusions formed are only assumptions

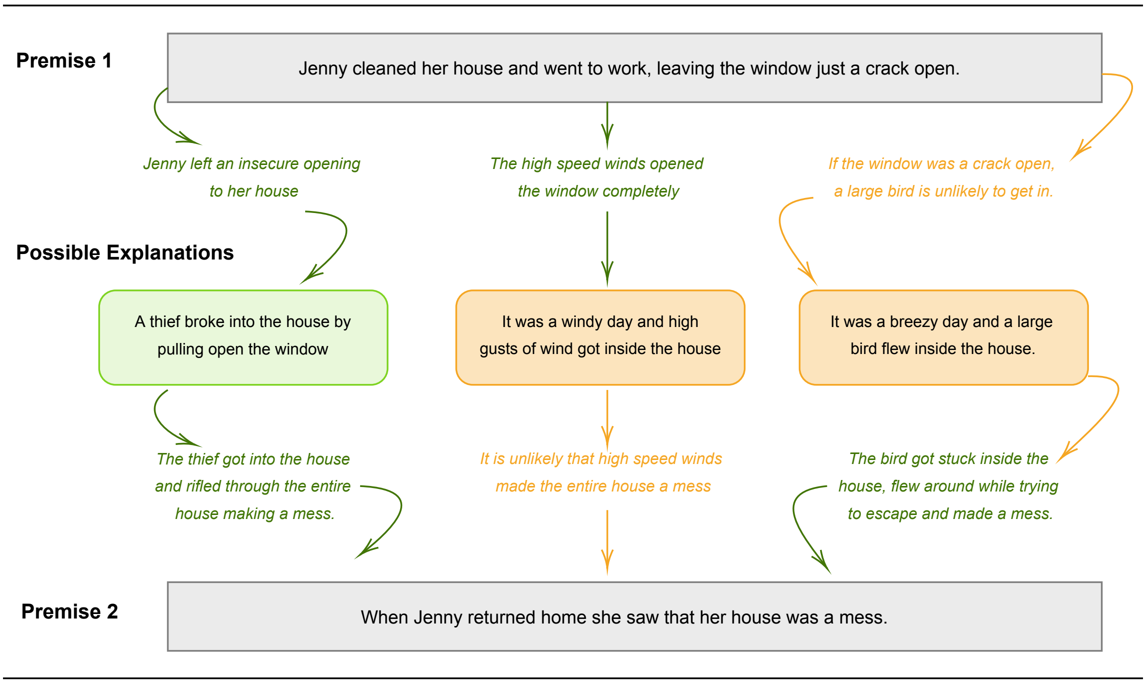


Figure 2.4: An example of Abductive Reasoning for a real-life scenario is presented. When presented with premises 1 and 2, the reasoner formulates different possible explanations (conclusions) based on certain reasonings to explain the premises or observations. In the end, the explanation with reasonings that support both premises is chosen as the most plausible explanation. In this case, the explanation that a thief broke into the house and ransacked the place can satisfy both premises and is reached through plausible reasoning. The other explanations do not satisfy both premises. Naturally, the number of explanations that can be formulated is huge, but this is restricted to three possibilities for illustration purposes.

and may be retracted if new contradictory information is acquired. In other words, abductive reasoning is nonmonotonic [37]. In the above example, if it is later found that Jenny lives in an area with frequent high-speed winds, then maybe the mess in the house was caused by the wind instead of a thief, even though it is quite unlikely the entire house would be in a mess. Abductive reasoning, by its definition, also enables the reasoning process to go in a backward direction compared to deductive reasoning since the outcome is known and aims to explain that outcome. If the above example is solved using a forward reasoning approach (deduction), the conclusion (outcome) becomes premise 2, and premise 2 is the conclusion derived from the premises. This sort of reasoning requires all possible information in the form of observations, which is less likely to be available in real-world scenarios.

2.5. Natural Language Reasoning with LLMs

This section will first describe natural language reasoning, why and how LLMs are utilized, and certain methodologies followed while designing reasoning approaches with LLMs.

2.5.1. Natural Language Reasoning

The previous sections explained the different logical reasoning methodologies along with their definitions and basis in logic. These methods have traditionally been formulated using formal language like first-order logic. However, reasoning through formal language has proven to be problematic, with expert systems failing due to missing data in the knowledge base, along with the requirement of having knowledge represented in symbolic format, which requires extensive human annotation [79, 17]. Moreover, practical applications involving reasoning would likely provide their task context and other information in natural language. Representing natural language in a formal language is challenging, with difficulties arising in handling raw text and questions regarding which aspect of the input should be considered for representation in the reasoning process [136, 135]. Our proposed reasoning approach is developed for natural language reasoning; therefore, before proceeding ahead, a formal definition from [137] is provided below:

> *”Natural language reasoning is a process to integrate multiple knowledge (e.g.encyclopedic knowledge and commonsense knowledge) to derive some new conclusions about the (realistic or hypothetical) world. Knowledge can be from both explicit and implicit sources. Conclusions are assertions or events assumed to be true in the world, or practical actions.”*

Designing a system for natural language reasoning using a rule-based system or formal language would be impractical due to the various aforementioned challenges. A model needs to be able to generalize to various reasoning situations, adapt to uncertainties and ambiguities realistically, and perform other natural language processing tasks. Language models, specifically large language models (LLMs), are an ideal candidate for achieving reliable natural language reasoning, as described in the next section.

2.5.2. Reasoning in LLMs

Large Language Models (LLMs) have shown remarkable performance on natural language tasks [145, 3] such as machine translation, information retrieval, and question answering. This is largely due to their ability to capture semantic information in the form of embeddings [74], which has enabled them to be used extensively for natural language reasoning [91, 113, 8]. LLMs are able to reason and process to mostly understand the meaning of the different symbols and raw text. This has led to LLMs being used to handle raw input and construct rules for information matching [135]. Moreover, LLMs can also act as knowledge bases since they possess knowledge themselves [19]; this enables them to provide reasonable responses even when context or data is missing from an external knowledge base [136, 114]. All this makes LLMs an ideal candidate for natural language reasoning since they can achieve every aspect of the above definition and overcome their challenges. An important distinction from formal reasoning is that knowledge sources are implicit and explicit. This can make formulating and representing knowledge an enormous task since knowledge is effectively endless. However, LLMs being trained on massive datasets and possessing exceptional natural language understanding abilities enables natural language reasoning in a much more refined and seamless manner.

The above description might present the notion that LLMs are truly capable of reasoning and understanding the context that is provided to them to generate a suitable response. This is not the case as demonstrated in [67], where the authors argue that ”language ability does not equal to thinking or reasoning” in LLMs. They further claim that LLMs have poor reasoning skills compared to humans, with most of the supposed reasoning being attributed to pattern matching with their corpus of training data and simply outputting the next most probable word. These are artifacts of the Supervised Fine Tuning (SFT) training process for LLMs [143], which can cause the LLM to simply mimic the patterns found in the context to generate a response with some surface-level relevance to the provided input [105, 56]. Moreover, LLMs are trained to minimize the word prediction error on large corpora [53, 107], and this does not always result in the most coherent answers. These two reasons combined challenge the reasoning ability of LLMs with various works exploring the reasoning capabilities largely reaching the same conclusion [40, 134]. These limitations of LLMs as reasoning agents are further elaborated in appendix A.

However, the fact remains that there is a lot of research that showcases the reasoning abilities of LLMs by utilizing their implicit and explicit knowledge. By adapting to the particular dataset, [15] initially illustrated that LLMs can use deductive reasoning based on explicitly given natural language statements, which can zero-shot transfer to various tasks without prior training. Furthermore, [114] demonstrated that LLMs can integrate memorized implicit taxonomic and real-world knowledge with explicitly provided information to facilitate deduction. Although language models with in-context learning were initially believed to lack the ability for multi-step reasoning, recent discoveries have revealed that their reasoning abilities can be activated by constructing forward reasoning paths leading up to the ultimate solution [129]. This technique, known as Chain-of-Thought (CoT) prompting, has enabled unlocking their reasoning capabilities. Additionally, LLMs have demonstrated the ability to engage in multi-step reasoning not only in a few-shot setting [69] but also through zero-shot ”Let’s think step by step” prompts, enabling them to generate intermediate steps even in zero-shot scenarios automatically [92, 55]. Remarkably, LLMs have shown the capacity to learn from the reasoning paths they generate themselves [39, 140]. Backward reasoning has also proven useful in tasks requiring multi-step reasoning such as multi-hop questions [82]. Numerous studies have already been conducted to evaluate the capacity of Language Models (LLMs) from different reasoning angles, such as multilingual reasoning

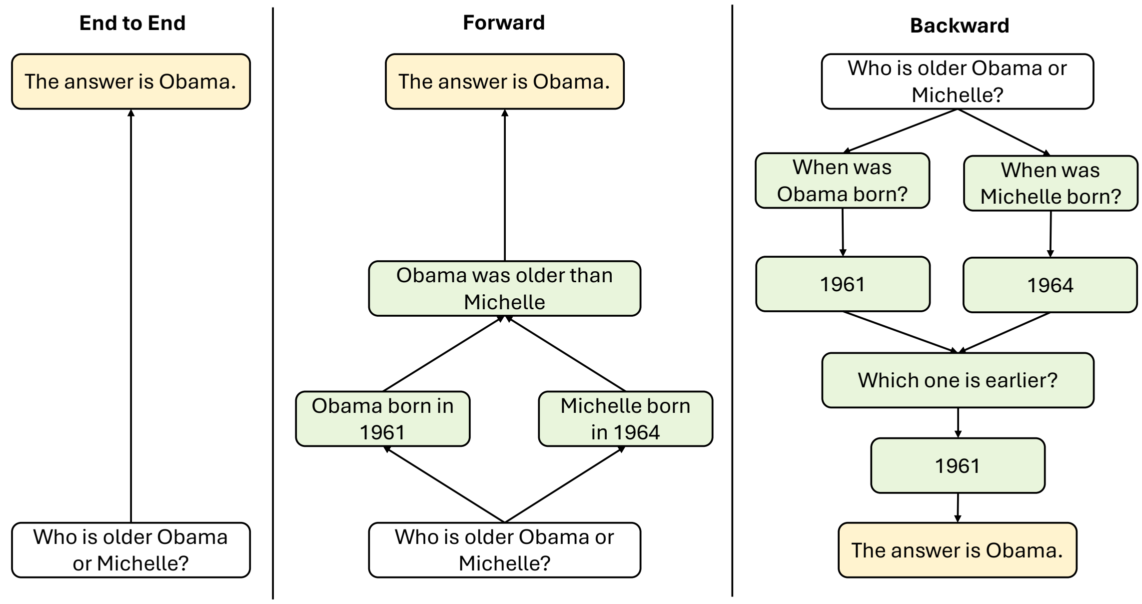


Figure 2.5: An example to demonstrate the reasoning procedure and direction.

[5], commonsense reasoning [9], and mathematical reasoning [44] which have showcased promising performance and potential for utilizing LLMs for reasoning.

2.5.3. Reasoning Direction

When discussing reasoning approaches, it's important to consider the direction of reasoning as it significantly impacts the reasoning process. There are primarily three types of reasoning methodologies distinguished by how they generate their reasoning paths. End-to-end reasoning predicts final answers directly without intermediate steps, whereas forward and backward approaches generate reasoning paths that include one or more intermediate steps and conclusions. These paths illustrate the reasoning process connecting premises to conclusions. An example of this can be seen in figure 2.5.

A majority of the research [134, 137] on LLM reasoning has focused primarily on deductive reasoning, which essentially involves reaching a conclusion from a set of premises. This sort of reasoning is most often seen in expert systems and decision support systems. The direction of reasoning follows from the premises to the conclusion, with the understanding that the conclusion is derived solely from the premises. This reasoning direction is the most natural and often termed forward reasoning. Another approach for reasoning is backward direction, which involves breaking down each problem into sub-problems and solving them until an answer has been reached. This reasoning direction can be associated with the process of abduction. Backward reasoning and abduction are often used analogously, but this thesis follows [137] and makes a distinction because backward reasoning is more comparable to a general strategy or approach, whereas abduction represents a specific method within that strategic framework.

2.5.4. Abductive Reasoning in LLMs

Unlike other reasoning types, abductive reasoning has not been studied extensively for natural language reasoning. While its basis has long been in logic [37], the first work to explore the possibility of abduction with language models was done in [7], where the authors created the first abductive reasoning benchmark dataset to promote future studies. Following this, there has been an increase in explorations of abductive reasoning evaluation for LLMs [134, 5, 136, 40]. The abduction capabilities of LLMs have largely shown potential, at least when compared to other forms of reasoning, such as deduction. One of the possible reasons for this was hypothesized in [134] where the authors argue that the setting of abductive reasoning requires the LLMs to reason in a reverse direction, which can likely activate the reasoning process of the LLMs to be more robust. Research into the use of abduction for accomplishing tasks has also shown promise with works like generating more reasonable and sound explanations [115], predicting future events based on past information [111], and improving the generation of personalized healthcare recommendations for patients [27].

Building on the promising potential of LLMs' reasoning abilities and the diverse range of applications where abductive reasoning has proven effective, the next chapter suggests an approach to enhance the utilization of LLM's reasoning capabilities. This involves adopting a backward reasoning method and integrating it with abductive reasoning to effectively detect and correct false premises.

3

FPDAR: False Premise Detection with Abductive Reasoning

This chapter will explain how abductive reasoning could mitigate false premises by first outlining the motivation behind our approach in a generalized context. It will then delve into the formulation of this approach for a Knowledge-Intensive Question-Answering setting, beginning with defining the QA environment and subsequently detailing the approach for QA. We will then integrate our methodology into a dedicated method called **F**alse **P**remise **D**etection with **A**bductive **R**easoning **(FPDAR)**. Finally, section 3.2.4 will discuss the operation of FPDAR across different scenarios, highlighting the conditions under which the technique is successful and identifying its inherent limitations.

3.1. How is Abductive Reasoning used to Detect False Premises?

The motivation of the detection approach is that given a solution to a problem, the solution should enable the model to reach the initial premise or the starting point of the problem. If the initial premise is not being reached, then that would indicate that the wrong or less plausible answer was considered the solution. In such cases, the model is instructed to generate the most plausible explanation using abductive reasoning, taking into account the context (any additional information) and conclusion (incorrect answer). This explanation could introduce new information not present in the context and initial answer but is taken as the final answer since the most plausible explanation is considered to be mutually exclusive [146, 29] to other explanations; that is, one explanation being most plausible automatically rules out other less plausible explanations which are likely to be hallucinated responses.

The motivation explained above is quite intuitive. It is a form of verification but in the backward direction by employing a bottom-up approach to reaching the initial starting point as described in section 2.5.3. This strategy should eliminate unlikely explanations at an early stage, as these explanations would fail to connect with the initial premise. Also, since our approach is designed to reason backward, it works naturally with the abduction process. While the approach may seem intuitive, it requires specific formulation to be effectively applied in a question-answering (QA) setting beyond the general description provided above. The next few sections will describe how this is achieved.

3.1.1. Task Formulation

Our proposed approach is applied to a Knowledge Intensive Question Answering task setting [89] where external knowledge can be obtained to assist in question answering (fig 3.1). Given an initial question Q, the task of the Large Language Model (LLM) is to provide an answer that adequately addresses the question. However, since the question Q in our scenario can contain false premises, which may sometimes overlap with dynamically changing information, the LLM is augmented with up-to-date, relevant external knowledge. External knowledge may be acquired through knowledge bases and tabular data but in our case, we use question Q to retrieve supporting documents by performing a web search (more details in section 3.2.1). These recent and relevant web search results serve as

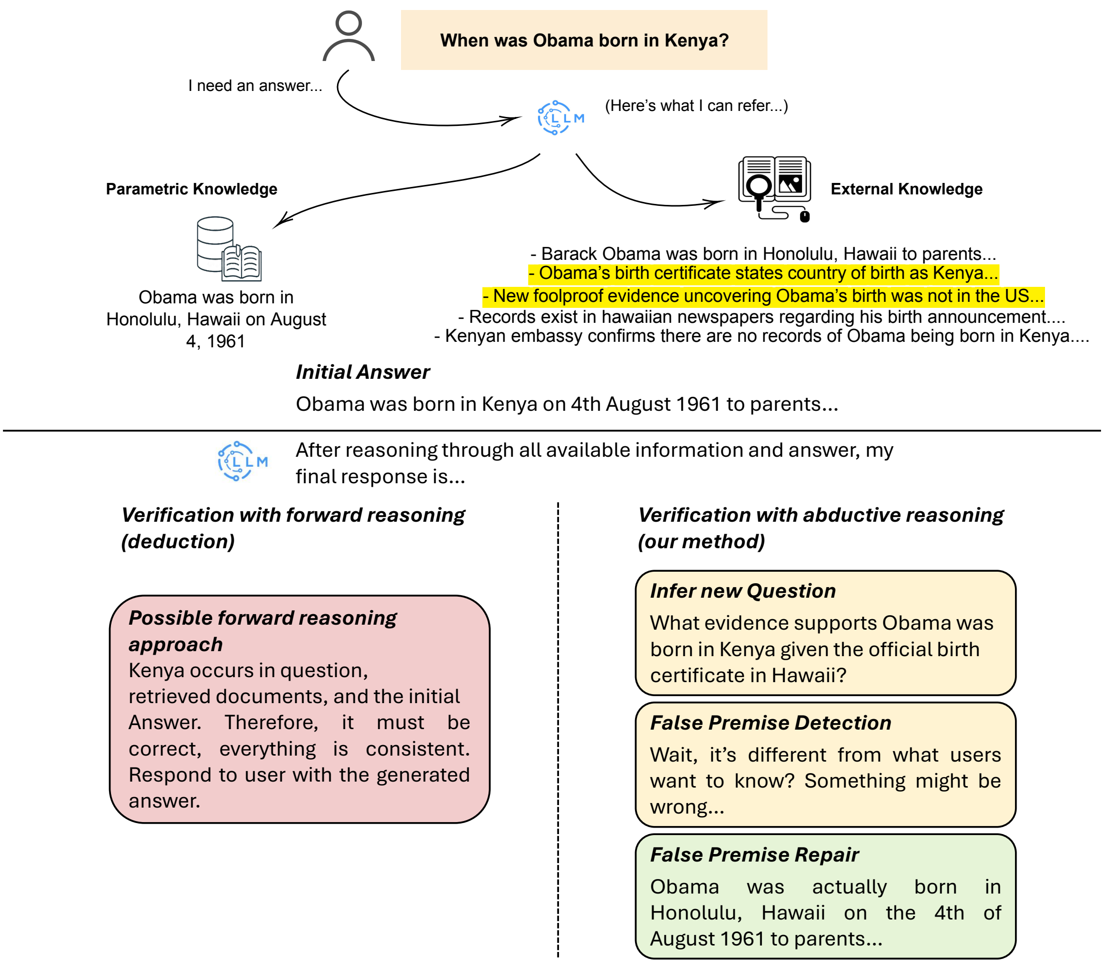


Figure 3.1: An illustration of the Knowledge Intensive QA setting and its formulation with an example of a question containing a false premise. The LLM is expected to provide a suitable response using the available knowledge (parametric and external). External knowledge is primarily provided to counter those questions that contain false premises that are dynamically changing. The answer should either point out the false premise or make it obvious in its response that certain premise(s) within the question are invalid.

additional context X and are intended to assist the model in recognizing false premises. The LLM is expected to analyze both the external knowledge and its own parametric knowledge to generate a suitable response Y to question Q. If a false premise is detected, then our proposed abduction method generates an alternate explanation Z, which serves as the final response. The model's response (Y or Z) may point out the presence of a false premise or make it obvious with its response. The primary criterion we are concerned with is that the response is always factually correct (correctness). At its core, this is our task setting, and every method we evaluate begins with this setup unless explicitly specified. This ensures that during comparisons among different methods, they are all on an equal footing, and no method has an unfair advantage over another. Any advantage a method may possess is inherent to it and not caused by external factors.

3.1.2. False Premise Detection for QA

Abductive Reasoning seeks to provide the simplest plausible explanation given an event [7]. Given certain context X and conclusion/final observations/initial answer/outcome Y, it allows us to infer explanation Z. Explanation Z is chosen because it is assumed it best explains conclusion Y [37] compared to the other explanations [146, 29]. An important point to recall is that explanation Z is not limited to the observations and has the potential to introduce new ideas [7, 146]. Additionally, explanation Z is generated through reasoning in a backward direction. It has been hypothesized [134] that these char-

Human: Which country was the largest producer of rice in 2020? Give me your best guess and answer as concisely as possible.

LLM: China was the largest producer of rice in 2020.

Human: I don't think that's right. Are you sure?

LLM: I apologize for the error. According to FAO data, India was the largest producer of rice in 2020. Thank you for bringing it to my attention.

Human: So what's the answer? Give me your best guess and answer as concisely as possible.

LLM: India was the largest producer of rice in 2020.

Figure 3.2: An example of LLM sycophancy and hallucination snowballing. The LLM's initial response was accurate. However, due to the influence of biased feedback, which the model chose to comply with, the response was changed. Subsequently, the LLM continued to uphold its revised response to ensure consistency. This example was taken from [106].

acteristics enable the model to reason more effectively beyond the information provided in the context *X*. This theoretical background of abductive reasoning is used as the foundation for the proposed false premise detection approach for QA and is as follows:

1. Given an initial question *Q*, generate an initial conclusion (answer) *Y* using any other method.
2. Given some additional context *X* and a conclusion *Y*, reason backwards and generate new question *Q'* using abductive reasoning. (X & Y $\rightarrow$ Q')
3. Questions *Q* and *Q'* will compared to check if they lead to the same conclusion, i.e., question intent verification.
4. If questions *Q* and *Q'* are verified to lead to the same conclusion, then the initial conclusion *Y* is accepted as the final answer.
5. If questions *Q* and *Q'* do not lead to the same conclusion, then an alternate explanation *Z* is generated using abductive reasoning through context *X* and initial conclusion (answer) *Y*. (X & Y $\rightarrow$ Z)
6. This alternate explanation *Z* is taken as the final LLM response to question *Q*.

The approach, as described in the steps above, starts by first generating an initial answer *Y* for the question *Q*. Step 2 is to check whether this initial answer *Y* is correct through a proxy. Instead of checking the veracity of the answer directly, the approach investigates whether the initial question (premise) *Q* can be successfully reached from the initial answer *Y*. To do this, another question *Q'* is generated by performing abduction using context *X* and the initial answer *Y*. In step 3, the initial question *Q* and the newly generated question *Q'* are compared, with the goal being to verify whether if both these questions were answered would lead to the same overall answer *Y* indicating that *Y* is correct. This task in NLP literature is termed Question Intent Verification or simply Duplicate Question Detection [128]. In step 4, if the questions *Q* and *Q'* are verified to be the same, then that would indicate that the initial answer *Y* was correct; this is then taken as the final answer. However, if the verification of the questions resulted in the question intent being different, then this would indicate that the initial answer *Y* is likely to be incorrect. An alternate answer in the form of an explanation *Z* is generated using abductive reasoning through context *X* and initial conclusion (answer) *Y*. This explanation *Z* is taken as the final due to it being the most plausible explanation, which should, by definition, eliminate other non-plausible (wrong) answers [146, 29]. The non-plausible explanations are likely to be hallucinated, and by avoiding them, hallucinations are being avoided, although they are not explicitly being detected. A high-level example of this method can be seen in figure 3.1. The reasoning behind why question *Q'* is so drastically different compared to *Q* will become apparent in section 3.2.4. We refer to our method as a **pseudo-detection** approach because it doesn't involve explicit false premise detection. Instead, it verifies the question's intents, operating on the intuition that if the intents match, the question is likely valid. We term this approach as **F**alse **P**remise **D**etection with **A**bductive **R**easoning **(FPDAR)**.

Why is verification done through a proxy?

Directly assessing the veracity of the answer *Y* by instructing the LLM to verify its response or compare it with any additional information could lead to an undesirable LLM behavior known as sycophancy. It is the tendency of LLMs to generate responses that match the user's feedback or beliefs over the truthfulness. LLMs are often trained and fine-tuned based on human preference judgment. Still, it has been investigated and empirically found that most of these judgments tend to favor the user's preconceptions rather than accuracy. This can cause models to output responses that sometimes sacrifice truthfulness in favor of sycophancy [106]. Sycophancy can also lead to hallucination snowballing, which is the tendency of the model to double down on its previous incorrect response to maintain consistency in subsequent conversations [142]. An example of these phenomena can be seen in the figure 3.2.

To avoid our approach from falling into the same trappings, the answer veracity comparison is done through a proxy, which in our case is question intent verification.

3.1.3. Why should Abductive Reasoning work?

Abductive reasoning works well for three main reasons. Firstly, abduction, by definition, seeks to provide the most plausible explanation. As seen in figure 2.4 in situation-based scenarios, this means choosing the most likely explanation to explain the outcome. In situations where factual information is concerned, such as our QA setting, this would mean analyzing the various points presented and the model utilizing that information along with its own knowledge of the world to provide a sensible, factual answer. This answer would be the most plausible explanation, and since it is factual, it would most likely also be correct. Applying another reasoning approach, such as deduction, would not be as effective because deduction, by definition, requires all premises to be true for the conclusion to be true. This is fine for valid questions, but for false premises, this can cause the reasoning to lead to an illogical answer. Second, since abductive reasoning explanations are mutually exclusive [146, 29], it would mean that the most plausible explanation would automatically eliminate other less likely explanations, which would most likely also have been incorrect. This finding, combined with the first reason, benefits abductive reasoning, especially in cases where factual information is concerned. Lastly, abduction as a reasoning process enables the reasoner to think backward, which is able to minimize less plausible explanations early and work towards the correct answer since it is a bottom-up approach [40]. This has been hypothesized to activate the LLMs to think much more rigorously than other reasoning methods [134].

3.2. FPDAR

Our approach will now be structured into three distinct stages that collectively form FPDAR. This section will explore the implementation of these processes and provide insights into the rationale behind our decisions.

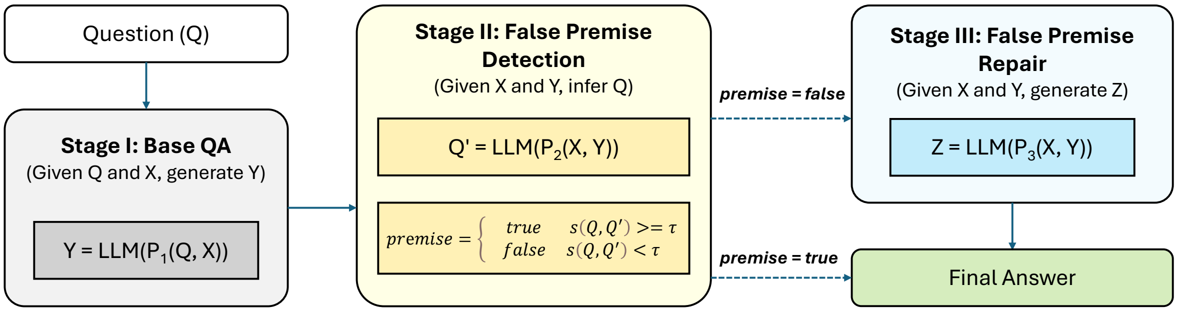


Figure 3.3: FPDAR workflow for detecting potential false premises and avoiding hallucinations for QA. The entire process is composed of three steps. The first is the Base QA stage, which involves gathering the additional context X and the initial answer Y. Context X is the scraped web search results and is included in the prompt to generate the initial answer Y. Stage two detects potential false premise based on the question intents of Q and Q', and accordingly, the premise is decided. If the premise is true, then answer Y is taken as the final response; else, stage three is triggered, and an alternate explanation Z is generated using abductive reasoning.

3.2.1. Stage I: Base QA

The purpose of the first stage is to create an initial starting point. Our method FPDAR (fig 3.3) is post-hoc and requires an initial starting point (answer Y and context X) before the actual abduction process from stages two and three can start. The initial answer Y is generated using a combination of zero-shot [55] with Chain of Thought (CoT) reasoning [129] based on question Q and context X. The prompt used to achieve this in our experiments is denoted as prompt P_1, which can be found in Appendix B.1.

$$\mathrm{Y} = \mathrm{LLM}(\mathrm{P}_1(\mathrm{Q}, \mathrm{X})) \tag{3.1}$$

What is Context X?

Relevant and up-to-date information from a search engine is scraped using SerpAPI[1]. For each question Q, relevant information is gathered by using Q as verbatim and querying the Google Search Engine. Search result information such as *organic results*, *answer box*, and *related questions* are used to augment the LLM with up-to-date information relevant to the question. For each result, the *name* and *text snippet* are extracted and included in the prompt. This information acts as our context X. A visual representation of various fields can be seen in figure 3.4. This implementation of RAG is quite simple and does not incorporate means to order the results according to a relevance metric. Details regarding the implementation can be found here B.2.4.

Need for Context X

The most essential aspect of the initial answer generation is the RAG. It is applied at generation time and acts as context knowledge for the LLM [109, 110]. This is important for our overall solution because questions from the FreshQA dataset contain a temporal aspect in addition to false premises. This means there can be questions to which the answers can vary depending on the real-world dynamics. For example, a simple question such as ”Who is the first female president of the United States of America?” has both a false premise and also contains a temporal aspect. Till the present day of July 2024, there has not been a female president in the US, which means that the question contains an incorrect assumption of there being a first female president in the US. In addition, this question also has a temporal aspect because there is always a possibility that a female president could be elected in future elections. LLMs, while capable of being utilized as a knowledge base, suffer from the disadvantage of stale information due to the changes happening in the real world at all times [110, 132]. If events

[1] SerpAPI (`https://serpapi.com`)

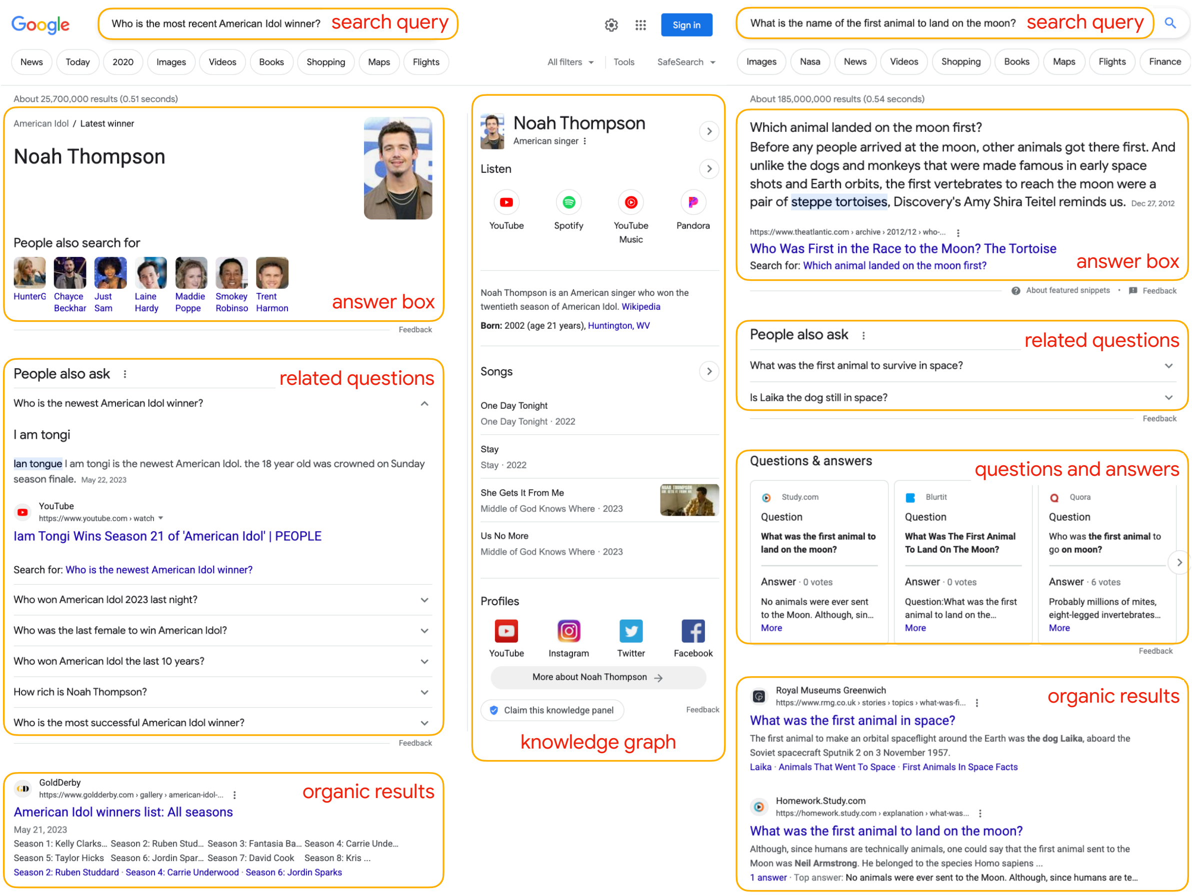


Figure 3.4: An example of the scraped information that is retrieved from SerpAPI. Various search parameters containing different information are retrieved; however, for our experiments, only the *answer box*, *related questions*, and *organic results* were used. Each of these results contains an associated text snippet along with other information, such as the name of the website. This was the information that was deemed important to provide the necessary, updated context to the LLM. The above image illustrating the additional context X is taken from [124]

in the world occurred which would change the premise of the question then the LLM would likely not have that information in its parametric knowledge due to stale data. It is infeasible to continuously train the model due to practical constraints such as cost, and computation resources. A remedy to this can be augmenting the LLM's knowledge with external knowledge and this is known as RAG. Fortunately, LLMs possess strong in-context learning capabilities [21], which makes RAG viable. The RAG in this implementation is a simple Web API call via SerpAPI to the Google search engine. The first two web search results, the related questions, and the answer box are provided to the model in the prompt for the initial answer Y generation. These web search results are necessary to generate a reasonable initial answer Y; not having any additional context would stack the odds against the LLM by a large degree since now the model would only be able to answer historical questions correctly since the other answers would likely contain stale data or hallucination.

3.2.2. Stage II: False Premise Detection

The main purpose of the False Premise Detection stage is to identify potential false premises through means of a proxy and, in turn, avoid hallucinations. This is achieved by first inferring question Q' and comparing its intent to the original question Q using semantic similarity. Another approach to achieving this is through Natural Language Inference (NLI)[2] [95], which involves analyzing the logical relationship between a premise and a hypothesis—in our case, Q and Q', respectively—to determine whether there is entailment or contradiction. However, since our task of intent comparison closely

[2] `https://www.sbert.net/examples/training/nli/README.html`

resembles the Quora Duplicate Questions task[3], where semantic similarity is a straightforward and effective approach, we have decided to keep our method simple and adopt semantic similarity as well.

The False Premise Detection process is achieved in two distinct steps as described below:

1. **Infer Question *Q'***
   A new question *Q'* needs to be inferred back from the context *X* (web search results) and conclusion (initial answer) *Y* using abductive reasoning. Question *Q'* is generated by prompting the LLM to perform abduction. The process of inferring a question from an answer is not a standard NLP use case or task. This requires the creation of a custom prompt P_2 to infer *Q'*:

   $$Q' = LLM(P_2(X, Y)) \tag{3.2}$$

   The snippet of the prompt P_2 containing the question generation aspect using abductive reasoning is given below. Observation *o1* is the context *X* and Conclusion *o2* is the initial answer/outcome/-conclusion *Y*. The entire prompt for question generation, along with all the other prompts, can be found here B.2.

   > *"Given the conclusion o2 and observation o1 along with your general knowledge of the world, engage in abductive reasoning to form a question that addresses the information provided in the conclusion o2. Ensure that the question is clear and directly relevant to the conclusion even if it introduces additional context."*

   Need for Context X
   Context *X* is used along with initial answer *Y* to infer question *Q'*. The primary purpose of including *X* is to provide sufficient up-to-date background knowledge to the LLM while constructing question *Q'* using abductive reasoning. If the initial answer *Y* is incorrect, during the process of abduction, the LLM should realize this since the prompt P_2 to infer *Q'* explicitly states that a question has to be generated that satisfies the answer *Y*, the LLM will try to formulate its question *Q'* in a way that asks for more information or clarification or new evidence to support answer *Y*. In most cases, this is a futile attempt since this additional information does not exist to support answer *Y* as this answer is likely to be hallucinated. This will lead to the next question intent comparison step failing and leading to the detection of a potential false premise and hallucination response *Y*. Context *X* is only required for questions that are quite recent and ask about information that happened after the LLM's training point or changes frequently. If question *Q* is a historical question, then context *X* is not required during the detection process. In such cases, the question *Q'* inferred will almost be the same in terms of phrasing. Examples highlighting the different scenarios are presented in section 3.2.4.

2. **Question Intent Verification**
   The intent similarity $s(Q, Q')$ between *Q* and *Q'* is calculated using the cosine similarity of their sentence embeddings obtained from SentenceBERT[4]. If this similarity exceeds a predefined threshold τ ($\tau \in [0, 1]$), we assume the original question *Q* does not contain a false premise and use the initial answer *Y*. If the similarity is below the threshold, we consider the possibility of a false premise in *Q* and proceed with the next stage to repair answer *Y* by generating an alternate response.

   $$premise = \begin{cases} true, & s(Q, Q') \geq \tau \\ false, & s(Q, Q') < \tau \end{cases} \tag{3.3}$$

3.2.3. Stage III: False Premise Repair

The Repair stage is only initiated when question *Q* is identified to contain a false premise. The main purpose is to analyze the available information (answer Y and context X) and generate an alternate

[3] https://huggingface.co/datasets/sentence-transformers/quora-duplicates

[4] This specific model from hugging face was used. (https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens)

response using abductive reasoning. Abductive reasoning enables the LLM to consider various explanations and their probabilities to arrive at the most plausible conclusion. This should offer enough flexibility in reasoning to avoid any potential false premises. The alternate explanation *Z* is generated with prompt P_3 using the context *X* and conclusion *Y* to act as the final answer.

$$Z = LLM(P_3(X, Y)) \tag{3.4}$$

The snippet of the prompt containing the question generation aspect using abductive reasoning is given below. Observation *o1* is the context *X* and Conclusion *o2* is the initial answer/outcome/conclusion *Y*. The entire prompt P_3 can be found here B.3.

> *"Given observation o1, which may or may not be accurate, and the subsequent conclusion o2 derived from o1 alongside general knowledge of the world, analyze the plausibility of o2 using abductive reasoning. Consider the potential explanations for the observed phenomenon in o1 and evaluate whether o2 logically follows from these premises and aligns with our broader understanding of reality. Provide a reasoned assessment of the coherence and credibility of o2 in light of the available evidence and background knowledge."*

The prompt instructs the LLM to analyze both the context *X* and the conclusion *Y* using abductive reasoning, taking into account the different possible explanations. The most plausible explanation *Z* is generated and chosen as the final answer since abductive reasoning explanations are mutually exclusive [146, 29] as explained in section 3.1.3.

Need for additional Context X

Context *X* in the generation process is used similarly to the first stage for answer *Y*. The additional information provides sufficient context for the abductive reasoning process.

3.2.4. FPDAR Workflow Scenarios

This section will explain how FPDAR operates in different scenarios, including when it may fail and the key dependencies required for its success.

False Premise - Successful Execution

Figure 3.5 showcases how FPDAR should function with a false premise. Given question *Q*, context *X* is acquired, and then the initial answer *Y* is generated. At first glance, question *Q* looks like a normal question, but it contains a false premise. President Andrew Johnson was not elected. Rather, he assumed that position after the assassination of President Abraham Lincoln since he was the vice president. Answer *Y* is hallucinated since it states Andrew Johnson was elected in 1864. Note that this question is historical, so the LLM likely has the correct information stored as parametric knowledge. Additionally, the correct information is also present in the web search results *X*, but the LLM is still unable to effectively reason and formulate a proper answer *Y*.

The incorrect answer *Y* and the context *X* leads to the formation of question *Q'*. In this case, during the generation of question *Q'*, the LLM realizes that the answer *Y* is unlikely to be plausible, but since it has to formulate a question that, if answered, would lead to answer *Y*, i.e., the intent of the questions are the same, question *Q'* ends up asking for additional information or clarification that could make answer *Y* true. This can be seen in the *Q'* generated since it asks about a significant event that led to Andrew Johnson becoming president. However, this is futile since this additional information that could make the initial answer *Y* true does not exist. Performing question intent comparison on *Q* and *Q'* would result in the questions being different, signifying that the initial answer *Y* is an incorrect response caused by the presence of a false premise in question *Q*. Initial answer *Y* cannot be used, and an alternative answer has to be provided, which is done using the repair stage.

In the repair stage, an alternative response is generated using context *X* and answer *Y* through abduction. The LLM is able to effectively reason that the plausibility of answer *Y* is unlikely. The alternate explanation *Z* is generated, which is chosen as the final answer.

Stage I: Base QA

Question (Q) - In what year was Andrew Johnson elected as President of the United States?

Context (X) - [{'name': 'Who is U.S. 17th president?', 'snippet': "Andrew Johnson was the 17th president of the United States, serving from 1865 to 1869. Johnson was Abraham Lincoln's vice president and so assumed the presidency after President Lincoln was assassinated."}
{'name': 'Andrew Johnson', 'snippet': 'With the assassination of President Abraham Lincoln, Andrew Johnson became the 17th President of the United States
(1865-1869), an old-fashioned southern ...'}

Initial Answer (Y) - Andrew Johnson was elected as President of the United States in 1864.

Stage II: False Premise Detection

Inferred Question (Q') - What significant historical event occurred in 1864 that led to Andrew Johnson being elected as President of the United States in 1865?

Q' not same Q in terms of the info that could be provided, proceed to stage III

Stage III: False Premise Repair

Explanation (Z) - The conclusion that Andrew Johnson was elected as President of the United States in 1864 is not plausible because historical records indicate that Abraham Lincoln was the President in 1864, and Andrew Johnson became President after Lincoln's assassination in 1865.

Ground Truth - Andrew Johnson was not elected to the Presidency but instead assumed the position following the assassination of Abraham Lincoln

Figure 3.5: FPDAR False Premise example with successful execution

True Premise - Successful Execution

Figure 3.6 showcases an example of how the workflow operates during a true/valid premise question with correct answer *Y*. Answer *Y* is generated using context *X* and LLM's parametric knowledge. Question *Q* is a valid question with no false premises. Question *Q'* is inferred back during the false premise detection stage. Since the answer *Y* is correct and the question *Q* premise is true, question *Q'* is nearly identical in terms of its intent to question *Q*. In most cases, when the question Q is valid, the inferred question Q' will likely be a paraphrase of Q. The question intent comparison succeeds, and the initial answer *Y* is taken as the final answer. In this scenario, there was no need to proceed to the repair stage.

True Premise with Abductive Explanation - Successful Execution

Figure 3.7 showcases an example of how the workflow operates during a true/valid premise question when the question intent process fails for a correct answer *Y*. This situation is crucial to highlight because it points to two important aspects of our method. First, the question generation for *Q'* is not robust enough to identify the main content of answer *Y*. It can get distracted by unnecessary information when inferring question *Q'*. In our implementation, the LLM is used to infer *Q'* using abduction, but the model is relatively free to focus on the aspect it thinks is the most relevant or, rather, is being presented in the conclusion *Y*. This can cause *Q'* to formulate its basis on an irrelevant aspect of question *Q*. Since this can be the case, it also weakens our initial motivation to be able to reach the original question Q to verify the veracity of answer Y, at least from an implementation perspective. Second, abductive reasoning in the repair stage should ensure that the answer Y remains consistent from a content perspective, assuming the initial answer Y was correct. This can also be an issue because, due to LLM sycophancy and the presence of biased feedback, the model could be inclined to modify its original response, but in most cases, with abduction, that should be avoided.

Returning to the example, it is observed that answer *Y* is correct, but it proceeds to provide additional

Stage I: Base QA

Question (Q) - After which country did Marie Curie name the first element that she discovered?

Context (X) – [{'name': 'Marie Curie – Questions and answers - NobelPrize.org', 'snippet': "And Marie was proven right: in 1898 the Curies discovered two new radioactive elements: radium (named after the Latin word for ray) and polonium (named after Marie's home country, Poland)."}
{'name': 'Why did Curie change her first name from Maria to Marie?' 'snippet': "Three years later in Paris, Maria met her research partner and future husband Pierre Curie, a tutor at the School of Physics and Chemistry...}]

Initial Answer (Y) - Marie Curie named the first element that she discovered after her home country, Poland.

Stage II: False Premise Detection

Inferred Question (Q') - How did Marie Curie choose the names of the two new radioactive elements she discovered in 1898?

Q' same Q in terms of the info that could be provided, consider Y as final answer

Stage III: False Premise Repair

Explanation (Z) – This stage was not required

Ground Truth - Poland

Figure 3.6: FPDAR True Premise example with successful execution

information by explaining why Socrates was late. This leads to the question *Q'* being formulated focusing on why Socrates was late rather than who was late to Plato's Symposium. As a result, questions *Q* and *Q'* are not the same in terms of their intent. The repair stage is triggered, and essentially, the same explanation *Z* is outputted and taken as the final answer.

True Premise with Abductive Explanation - Unsuccessful Execution

The previous examples showcased how the workflow can reason effectively, overcome false premises, and output the correct response. However, it is important to note that in the entire workflow, there is no explicit false premise detection or even hallucination detection present; rather, these are avoided through the use of abductive reasoning. Recall how our method FPDAR is termed as **pseudo detection**. The workflow depends on at least the context *X* and/or the initial answer *Y* to be reliable, relevant, and correct. If both of these are wrong or irrelevant, then the reasoning process will not lead to a factually correct response, even if the LLM thinks that the explanation it generated is logically correct.

Case in point, the scenario in figure 3.8 showcases this limitation of the workflow. Question *Q* asks how many presidents Peru has had in the last five years. This question is completely valid and has a factually correct answer. However, the context *X* acquired is irrelevant. Recall that these are web search results obtained after querying question *Q* verbatim, and this is not guaranteed always to return relevant results, especially since web search engines struggle with natural language queries [12]. The LLM also does not possess relevant, up-to-date information regarding the question since this is an event that happened quite recently.

All of this leads to the LLM also incorrectly interpreting the question *Q* question and answering that a country can only have a single president at a time. The irrelevant context *X* and incorrect answer *Y* lead to the formation of a question *Q'* which does not have the same intent. The generation process is expected to rectify the answer, but since its dependencies (X and Y) are irrelevant and incorrect, com-

Stage I: Base QA

Question (Q) - Which character arrives late to the titular banquet in Plato's Symposium?

Context (X) – [{'name': 'Why is Socrates late in the Symposium?', 'snippet': "Plato's Symposium was written in the fourth century B.C.E. and offers a fictitious account of a gathering of notable Greek citizens at the home of Agathon. The famous philosopher Socrates is late to the banquet because he has fallen into a philosophical reverie in the doorway of a neighbouring home."},
{'name': "Who was present at Plato's Symposium?", 'snippet': 'It depicts a friendly contest of extemporaneous speeches given by a group of notable men
attending a banquet...}]

Initial Answer (Y) - Socrates arrives late to the banquet in Plato's Symposium due to...

Stage II: False Premise Detection

Inferred Question (Q') - Why does Socrates arrive late to the banquet in Plato's Symposium?

Q' not same Q in terms of the info that could be provided, proceed to stage III

Stage III: False Premise Repair

Explanation (Z) - In Plato's Symposium, it is mentioned that Socrates arrives late to the banquet because he has been lost in thought on a neighbouring porch. This aligns with the conclusion that Socrates arrives late to the banquet in the Symposium.

Ground Truth - Socrates

Figure 3.7: FPDAR True Premise with Abductive Explanation example and successful execution

bined with the fact that the model lacks the relevant parametric knowledge, the alternate explanation Z is also factually incorrect.

Stage I: Base QA

Question (Q) - How many presidents has Peru had in the last 5 years?

Context (X) – [{'name': 'How many years is a presidential term in Peru?', 'snippet': 'The president is directly elected for a five year term. Power is shared with a prime minister and a governing council of ministers, appointed by the president. The president can seek re-election, but not for consecutive terms. Unicameral congress of 120 members elected to represent geographical areas.'}
{'name': 'Does Peru have a President right now?', 'snippet': 'The first president was José de la Riva Agüero and the current president in office is Dina Boluarte, the first woman to hold the position.'...}]

Initial Answer (Y) - The current president of Peru is Dina Boluarte, a country can only have 1 president at a time...

Stage II: False Premise Detection

Inferred Question (Q') - Who is the current president of Peru and can Peru have multiple presidents?

Q' not same Q in terms of the info that could be provided, proceed to stage III

Stage III: False Premise Repair

Explanation (Z) - The conclusion that Peru has had only one president in the last 5 years is plausible because the current president, Dina Boluarte, took office in July 2021, and there is no indication of any other presidents during this time period and a country can only have one president active at a time.

Ground Truth - Five

Figure 3.8: FPDAR True Premise with alternative explanation and unsuccessful execution

4

Experiments and Results

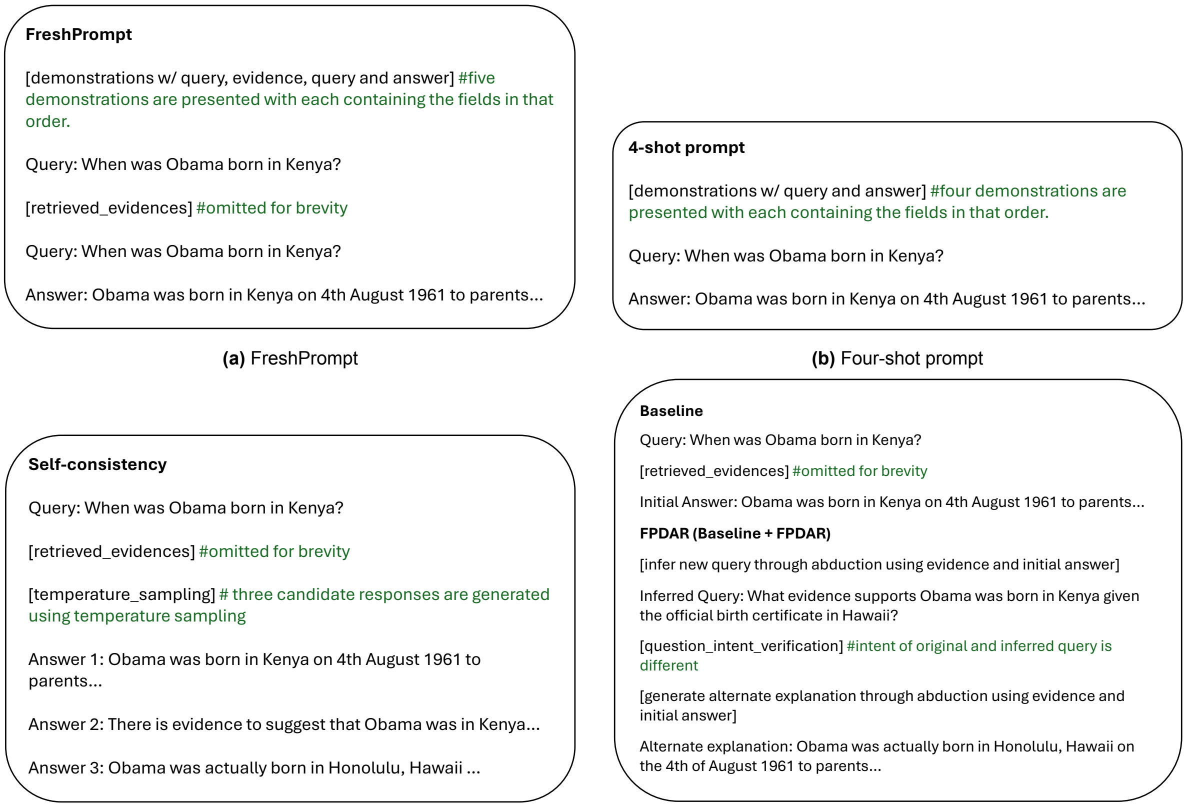
A series of experiments were conducted to evaluate the performance of FPDAR compared to other strategies for overcoming false premises in QA. This chapter details the experimental setup, the benchmarks, metrics, and methods used for comparison. The results will be discussed in terms of the specific improvements observed in a certain setting with a particular model. The next chapter will contextualize these results to better understand their broader implications.

4.1. Experiment

Benchmarks: The two benchmarks considered are *FreshQA* [124] and $(QA)^2$ [54]. These benchmarks were chosen because they were created to serve as an evaluation benchmark rather than a complete training dataset. Since our proposed reasoning approach operates at an inference level, these benchmarks are ideal for evaluation in our case.

- **FreshQA** contains 500 human-written open-ended questions that are dynamically changing. Out of the 500, 124 questions, i.e., approx 25% contain false premises, and the remaining are true premise questions. The temporal aspects are categorized based on how quickly the answer to certain questions might change depending on events in the real world. These categories are *fast-changing (Fast)*, *slow-changing (Slow)* and *never-changing (never)*. FreshQA also provides information regarding the knowledge cutoff by designating whether questions require knowledge of events that occurred before 2022 *($<$ 2022)* or since 2022 *($\geq$ 2022)*. Lastly, questions are also categorized according to the number of hops they require in terms of the various concepts discussed and their complexity as *(1-hop)* and *(m-hop)*. These categories are present in both false and true premise, but due to the smaller number of instances for each category, only the *($<$ 2022)* for the false premise is considered. Approximately 70% of the false premise questions are based on historical events and under the category *($<$ 2022)*. It is important to note that FreshQA is a dynamic benchmark; the answers to certain questions can change over time. Newer versions are released every week. Our experiments were performed on the March 18th, 2024 version.
- **$(QA)^2$** dataset consists of 570 questions that both expert annotators and crowd workers have annotated. They are equally distributed, i.e., 50% between questions with and without false premises. Unlike FreshQA, this dataset is static in nature in terms of updates to the dataset instances. The temporal aspect in $(QA)^2$ is restricted to questions containing a false premise and simply indicates whether this false assumption could change in the future. No further breakdown of questions, false premises, or temporal aspects are provided. However, these questions are derived from common search engine queries, which should aid in indicating the performance of different methods in real user queries.

Metric: The primary metric considered is the accuracy of the various methods and models across benchmarks since FreshQA and $(QA)^2$ provide ground truth answers for comparison. Both of these datasets do not provide further metrics related to false premises. A response to a question is considered correct if it matches the ground truth answer, i.e., response correctness evaluation. We also care about the precision of false premise detection and repair stages of FPDAR. Both the benchmarks have categorized their questions according to their premise; this also allows us to evaluate the performance of our method in identifying and mitigating false premises, essentially stages two and three.



(a) FreshPrompt **(b)** Four-shot prompt

(c) Self-consistency **(d)** FPDAR (Ours)

Figure 4.1: Example of inputs and outputs for different methods. **(a)** FreshPrompt: a few-shot approach with external knowledge [124], **(b)** 4-shot prompt: a few-shot approach without external knowledge [54], **(c)** Self-consistency: consistency based approach with three candidate responses using external knowledge and temperature sampling [127], **(d)** FPDAR (ours): a post-hoc false premise detection method augmented with external knowledge and using abduction to infer question based on original query and answer, verify their intent (detect false premise) and generate an alternate explanation using abduction if a false premise is detected.

Models: Three widely used LLMs are considered for evaluation, including *GPT 3.5-turbo* [13], *Llama2-70b* [118] and *Mistral-Small* [76]. GPT 3.5-turbo and Mistral Small are closed-sourced models, whereas Llama2-70b is open-source. This allows us to have a good balance between open and commercial models which enhances the robustness of our evaluation results. All three of these models are at par when it comes to their performance on various benchmarks for logical reasoning and language understanding[1]. This ensures that the models are on an equal footing and helps ensure that any observed results are more reflective of the method itself rather than the models' inherent capabilities. Details regarding the specific endpoints used to access these models are provided here B.2.1.

Methods to compare: The following methods were chosen for comparison because they are quite similar to FPDAR in their focus on tackling false premises and operating without training/fine-tuning. An overview of the chosen methods and the various forms of inputs and outputs they provide is shown in figure 4.1.

- FreshQA proposed a custom few-shot prompting technique called **FreshPrompt** [124], which incorporates five question-answer demonstrations at the beginning of the prompt, and each of these demonstrations has been augmented with two relevant web search results which have been sorted according to the current date.

[1] https://artificialanalysis.ai/

- $(QA)^2$ [54] does not propose its own method but evaluates various existing prompting strategies. In their evaluation, the best accuracy was generated by a **4-shot prompt** approach with question-answer demonstrations. Unlike the other methods, this 4-shot prompt approach is not augmented with web search results.
- **Self-consistency** [68] was chosen due to its remarkable performance on various hallucination and reasoning benchmarks [68, 127]. Moreover, self-consistency operates at an inference level, enabling a fair comparison since FPDAR also functions at inference. Our FPDAR method also seems most similar to the self-consistency (section 2.2.2) method in terms of its pseudo-detection nature. Instead of checking the consistency between responses, our method checks whether the intent between questions is consistent. Self-consistency does not have an explicit hallucination detection stage. The final response is chosen through majority voting with the assumption that since this response is consistent, it will likely be correct, which in turn indicates that the responses not chosen were hallucinated. Our implementation of self-consistency is augmented with web search results, as mentioned in section 3.2.1. Specific implementation details can be found here B.2.2.
- We categorize our first Base QA stage as the **Baseline** because it is one of the simplest strategies that could be adopted to handle false premises. It is essentially a zero-shot CoT prompting strategy with retrieval augmented generation.
- **FPDAR:** Our proposed false premise detection method based on abductive reasoning.

Evaluation Protocol: The evaluation is automated and achieved through *GPT4-turbo* [1]. To avoid any potential bias caused by the evaluation protocol (based on *GPT-4-Turbo*), the *GPT-4* series models are not used for any of the main experiments. A *RELAXED* evaluation approach is followed, which is solely focused on evaluating the correctness of the answer. Under the *RELAXED* evaluation, as long as the primary information in the LLM response is similar and consistent with the ground truth answer, the LLM response is considered correct. This is followed as long as the additional information in the LLM response does not contradict or change the perception of the primary answer. If there is a contradiction, then that response is marked as incorrect. For responses that include names of individuals or entities, widely recognized names or abbreviations are accepted. For numerical answers, exact figures are typically preferred unless the question specifically allows for approximations. Responses that may be grammatically incorrect or in a language other than English are acceptable. Responses that contain outdated or hallucinated information are accepted as long as this doesn't drastically alter the main answer. This *RELAXED* evaluation approach and the evaluation prompt are adopted from the *FreshQA* paper [124]. The evaluation prompt can be found here B.4.

4.2. Results

The accuracy performance of all models with different methods on the FreshQA and $(QA)^2$ benchmarks is reported. A stage-wise evaluation of our abductive reasoning processes for detection and repair is also performed. Additional analysis is done to understand the generalizability of FPDAR as a post-hoc method and the semantic similarity threshold sensitivity employed in the false premise detection stage. This is followed by an ablation study to understand the impact of various individual aspects of FPDAR.

4.2.1. Main Results

The accuracy results are shown in table 4.1. These are split into false and true premises for FreshQA and $(QA)^2$. FreshPrompt has consistent performance on $(QA)^2$ for all splits, but its performance varies on FreshQA, struggling largely on True Premise compared to other methods, leading to a reduction in its overall accuracy. The 4-shot prompt performs the worst on FreshQA, especially on True Premise. It is able to perform better on False Premise, but that is likely because a large number of instances in False Premise are historical. Given how the 4-shot prompt method does not incorporate any external knowledge but FreshQA being a temporal dataset essentially requiring up-to-date knowledge, this finding is unsurprising. This method primarily relies on the model's parametric knowledge to provide an answer that can work reasonably well for historical questions (in FreshQA, majorly consisting of false premise) but not for recent events (in FreshQA, majorly consisting of true premise). The static nature of $(QA)^2$ likely explains the reasonable performance observed on all splits when using the 4-shot prompt method. A fine-grained evaluation of temporal performance is presented in table B.1. Information

Methods	LLM	FreshQA			(QA)^2		
		All	FP	TP	All	FP	TP
FreshPrompt	GPT3.5-turbo	63.4	58.9	64.9	67.2	64.6	69.8
	LLama2 70B	35.2	36.3	34.8	60.5	60.4	60.7
	Mistral Small	57.8	61.3	56.6	64.0	64.2	63.9
4-shot prompt	GPT3.5-turbo	38.8	52.4	34.3	63.9	61.8	66.0
	LLama2 70B	31.2	55.6	23.1	43.0	53.3	32.6
	Mistral Small	41.8	62.1	35.1	63.3	70.5	56.1
Self-Consistency	GPT3.5-turbo	57.6	39.5	63.6	58.6	48.1	69.1
	LLama2 70B	62.6	50.8	66.5	50.9	36.1	65.6
	Mistral Small	56.6	57.3	56.4	47.5	41.8	53.3
Baseline	GPT3.5-turbo	64.2	53.2	67.8	74.7	65.6	<u>83.9</u>
	LLama2 70B	63.6	49.2	<u>68.4</u>	70.4	63.2	77.5
	Mistral Small	<u>67.4</u>	<u>65.3</u>	68.1	<u>77.2</u>	<u>73.7</u>	80.7
FPDAR	GPT3.5-turbo	64.0	54.8	67.0	75.4	67.7	**83.2**
	LLama2 70B	63.4	53.2	66.8	70.7	66.7	74.7
	Mistral Small	**68.2**	**70.2**	**67.6**	**79.1**	**77.2**	81.1
Performance Gain		+ 0.8	+ 4.9	- 0.8	+ 1.9	+ 3.5	- 0.7

Table 4.1: Performance comparison (accuracy in percent) of different methods on FreshQA and (QA)^2 benchmarks. Bold denotes the best performance of our approach. Underline denotes the best performance in other methods. The performance gain is calculated between the best performance of our approach and the best performance of the other methods on each split. 'FP' denotes False Premise, and 'TP' denotes True Premise based on the premise split of the respective benchmarks.

LLM	FreshQA			(QA)^2		
	All	FP	TP	All	FP	TP
GPT3.5-turbo	69.8	16.1	87.5	49.3	12.3	86.3
LLama2 70B	60.8	38.7	68.1	52.3	44.9	59.6
Mistral Small	67.2	20.2	82.7	53.0	18.2	87.7

Table 4.2: False premise detection performance (based on binary classification) of our approach *FPDAR* on FreshQA and (QA)^2 benchmarks. Question intent threshold at $\tau = 0.7$

regarding the sample size of each category is also presented in the fine-grained evaluation.

Self-consistency performs consistently well on both datasets when it concerns True Premise, but its performance drops drastically on False Premise. One possible reason is that the self-consistency method might be trying to generate answers that are coherent with the premise of the question, which, if they are false, could lead to an incorrect answer. Such responses might be consistent, but that does not necessarily imply that they are correct.

Interestingly, the best performance for all the comparison methods other than FPDAR was achieved through the Baseline method, the first stage of FPDAR, and essentially a zero-shot CoT approach with external knowledge. If we consider the Baseline as a standard approach, then FPDAR is applied on top of the Baseline as a posthoc method to detect and mitigate false premises using abduction. FPDAR largely succeeds with this endeavor, as seen in the performance gains in False Premises and overall accuracy. However, it does come at the cost of accuracy reduction in True Premise on both datasets. Since Mistral-Small achieved the highest performance across most of the splits, the subsequent analysis will primarily compare the effects seen on Mistral-Small.

4.2.2. Stage-Wise Evaluation

The evaluation of the abduction process in stage two, False Premise Detection, and stage three, False Premise Repair, is presented below.

LLM	FreshQA			$(QA)^2$		
	All	FP	TP	All	FP	TP
GPT3.5-turbo	53.4	76.6	45.7	53.7	62.8	44.6
LLama2 70B	30.4	90.3	10.6	51.9	91.9	11.9
Mistral Small	42.8	71.8	33.2	57.2	80.4	34.0

Table 4.3: False premise detection performance (based on binary classification) of our approach *FPDAR* on FreshQA and $(QA)^2$ benchmarks. Question intent threshold at $\tau = 0.9$

LLM	FreshQA			$(QA)^2$		
	Detected	FP	TP	Detected	FP	TP
GPT3.5-turbo	68.7	55.0	74.5	74.3	62.9	84.6
GPT3.5-turbo+	65.7	50.0	72.3	71.6	65.7	76.9
LLama2 70B	71.4	64.6	74.2	70.8	64.8	77.4
LLama2 70B+	70.2	66.7	71.7	70.8	68.8	73.0
Mistral Small	68.9	68.0	69.2	73.6	75.0	71.4
Mistral Small+	68.9	72.0	67.7	79.3	82.7	74.3

Table 4.4: False premise Repair performance (based on QA accuracy on tasks detected with false premises) of our approach on FreshQA and $(QA)^2$ benchmarks. LLM name and LLM name+ represent *Baseline* and *FPDAR* based on that LLM, respectively. 'FP' denotes False Premise, and 'TP' denotes True Premise based on the premise split of FPDAR. 'Detected' is the total of 'FP' and 'TP'.

False Premise Detection Evaluation

Based on table 4.2, it is clearly observed that our method FPDAR as a false premise detection cannot precisely identify false premise questions, and this is consistent on both datasets. This means that most of the actual false premise questions were incorrectly classified as true premise. Detection performance for True Premise is effectively the opposite of False Premise since, across both datasets, a significantly higher detection performance is achieved. When discussing detection performance, it is important to recall that our detection process also depends on the value of τ set in (cf. formula 3.3). The results in table 4.2 are obtained at a $\tau = 0.7$ threshold. If this threshold is increased to $\tau = 0.9$, then the detection performance seen in table 4.3 is acquired. At a glance, the detection of false premises has improved drastically, but this has come at the cost of decreased performance in true premise detection. The performance across different threshold values will be discussed in section 4.2.3.

False Premise Repair Evaluation

After the premise detection step, we analyze how our premise repair step affects final performance. The results are reported in Table 4.4. FP (False Premise) and TP (True Premise) denote the accuracy of those instances that were correctly detected according to their respective premise and were also evaluated to be correct in their answer. Detected is the combined performance of both FP and TP. The performance seen across both datasets and splits is promising since there is no major drop across the False or True Premise split. This signifies that our repair process can function sufficiently well, irrespective of the premise. Note that FP and TP refer to the False and True premises detected by our method, FPDAR. The FP and TP set seen in table 4.1 refers to the split from the respective dataset benchmarks.

4.2.3. Further Analysis

To enhance our understanding of generalizability and the impact of technical designs, we conduct a detailed analysis of our approach.

Sensitivity Analysis of Question Intent Similarity Threshold

The false premise detection heavily relies on the question intent similarity (cf. formula 3.3). To understand the effect of the semantic similarity threshold on our approach, a sensitivity plot for all backbone models is presented in figure 4.2, which showcases the total accuracy across both benchmarks. Observing the plot, it is clear that being too strict with the semantic similarity threshold could be detrimental to the overall accuracy, as this trend is seen in all models except Mistral. A clear threshold value is

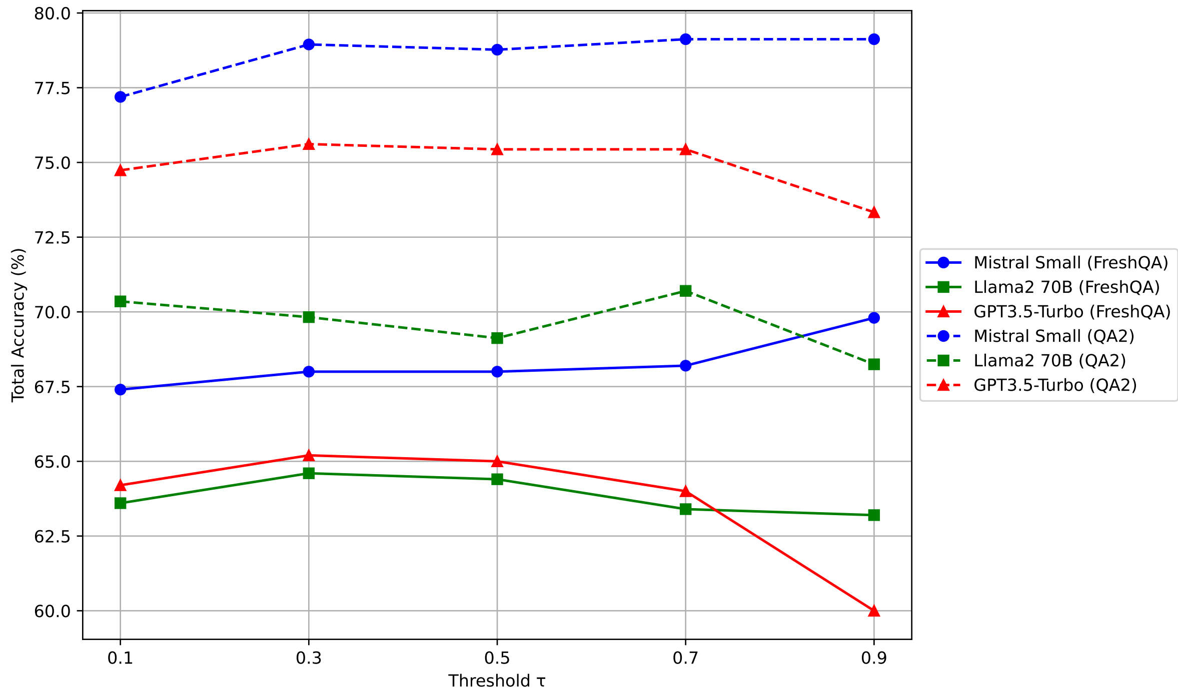


Figure 4.2: Semantic similarity threshold sensitivity plot for FreshQA and $(QA)^2$ showing the total accuracy varies depending on the threshold for the LLMs considered in our experiments

Method + LLM	FreshQA		
	All	**FP**	**TP**
FreshPrompt + GPT-3.5 turbo	66.4	58.1	69.1
Self-consistency + LLama2 70B	61.2	46.0	66.2

Table 4.5: Generalisation baselines accuracy on FreshPrompt and Self-consistency with FPDAR

difficult to select since no value provides consistently high accuracy for all models and benchmarks. If we look at figure 4.3, it presents a similar challenge in picking a threshold that maximizes the accuracy across all cases. In this study, our main goal was to detect false premises and also ensure that the accuracy (correctness) is maximized. Achieving a good balance between both these criteria is challenging as favoring one metric would cause the other to decrease. This can also be seen in the earlier false premise detection evaluation done with different thresholds. Calculating the average of all the accuracy values for every dataset and each model yields $\tau = 0.3$ as the threshold that maximizes the overall accuracy. Doing the same for the false premise split results in $\tau = 0.7$. The difference between each threshold is quite minor, as can be seen from the plots, but a preference was given to maximizing accuracy for false premises; therefore, $\tau = 0.7$ was chosen as the final threshold. All of our experiments with FPDAR are performed with this threshold.

Generalization Analysis across baselines

To verify the effectiveness of our approach as a post-hoc method over the QA methods, we substitute our baseline with other methods such as FreshPrompt and Self-consistency. We select the best-performing model variant for each of these methods and only perform the analysis on FreshQA due to time constraints. As seen in table 4.5, there is an improvement for the FreshPrompt variant on the overall and true premise split. However, the Self-consistency variant shows a decrease in accuracy for all splits. The overall accuracy for both variants is still comparable to their original method and, in some cases, even exceeds individual splits. This highlights that FPDAR could potentially be viable as a post-hoc solution.

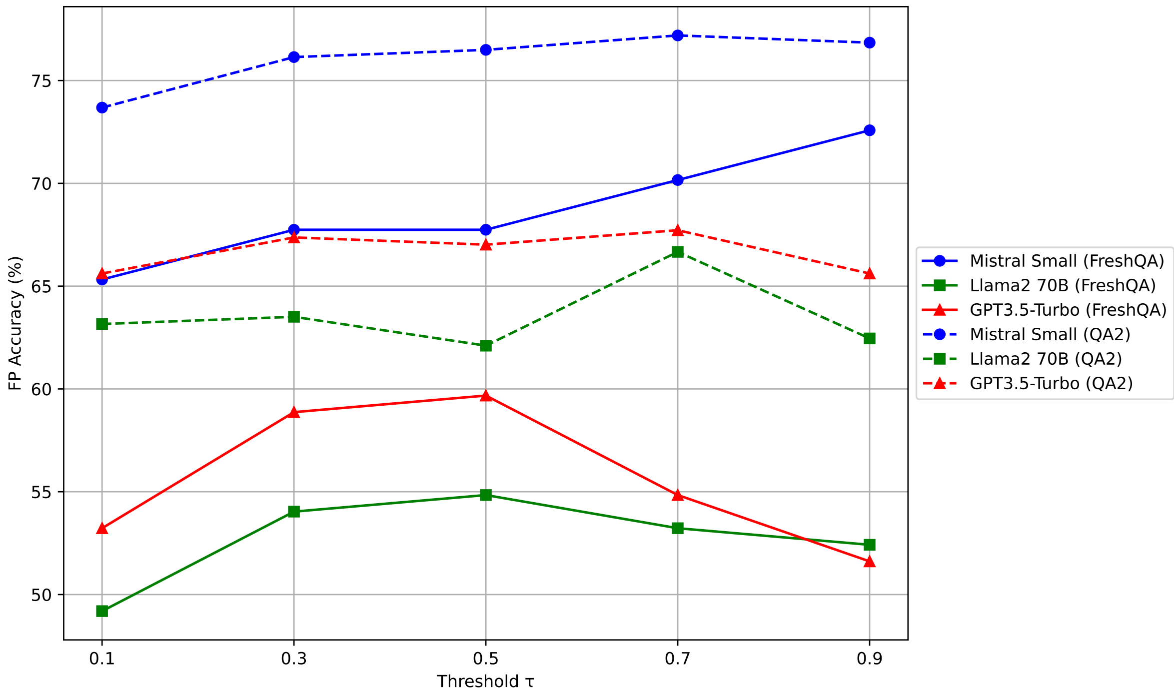


Figure 4.3: Semantic similarity threshold sensitivity plot for FreshQA and $(QA)^2$ showing the false premise accuracy varies depending on the threshold for the LLMs considered in our experiments

Methods	**All**	**FP**	**TP**
FPDAR	68.2	70.2	67.6
w/o context X			
- stage II	66.8	66.1	67.0
- stage III	64.6	66.9	63.8
- stage II & III	41.6	52.4	38.0
deductive reasoning	66.0	62.9	67.0
Stage III w/ Q' and X'	67.8	66.1	68.4
Extra input Q'	67.2	66.1	67.6
Extra input Q	68.0	70.2	67.3

Table 4.6: Ablation analysis on FreshQA dataset.

Ablation study

A number of ablation studies were done to understand the impact of various processes and stages of FPDAR.

- *w/o context* X. The variant removes the factual context in our approach. As X is used in both stage II and stage III of our approach, we explore all potential variants by removing them.
- *deductive reasoning*. In stage III, replace the framing of abductive reasoning in alternate explanation generation to deductive reasoning with the input of X and Y.
- *Stage III w/ Q' and X'*. The variant replaces the third stage of our methods by answering the generated question intent Q' and generating a new X' (using the same method as stage I).
- *Extra input Q'*. Include the generated question intent Q' as part of the input of stage III.
- *Extra input* Q. Include the original question Q as part of the input of stage III.

All variants are based on the Mistral-Small model and a semantic threshold $\tau = 0.7$, which achieved the best performance in our main experiment. A complete ablation evaluation for all other models is presented in table B.2.

From table 4.6, we can see that the external knowledge X is important in all stages of our approach. Removing X at stage II will result in a performance decrease due to failure to recognize potential false premises while removing it at stage III will significantly decrease the quality of the alternative explanation. After reframing the abductive reasoning prompt used for false premise repair with deductive reasoning, there is a performance drop (7.3%) on the split of False Premise. This verifies the effectiveness of utilizing abductive reasoning to address the impact of false premise. Meanwhile, we also found that directly answering Q' will result in somewhat comparable results to the original variant when considering the false premise split and manages to exceed the true premise. This indicates that our motivation for checking false premises on the basis of reaching the original premise (original question Q) is correct in most cases. This finding is also strengthened by the results seen for the inclusion of Q' and even Q while generating the alternative explanation using abduction. While the accuracy does not increase beyond the original FPDAR variant, it is comparable in almost every split with minor reductions.

5
Discussion and Conclusion

The chapter will explain the main takeaways from the previous results section. This will be followed by the limitations section, where certain assumptions that could influence the findings will be discussed. Finally, we will conclude by discussing potential future research directions.

5.1. Key Findings

FPDAR improves over other methods in overall and false premise accuracy

FPDAR demonstrates improvements in overall accuracy compared to other methods such as FreshPrompt [124], 4-shot prompt [54], Self-Consistency [127], and the Baseline. It consistently surpasses FreshPrompt and the 4-shot prompt on their respective datasets, FreshQA and $(QA)^2$, across all LLMs with respect to overall accuracy. A key factor in this success is the significant improvements in the false premise split for both the FreshQA and $(QA)^2$ benchmarks. However, there is a decrease in accuracy for the true premise split on both benchmarks, indicating that FPDAR is more effective for false premises and may not perform as well with valid premises.

FPDAR struggles to detect false premises without sacrificing accuracy

In the stage-wise evaluation, at threshold $\tau = 0.7$, FPDAR exhibits inadequate performance in detecting false premises. While our approach is termed a 'pseudo-detection' method due to the absence of explicit detection mechanisms, one of its objectives remains the identification of any false premises within questions. However, FPDAR struggles significantly in this regard. Improving detection performance can be achieved by enforcing a stricter similarity threshold, such as $\tau = 0.9$. However, this adjustment comes at the expense of reduced overall accuracy, which is undesirable. If false premise detection stands as the primary concern for a system, FPDAR could still be employed effectively by raising the threshold, albeit with the acknowledged trade-off of accuracy.

Backward direction and abductive reasoning could be a viable reasoning methodology for LLM

Our research question defined in 1.3 was to investigate whether abductive reasoning can be used to detect false premises and hallucinations in a QA scenario. The development of FPDAR and the accuracy results in table 4.1 across all LLMs being considered shows that a backward reasoning methodology grounded in abductive reasoning could be a viable strategy for instructing LLMs to reason effectively. While the overall accuracy and false premise accuracy have increased, there has also been a decrease in accuracy for true premises. The performance is still comparable but does not completely align with the findings and notions presented in [134, 52] regarding abductive reasoning being superior to other forms of approaches, such as forward reasoning. While LLM's reasoning capabilities could be more effectively utilized in a backward reasoning methodology for false premises, this cannot be confidently stated for true premises. Further experiments and evaluations using different QA datasets are necessary before drawing any definitive conclusions. Additionally, the potential transferability of FPDAR to other reasoning tasks in QA needs to be explored. If it turns out that FPDAR cannot be transferred, this may highlight some limitations of our methodology that are not yet apparent.

5.2. Limitations

This section will discuss certain limitations concerning the overall study, findings, the implementation of FPDAR, and the experiment setup. Since these limitations could influence the main takeaways of the findings, we categorize each discussion point explained in the previous according to how significant the underlying assumptions are in order for the findings to be valid. A three-level categorization with labels **"weak"**, **"moderate"** and **"strong"** is chosen. A "weak" assumption signifies that the underlying premise for that specific discussion point can safely be assumed to nearly always hold and be valid. A "moderate" assumption would indicate that that specific finding might not be true under certain conditions. A "strong" assumption is detrimental to the finding it is assigned to because it signifies there are strong counterarguments to be made, which can considerably weaken that specific finding. The assumptions listed below can be associated with multiple findings since certain points can overlap and coincide.

Reliance on LLM emergent abilities and reasoning might prevent scaling to smaller models

The first finding discusses how FPDAR, which at its core is a reasoning approach that primarily relies on the LLM's reasoning capabilities, achieves superior performance compared to other methods. However, the implementation of reasoning is achieved through specific capabilities of LLMs, which come with certain caveats. The entire abduction process and the RAG implementation rely heavily on the LLM's in-context learning abilities [130]. These are known as emergent abilities and emerge as the model scales up. Larger models demonstrate greater effectiveness at a different rate compared to smaller models. Our abduction process expects the LLM to reason over various responses and information to generate an explanation. This would indicate that the effectiveness of FPDAR would deteriorate drastically when applied to smaller, less capable models. Tests to verify this were not conducted due to time constraints, but based on previous research [130, 21], it can be safe to assume that the overall performance would suffer and FPDAR would likely not function appropriately for smaller models. Related to LLM abilities is also the notion that these models do not genuinely comprehend or reason because they operate purely on statistical patterns in data rather than true understanding. LLMs generate text by predicting the most likely sequence of words based on their training data without any conscious awareness or cognitive processes. While they can emulate reasoning by replicating patterns that appear logical, they lack the ability to grasp context, meaning, or intent in a human-like manner. Their "intelligence" is an illusion created by advanced algorithms, not actual comprehension. Since our entire work relies on reasoning achieved through LLMs, this raises questions about the reliability of our accuracy benchmarks and the motivation of utilizing backward reasoning or LLM reasoning in general. We categorize this assumption as **"strong"**. This is a hotly debated topic, and if the reader is interested, more explanation is provided in appendix A.

What if the context X is irrelevant and inaccurate?

In our overall implementation, there is a huge emphasis on the importance of relevant, up-to-date information, which in our experiments is the scraped web search results X. It is discussed how necessary context X is to provide sufficient context to the reasoning processes. However, this also assumes that the context X retrieved is, in fact, relevant and accurate. In our implementation of FPDAR, there is no process to determine the veracity of the web search results. We have also not performed any experiments to assess the relevance of the web search results. Since we use question Q verbatim to query the web search API to generate results, we assume that those results will be relevant and the chances of receiving irrelevant results are low. This strategy has also been used in other works involving RAG, such as [124, 60]. However, web search engines have been known to struggle with natural language queries [12]. Moreover, there is no guarantee that the web results retrieved will be relevant. Since our implementation depends on the veracity of these results, this is an assumption that should be highlighted. We categorize this assumption as **"weak"**.

LLM as a judge; concerns surrounding bias

Our evaluation protocol is automated as described in section 4.1. This allows us to efficiently scale the evaluation protocol to various models and experiment variations. The alternative to this would have been human evaluation, but that would have cost more time and required additional compensation. However, utilizing LLMs for evaluation is not without its downsides. First of all, there is the possibility of models exhibiting bias. This could be in position bias, where models tend to favor responses in a specific position over others simply due to their position. For example, in [147], the authors found that while comparing two responses with GPT4, the model tends to favor the first response over the second response from a different model. They swapped the responses, and now GPT4's judgment would switch, and it would pick the first response again, contradicting its earlier decisions. Another possible bias could be verbosity bias, where models tend to select responses that are long and verbose even if they are not clear or accurate compared to shorted alternatives. LLMs are also known to favor responses generated by themselves or other models [147]. These biases can reduce the reliability of LLM evaluation judgments and inadvertently skew the results. However, even with these limitations, automated evaluations are used and often accompanied by a human evaluation of a small subset of the data [124, 88]. An inter-annotator agreement between the annotator and the LLM responses is then calculated to gauge the reliability of the model's judgment. If the agreement between the annotators and the model responses is high, then the model response is assumed to be reliable. Due to time and cost constraints, a human evaluation was not conducted for this thesis. Since there is a possibility that a human evaluation might change the trends seen in the benchmarks, we categorize this assumption as **"moderate"**.

5.3. Conclusion

In this research, we introduced a novel method, FPDAR, for detecting false premises in question answering using Large Language Models (LLMs) through abductive reasoning. The main concept is to use abductive reasoning to identify potential false premises and then rectify them by creating an alternative explanation as the final response. We first use abductive reasoning to understand the question's intent, which helps us detect false premises by comparing their semantic similarity with the original question. After identifying potential hallucinated responses and gathering factual context, we employ abductive reasoning again to diagnose these false premises and improve the response quality. Our method has shown promise in generating accurate, coherent responses by performing abductive reasoning, but it struggles significantly in accurately detecting false premises in questions. The detection performance can be improved, but it comes at the cost of a reduction in correctness. Moreover, our method FPDAR performs well on false premises but is slightly worse, albeit still comparable on true premises, signaling that reasoning through abduction is not foolproof and can sometimes lead to incorrect conclusions.

To enhance our approach and ensure its applicability across various scenarios, it would be beneficial to test FPDAR on additional false premise benchmarks. FPDAR mainly depends on abductive reasoning and the inherent abilities of the large language model (LLM). We hypothesize that FPDAR may struggle with models that have fewer parameters. However, evaluating our method across a wide range of models will provide insights into its overall usability. While larger models are less likely to benefit significantly, smaller models might experience a substantial improvement, assuming they have reasonable in-context learning capabilities. Experiments regarding the choice of fields to be used as external knowledge should be performed along with the number of results that should be incorporated; this analysis was missing due to time constraints but could provide valuable insights to improve overall performance. Given its satisfactory performance on valid premise questions in generating accurate responses, FPDAR could potentially be applied to other QA tasks. To further assess its effectiveness, FPDAR could be evaluated on natural language understanding benchmarks like MMLU [36], possibly incorporating more pertinent reasoning methods such as Chain of Thought (CoT) reasoning.

5.4. Ethics Statement

In this thesis, our work and experiments have primarily involved Large Language Models, which have a tendency to provide false information and hallucinate. These models can also output biased and malicious information. This can happen through tweaking certain parameters that control the output or through adversarial methods designed to bypass the guardrails of models. During various experiments, the researchers involved with this study did not observe any such instances, but we have not performed any experiments to verify the same. Since our work has centered around identifying and mitigating false premises, we are not completely certain whether our method could be used for malicious intents. As always, the end user should always be careful and take into consideration various conditions before accepting any AI-generated output for use in practical systems. If any malicious or biased output is generated by a model using the method provided in this study, then this was not the intention of the author or the researchers associated with this study.

References

[1] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[2] Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. “Do Language Models Know When They’re Hallucinating References?” In: *arXiv preprint arXiv:2305.18248* (2023).

[3] Mostafa M Amin, Erik Cambria, and Björn W Schuller. “Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of chatgpt”. In: *IEEE Intelligent Systems* 38.2 (2023), pp. 15–23.

[4] Henning Andersen. “Abductive and deductive change”. In: *Language* (1973), pp. 765–793.

[5] Yejin Bang et al. “A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity”. In: *arXiv preprint arXiv:2302.04023* (2023).

[6] Emily M. Bender and Alexander Koller. “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 5185–5198. DOI: `10.18653/v1/2020.acl-main.463`. URL: `https://aclanthology.org/2020.acl-main.463`.

[7] Chandra Bhagavatula et al. “Abductive commonsense reasoning”. In: *arXiv preprint arXiv:1908.05739* (2019).

[8] Prajjwal Bhargava and Vincent Ng. “Commonsense knowledge reasoning and generation with pre-trained language models: A survey”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 11. 2022, pp. 12317–12325.

[9] Ning Bian et al. “Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models”. In: *arXiv preprint arXiv:2303.16421* (2023).

[10] Meng Cao et al. “Factual error correction for abstractive summarization models”. In: *arXiv preprint arXiv:2010.08712* (2020).

[11] Yihan Cao et al. “A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt”. In: *arXiv preprint arXiv:2303.04226* (2023).

[12] Yi Chang and Hongbo Deng. *Query understanding for search engines*. Springer, 2020.

[13] OpenAI ChatGPT. URL: `https://platform.openai.com/docs/models/gpt-3-5-turbo`.

[14] Wenhu Chen, Xinyi Wang, and William Yang Wang. “A Dataset for Answering Time-Sensitive Questions”. In: *Relation* (1758).

[15] Peter Clark, Oyvind Tafjord, and Kyle Richardson. “Transformers as soft reasoners over language”. In: *arXiv preprint arXiv:2002.05867* (2020).

[16] Allan Collins, John Seely Brown, and Kathy M Larkin. “Inference in text understanding”. In: *Theoretical issues in reading comprehension*. Routledge, 2017, pp. 385–408.

[17] Andrew Cropper et al. “Inductive logic programming at 30”. In: *Machine Learning* 111.1 (2022), pp. 147–172.

[18] Jiaxi Cui et al. “Chatlaw: Open-source legal large language model with integrated external knowledge bases”. In: *arXiv preprint arXiv:2306.16092* (2023).

[19] Joe Davison, Joshua Feldman, and Alexander Rush. “Commonsense Knowledge Mining from Pretrained Models”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1173–1178. DOI: `10.18653/v1/D19-1109`. URL: `https://aclanthology.org/D19-1109`.

[20] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[21] Qingxiu Dong et al. “A survey on in-context learning”. In: *arXiv preprint arXiv:2301.00234* (2022).

[22] Hartry Field. “What is logical validity”. In: *Foundations of logical consequence* (2015), pp. 33–70.

[23] Joseph Fields. *A Gentle Introduction to the Art of Mathematics; Version 2.0.* http://www. southernct. edu, 2010.

[24] James Fieser and Bradley Dowden. “Internet encyclopedia of philosophy”. In: (2011).

[25] Peter A Flach and Antonis C Kakas. “Abductive and inductive reasoning: background and issues”. In: *Abduction and induction: Essays on their relation and integration* (2000), pp. 1–27.

[26] Yao Fu, Hao Peng, and Tushar Khot. “How does gpt obtain its ability? tracing emergent abilities of language models to their sources”. In: *Yao Fu’s Notion* (2022).

[27] Boris A Galitsky. “LLM-Based Personalized Recommendations in Health”. In: (2024).

[28] Jianfeng Gao and Chin-Yew Lin. *Introduction to the special issue on statistical language modeling*. 2004.

[29] Andrew S. Gordon and Jerry R. Hobbs. “Explanation”. In: *A Formal Theory of Commonsense Psychology: How People Think People Think*. Cambridge University Press, 2017, pp. 299–305.

[30] Zhibin Gou et al. “CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing”. In: *Second Agent Learning in Open-Endedness Workshop*. 2023.

[31] Arnav Gudibande et al. “The false promise of imitating proprietary llms”. In: *arXiv preprint arXiv:2305.15717* (2023).

[32] Chuan Guo et al. “On calibration of modern neural networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1321–1330.

[33] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. “A survey on automated fact-checking”. In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 178–206.

[34] Joseph Y Halpern. “Reasoning about knowledge: An overview”. In: *Theoretical aspects of reasoning about knowledge*. Elsevier. 1986, pp. 1–17.

[35] Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. “The faculty of language: what is it, who has it, and how did it evolve?” In: *science* 298.5598 (2002), pp. 1569–1579.

[36] Dan Hendrycks et al. “Measuring massive multitask language understanding”. In: *arXiv preprint arXiv:2009.03300* (2020).

[37] Jerry R Hobbs. “Abduction in natural language understanding”. In: *The handbook of pragmatics* (2006), pp. 724–741.

[38] Ari Holtzman et al. “The Curious Case of Neural Text Degeneration”. In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=rygGQyrFvH`.

[39] Jiaxin Huang et al. “Large language models can self-improve”. In: *arXiv preprint arXiv:2210.11610* (2022).

[40] Jie Huang and Kevin Chen-Chuan Chang. “Towards reasoning in large language models: A survey”. In: *arXiv preprint arXiv:2212.10403* (2022).

[41] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. “Are Large Pre-Trained Language Models Leaking Your Personal Information?” In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2038–2047. DOI: `10.18653/v1/2022.findings-emnlp.148`. URL: `https://aclanthology.org/2022.findings-emnlp.148`.

[42] Lei Huang et al. “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions”. In: *arXiv preprint arXiv:2311.05232* (2023).

[43] Shaohan Huang et al. “Language is not all you need: Aligning perception with language models”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[44] Shima Imani, Liang Du, and Harsh Shrivastava. “Mathprompter: Mathematical reasoning using large language models”. In: *arXiv preprint arXiv:2303.05398* (2023).

[45] Pauline Jacobson. “THE SHORT ANSWER: IMPLICATIONS FOR DIRECT COMPOSITIONALITY (AND VICE VERSA)”. In: *Language* 92.2 (2016), pp. 331–375. ISSN: 00978507, 15350665. URL: `http://www.jstor.org/stable/44164093` (visited on 05/25/2024).

[46] Ziwei Ji et al. “Survey of Hallucination in Natural Language Generation”. In: *ACM Comput. Surv.* 55.12 (2023). ISSN: 0360-0300. DOI: `10.1145/3571730`. URL: `https://doi.org/10.1145/3571730`.

[47] Zhengbao Jiang et al. “How can we know what language models know?” In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 423–438.

[48] Phil Johnson-Laird. “Deductive reasoning”. In: *Wiley Interdisciplinary Reviews: Cognitive Science* 1.1 (2010), pp. 8–17.

[49] Subbarao Kambhampati. “Can large language models reason and plan?” In: *Annals of the New York Academy of Sciences* 1534.1 (2024), pp. 15–18.

[50] Jared Kaplan et al. “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361* (2020).

[51] Jungo Kasai et al. “RealTime QA: What’s the Answer Right Now?” In: *arXiv preprint arXiv:2207.13332* (2022).

[52] Mehran Kazemi et al. “Lambada: Backward chaining for automated reasoning in natural language”. In: *arXiv preprint arXiv:2212.13894* (2022).

[53] Zachary Kenton et al. “Alignment of language agents”. In: *arXiv preprint arXiv:2103.14659* (2021).

[54] Najoung Kim et al. “$(QA)^2$: Question Answering with Questionable Assumptions”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 8466–8487. DOI: `10.18653/v1/2023.acl-long.472`. URL: `https://aclanthology.org/2023.acl-long.472`.

[55] Takeshi Kojima et al. “Large language models are zero-shot reasoners”. In: *Advances in neural information processing systems* 35 (2022), pp. 22199–22213.

[56] Po-Nien Kung and Nanyun Peng. “Do models really learn to follow instructions? an empirical study of instruction tuning”. In: *arXiv preprint arXiv:2305.11383* (2023).

[57] Angeliki Lazaridou et al. “Internet-augmented language models through few-shot prompting for open-domain question answering”. In: (2022).

[58] Mike Lewis et al. “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. In: *arXiv preprint arXiv:1910.13461* (2019).

[59] Junyi Li et al. “Halueval: A large-scale hallucination evaluation benchmark for large language models”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 6449–6464.

[60] Junyi Li et al. “The dawn after the dark: An empirical study on factuality hallucination in large language models”. In: *arXiv preprint arXiv:2401.03205* (2024).

[61] Yinheng Li et al. “Large language models in finance: A survey”. In: *Proceedings of the fourth ACM international conference on AI in finance*. 2023, pp. 374–382.

[62] Stephanie Lin, Jacob Hilton, and Owain Evans. “Truthfulqa: Measuring how models mimic human falsehoods”. In: *arXiv preprint arXiv:2109.07958* (2021).

[63] Adam Liska et al. “Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 13604–13622.

[64] Xiaoyong Liu and W Bruce Croft. "Statistical language modeling for information retrieval." In: *Annu. Rev. Inf. Sci. Technol.* 39.1 (2005), pp. 1–31.

[65] Jieyi Long. "Large language model guided tree-of-thought". In: *arXiv preprint arXiv:2305.08291* (2023).

[66] Junliang Luo et al. "Hallucination Detection and Hallucination Mitigation: An Investigation". In: *arXiv preprint arXiv:2401.08358* (2024).

[67] Kyle Mahowald et al. "Dissociating language and thought in large language models: a cognitive perspective". In: *arXiv preprint arXiv:2301.06627* (2023).

[68] Potsawee Manakul, Adian Liusie, and Mark Gales. "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

[69] Ben Mann et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[70] Conor McHugh and Jonathan Way. "What is reasoning?" In: *Mind* 127.505 (2018), pp. 167–196.

[71] Jacob Menick et al. "Teaching language models to support answers with verified quotes". In: *arXiv preprint arXiv:2203.11147* (2022).

[72] Hugo Mercier and Dan Sperber. *The enigma of reason*. Harvard University Press, 2017.

[73] Grégoire Mialon et al. "Augmented language models: a survey". In: *arXiv preprint arXiv:2302.07842* (2023).

[74] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[75] Pasquale Minervini et al. *awesome-hallucination-detection*. `https://github.com/EdinburghNLP/awesome-hallucination-detection`. 2014.

[76] Mistral AI Mistral AI. *Models: Mistral AI large language models*. URL: `https://docs.mistral.ai/getting-started/models/`.

[77] Dor Muhlgay et al. "Generating benchmarks for factuality evaluation of language models". In: *arXiv preprint arXiv:2307.06908* (2023).

[78] Niels Mündler et al. "Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation". In: *arXiv preprint arXiv:2305.15852* (2023).

[79] Mark A Musen and Johan Van der Lei. "Of brittleness and bottlenecks: Challenges in the creation of pattern-recognition and expert-system models". In: *Machine intelligence and pattern recognition*. Vol. 7. Elsevier, 1988, pp. 335–352.

[80] Reiichiro Nakano et al. "Webgpt: Browser-assisted question-answering with human feedback". In: *arXiv preprint arXiv:2112.09332* (2021).

[81] John D Norton. "A little survey of induction". In: (2003).

[82] Pruthvi Patel et al. "Is a Question Decomposition Unit All We Need?" In: *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. 2022.

[83] Gabriele Paul. "Approaches to abductive reasoning: an overview". In: *Artificial intelligence review* 7.2 (1993), pp. 109–152.

[84] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. 1st. USA: Basic Books, Inc., 2018. ISBN: 046509760X.

[85] Charles Sanders Peirce. *Collected papers of charles sanders peirce*. Vol. 5. Harvard University Press, 1974.

[86] Charles Sanders Peirce. *Reasoning and the logic of things: The Cambridge conferences lectures of 1898*. Harvard University Press, 1992.

[87] Baolin Peng et al. "Check your facts and try again: Improving large language models with external knowledge and automated feedback". In: *arXiv preprint arXiv:2302.12813* (2023).

[88] Baolin Peng et al. "Instruction tuning with gpt-4". In: *arXiv preprint arXiv:2304.03277* (2023).

[89] F Petroni et al. “KILT: a Benchmark for Knowledge Intensive Language Tasks.” In: *NAACL-HLT*. Association for Computational Linguistics. 2021, pp. 2523–2544.

[90] Steven T Piantadosi and Felix Hill. “Meaning without reference in large language models”. In: *arXiv preprint arXiv:2208.02957* (2022).

[91] Shuofei Qiao et al. “Reasoning with language model prompting: A survey”. In: *arXiv preprint arXiv:2212.09597* (2022).

[92] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[93] Jack W Rae et al. “Scaling language models: Methods, analysis & insights from training gopher”. In: *arXiv preprint arXiv:2112.11446* (2021).

[94] Yasaman Razeghi et al. “Impact of pretraining term frequencies on few-shot reasoning”. In: *arXiv preprint arXiv:2202.07206* (2022).

[95] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: `https://arxiv.org/abs/1908.10084`.

[96] Ruiyang Ren et al. “Investigating the factual knowledge boundary of large language models with retrieval augmentation”. In: *arXiv preprint arXiv:2307.11019* (2023).

[97] Adam Roberts, Colin Raffel, and Noam Shazeer. “How Much Knowledge Can You Pack Into the Parameters of a Language Model?” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 5418–5426.

[98] Ronald Rosenfeld. “Two decades of statistical language modeling: Where do we go from here?” In: *Proceedings of the IEEE* 88.8 (2000), pp. 1270–1278.

[99] Pranab Sahoo et al. “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications”. In: *arXiv preprint arXiv:2402.07927* (2024).

[100] Merrilee H Salmon. “Introduction to logic and critical thinking”. In: (2013).

[101] Victor Sanh et al. “Multitask prompted training enables zero-shot task generalization”. In: *arXiv preprint arXiv:2110.08207* (2021).

[102] Justyna Sarzynska-Wawer et al. “Detecting formal thought disorder by deep contextualized word representations”. In: *Psychiatry Research* 304 (2021), p. 114135.

[103] Joshua Schechter. “Deductive Reasoning”. In: *The Encyclopedia of the Mind*. Ed. by Hal Pashler. SAGE Reference, 2013.

[104] John Schulman. *John Schulman - reinforcement learning from human feedback: Progress and challenges*. 2023. URL: `https://www.youtube.com/watch?v=hhiLw5Q_UFg&t=1s`.

[105] John Schulman. “Reinforcement learning from human feedback: Progress and challenges”. In: *Berkley Electrical Engineering and Computer Sciences. URL: https://eecs. berkeley. edu/research/colloquium/230419 [accessed 2023-11-15]*. 2023.

[106] Mrinank Sharma et al. “Towards understanding sycophancy in language models”. In: *arXiv preprint arXiv:2310.13548* (2023).

[107] Emily Sheng et al. “The woman worked as a babysitter: On biases in language generation”. In: *arXiv preprint arXiv:1909.01326* (2019).

[108] Freda Shi et al. “Large language models can be easily distracted by irrelevant context”. In: *Proceedings of the 40th International Conference on Machine Learning*. 2023, pp. 31210–31227.

[109] Weijia Shi et al. “Replug: Retrieval-augmented black-box language models”. In: *arXiv preprint arXiv:2301.12652* (2023).

[110] Weijia Shi et al. “Trusting your evidence: Hallucinate less with context-aware decoding”. In: *arXiv preprint arXiv:2305.14739* (2023).

[111] Xiaoming Shi et al. “Language models can improve event prediction by few-shot abductive reasoning”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[112] Chenglei Si et al. "Prompting gpt-3 to be reliable". In: *arXiv preprint arXiv:2210.09150* (2022).

[113] Alon Talmor et al. "Commonsenseqa: A question answering challenge targeting commonsense knowledge". In: *arXiv preprint arXiv:1811.00937* (2018).

[114] Alon Talmor et al. "Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20227–20237.

[115] Ana Clara Teixeira et al. "Enhancing Credit Risk Reports Generation using LLMs: An Integration of Bayesian Networks and Labeled Guide Prompting". In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. 2023, pp. 340–348.

[116] Scott M Thede and Mary Harper. "A second-order hidden Markov model for part-of-speech tagging". In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*. 1999, pp. 175–182.

[117] Arun James Thirunavukarasu et al. "Large language models in medicine". In: *Nature medicine* 29.8 (2023), pp. 1930–1940.

[118] Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[119] Alan M Turing. *Computing machinery and intelligence*. Springer, 2009.

[120] Karthik Valmeekam et al. "Large language models still can't plan (a benchmark for llms on planning and reasoning about change)". In: *arXiv preprint arXiv:2206.10498* (2022).

[121] Neeraj Varshney et al. "A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation". In: *arXiv preprint arXiv:2307.03987* (2023).

[122] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[123] Juraj Vladika and Florian Matthes. "Scientific Fact-Checking: A Survey of Resources and Approaches". In: *Findings of the Association for Computational Linguistics: ACL 2023*. 2023, pp. 6215–6230.

[124] Tu Vu et al. "Freshllms: Refreshing large language models with search engine augmentation". In: *arXiv preprint arXiv:2310.03214* (2023).

[125] Thomas Wang et al. "What language model architecture and pretraining objective works best for zero-shot generalization?" In: *International Conference on Machine Learning*. PMLR. 2022, pp. 22964–22984.

[126] William Yang Wang. "" liar, liar pants on fire": A new benchmark dataset for fake news detection". In: *arXiv preprint arXiv:1705.00648* (2017).

[127] Xuezhi Wang et al. "Self-consistency improves chain of thought reasoning in language models". In: *arXiv preprint arXiv:2203.11171* (2022).

[128] Zhiguo Wang, Wael Hamza, and Radu Florian. "Bilateral multi-perspective matching for natural language sentences". In: *arXiv preprint arXiv:1702.03814* (2017).

[129] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.

[130] Jason Wei et al. "Emergent abilities of large language models". In: *arXiv preprint arXiv:2206.07682* (2022).

[131] Tsung-Han Wu et al. "See, say, and segment: Teaching lmms to overcome false premises". In: *arXiv preprint arXiv:2312.08366* (2023).

[132] Jian Xie et al. "Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts". In: *The Twelfth International Conference on Learning Representations*. 2023.

[133] Miao Xiong et al. "Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs". In: *The Twelfth International Conference on Learning Representations*. 2023.

[134] Fangzhi Xu et al. “Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views”. In: *arXiv preprint arXiv:2306.09841* (2023).

[135] Zonglin Yang et al. “Language models as inductive reasoners”. In: *arXiv preprint arXiv:2212.10923* (2022).

[136] Zonglin Yang et al. “Logical reasoning over natural language as knowledge representation: A survey”. In: *arXiv preprint arXiv:2303.12023* (2023).

[137] Fei Yu, Hongbo Zhang, and Benyou Wang. “Nature language reasoning, a survey”. In: *arXiv preprint arXiv:2303.14725* (2023).

[138] Xinyan Yu et al. “CREPE: Open-Domain Question Answering with False Presuppositions”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 10457–10480. DOI: 10.18653/v1/2023.acl-long.583. URL: https://aclanthology.org/2023.acl-long.583.

[139] Hongbang Yuan et al. “Whispers that Shake Foundations: Analyzing and Mitigating False Premise Hallucinations in Large Language Models”. In: *arXiv preprint arXiv:2402.19103* (2024).

[140] Eric Zelikman et al. “Star: Bootstrapping reasoning with reasoning”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15476–15488.

[141] Michael Zhang and Eunsol Choi. “SituatedQA: Incorporating Extra-Linguistic Contexts into QA”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 7371–7387.

[142] Muru Zhang et al. “How language model hallucinations can snowball”. In: *arXiv preprint arXiv:2305.13534* (2023).

[143] Shengyu Zhang et al. “Instruction tuning for large language models: A survey”. In: *arXiv preprint arXiv:2308.10792* (2023).

[144] Yue Zhang et al. “Siren’s song in the AI ocean: a survey on hallucination in large language models”. In: *arXiv preprint arXiv:2309.01219* (2023).

[145] Wayne Xin Zhao et al. “A survey of large language models”. In: *arXiv preprint arXiv:2303.18223* (2023).

[146] Wenting Zhao et al. “Abductive commonsense reasoning exploiting mutually exclusive explanations”. In: *arXiv preprint arXiv:2305.14618* (2023).

[147] Lianmin Zheng et al. “Judging llm-as-a-judge with mt-bench and chatbot arena”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[148] Wenxuan Zhou et al. “Context-faithful prompting for large language models”. In: *arXiv preprint arXiv:2303.11315* (2023).

[149] Daniel M Ziegler et al. “Fine-tuning language models from human preferences”. In: *arXiv preprint arXiv:1909.08593* (2019).

A

Do LLMs Truly Understand?

LLMs have shown tremendous performance on various downstream tasks that involve reasoning and thinking. Some recent studies suggest that LLMs may exhibit certain aspects of reasoning, such as the ability to follow step-by-step prompts [129] and reflect human-like content effects on reasoning. However, these findings don't definitively prove that LLMs can genuinely reason as demonstrated in [67]. For instance, it's unclear whether the models rely on true reasoning or simply follow heuristic rules for prediction. Even though LLMs might appear to reason sequentially, the outcomes may be incorrect or inconsistent, raising questions about whether they truly reason or merely produce reasoning-like responses. Additionally, while LLMs may demonstrate some human-like reasoning behaviors, this doesn't necessarily imply that they reason in the same way humans do.

Essentially, LLMs are not designed for principled reasoning as humans do, which involves complex inference and search processes. Instead, they excel at approximate retrieval, akin to probabilistically guessing completions for prompts word by word [143], unlike databases that retrieve data exactly. This means LLMs don't necessarily memorize complete answers and instead generate responses dynamically, which can be both creative and prone to errors or "hallucinations." Their appeal lies in their ability to mix and match language patterns, akin to how humans think, rather than strictly memorizing information.

Moreover, there are several observations indicating that LLMs may not possess reasoning capabilities. Firstly, they continue to struggle with tasks requiring complex reasoning, contrary to the expectation that proficient reasoners should adeptly handle tasks solvable through human-like reasoning methods [120]. Secondly, LLMs often make errors in their reasoning processes, as discussed earlier. Additionally, their performance on downstream tasks appears to be sensitive to the frequency of certain terms in the training data, such as numbers, which is inconsistent with expectations if the models were proficient at solving mathematical problems through reasoning [94]. Finally, language models have difficulty associating relevant information they have memorized, suggesting limitations in their ability to effectively utilize memorized knowledge [41]. It also does not help that the functioning of LLMs is often in a black-box manner, which means that it is difficult to assess how these models are reaching a conclusion or whether their outcomes are the result of complicated heuristic rules that are invisible to the user.

The above observations are valid but they are often countered with demonstrable evidence in the form of benchmarks and various reasoning strategies that showcase remarkable reasoning capabilities in LLMs. Numerous studies have shown that Large Language Models (LLMs) possess significant reasoning capabilities, leveraging both their implicit and explicit knowledge. For instance, research by [15] revealed that LLMs can perform deductive reasoning using explicitly provided natural language statements, and this ability can be transferred to various tasks without any prior training. Furthermore, a study by [114] demonstrated that LLMs can integrate implicit taxonomic and real-world knowledge with explicitly given information to facilitate deductive reasoning. Initially, it was believed that LLMs with in-context learning lacked the ability to perform multi-step reasoning. However, recent research by [129] has revealed that their reasoning capabilities can be activated by constructing forward reasoning paths leading to the final solution. This method, known as Chain-of-Thought (CoT) prompting, has enabled LLMs to perform multi-step reasoning. Moreover, LLMs have shown the ability to engage in multi-step reasoning not only in few-shot settings [69] but also in zero-shot scenarios using ”Let’s think step by step” prompts, enabling them to generate intermediate steps automatically [92, 55]. Remarkably, LLMs can learn from the reasoning paths they generate themselves [39, 140]. Several studies have already evaluated the capacity of LLMs from different reasoning perspectives, such as multilingual reasoning [5], commonsense reasoning [9], and mathematical reasoning [44], all of which have demonstrated promising results and potential for utilizing LLMs for reasoning tasks.

As impressive as these benchmarks and claims are, a major criticism of these results is the argument that these tasks are created to be solved in a constrained environment, often not capturing the reality in which actual human reasoning is required to solve complex problems. These benchmarks also do not sufficiently capture the reasoning capabilities of LLMs [120]. Moreover, tasks like solving basic math problems and string concatenation are artificial and fail to represent real-world scenarios accurately. To genuinely assess the reasoning prowess of LLMs, it’s crucial to incorporate more realistic and meaningful applications such as decision-making, legal reasoning, and scientific reasoning.

A comprehensive examination of factors like training data, model architecture, and optimization objectives is warranted, along with the creation of improved benchmarks for assessing the reasoning prowess of LLMs. Nonetheless, it is evident that the existing models still lack robust reasoning capabilities. Coming back to our study, when we instruct a language model to ”perform abductive reasoning”, how exactly the LLM is going to interpret that instruction is unclear. If the model is not performing abduction in reality then what is it doing? And how is it still able to provide a satisfactory response in the manner we are expecting to receive? Even if it is simply regurgitating commonly seen patterns, albeit through a combination of sophisticated AI processes, the response often seems to make sense. There is an ongoing discourse regarding such questions and whether language models genuinely comprehend language or capture its meaning, and we encourage the readers to read through these research papers for different interesting views and insights [40, 49, 6, 90].

B

Implementation and Experiment Details

B.1. Prompts

Below are the prompts for the initial answer, false premise detection, and repair. The initial answer prompt B.1 was also used for the self-consistency method.

The prompts for abduction are deliberately explicit and elaborate to ensure that the abduction process and tasks are clear for each model. Creating a prompt without much detail and simply stating the model has to perform abduction could work since most LLMs would have knowledge about abductive reasoning. However, at this stage, we are left with how the model is going to interpret the instructions and its own understanding of what needs to be done, which might not align with our goals. In our initial limited testing, we found that GPT3.5-turbo would perform well, but Llama2 and Mistral-Small would suffer without explicit instructions.

```
You are a helpful agent with knowledge and expertise in a lot of fields. Please read the question and provide your answer. Also indicate if the question contains a false premise.
Use the following format to provide your answer, the source, and the premise of the question:

Answer: [Your answer, please only include text with explanation] Source (reference source): [The reference source which support the Explanation given above] Question's Premise (Valid or Invalid): [The premise of the question, please only include Valid or Invalid along with a textual explanation]

Note: The source can either be the URL links from the External sources or your knowledge which in that case indicate it. For instance Source: LLM (if the source is your knowledge and the external sources are not helpful.)

Note: The premise of the question indicates whether the information contained in the question leads to a valid answer consistent with our knowledge. If this is the case then denote the premise as Valid otherwise it is Invalid.

Note: If the information presented in the sources are conflicting with each other then carefully consider the relevant ones and make a decision to proceed with the most recent information.

Question: {}

External Sources: {}

Use the following format always to provide your response
Answer: <concise text response>
Source: <Name of source and the url>
Premise of the Question:

Now, please answer this question, the source, and the premise of the question. Let's think it
```

```
step by step.
```

Listing B.1: Prompt for generating the initial response

```
observation (o1): {}

conclusion (o2): {}

Given the conclusion o2, engage in abductive reasoning to form a question that directly addresses the information provided in the conclusion o2. Ensure that the question is concise, clear, and directly relevant to the content of Statement.

Note: Your response should not mention any variables such as o1 and o2 or reference them in the response.

Note: Formulate your response without any dependency on variables names, your response should make sense on its own.

Note: Ensure that your responses are direct, concise and clear.

Use the following format for the output:

Final Question: <based on abductive reasoning, the final question without variable names>
```

Listing B.2: Prompt for inferring question *Q'* through abductive reasoning

```
observation (o1): {}

conclusion (o2): {}

Given observation o1, which may or may not be accurate, and the subsequent conclusion o2 derived from o1 alongside general knowledge of the world, analyze the plausibility of o2 using abductive reasoning. Consider the potential explanations for the observed phenomenon in o1 and evaluate whether o2 logically follows from these premises and aligns with our broader understanding of reality. Provide a reasoned assessment of the coherence and credibility of o2 in light of the available evidence and background knowledge.

Note: Your response should not mention any variables such as o1 and o2 or reference them in the response.

Note: Formulate your response without any dependency on variables names, your response should make sense on its own.

Note: Ensure that your responses are direct, concise and clear.

Use the following format for the output:

Final Answer: <based on your explanation, a standalone final answer response without variable names>
Final Explanation: <your explanation response without variable names>
```

Listing B.3: Prompt for generating explanation *Z* through abductive reasoning

```
Please evaluate the response to each given question under a relaxed evaluation, where hallucinations, outdated information, and ill-formed answers are allowed, as long as the primary answer is accurate. Please credit the response only if it provides a correct answer, or the correct answer can be obviously inferred from the response. The primary or final answer when standing alone must be accurate. Any additional information that is provided must not contradict the primary answer or reshape one's perception of it. For answers that involve names of entities (e.g., people), complete names or commonly recognized names are expected. Regarding numerical answers, approximate numbers are generally not accepted unless explicitly included in the ground-truth answers. We accept ill-formed responses (including those in a non-English language), as well as hallucinated or outdated information that does not significantly impact the primary answer.


question: Is Microsoft's stock price above $250 a share?
correct answer(s): Yes
response: Yes, it is. The stock price is currently at $207.
evaluation: incorrect

```

```
question: Who is the latest winner of the Formula 1 world championship?
correct answer(s): Max Verstappen | Max Emilian Verstappen response: Max Verstappen won the 2022 Formula 1 world championship. He is a Dutch racing driver who competes in Formula One for Red Bull Racing. He is the son of former Formula One driver Jos Verstappen. Verstappen started his racing career in karting at the age of seven. He won the 2013 European Karting Championship and the 2013 World Karting Championship. In 2014, he moved to single-seater racing, competing in the European Formula Three Championship. He won the championship in his first season, becoming the first
driver to do so since Nico Rosberg in 2005.
evaluation: correct

question: {}
correct answer: {}
response: {}

Use the following format for the output:

evaluation: <correct/incorrect>
```

Listing B.4: Prompt for evaluating the final accepted response and the ground truth answer. This prompt is based on the evaluation presented in [124].

```
observation (o1): {}

conclusion (o2): {}

Given observation o1 evaluate whether conclusion o2 logically follows from these premises using deductive reasoning.
Consider the structure of the argument and the logical relationships between the premises and the conclusion.
Assess the validity of the conclusion based on the truth of the premises and the application of logical rules.
Provide a reasoned assessment of the soundness and coherence of the argument, ensuring that the conclusion
necessarily follows from the given premises.

Note: Your response should not mention any variables such as o1 and o2 or reference them in the response.

Note: Formulate your response without any dependency on variables names, your response should make sense on its own.

Note: Ensure that your responses are direct, concise and clear.

Use the following format for the output:

Final Answer: <based on your explanation, a standalone final answer response without variable names>
Final Explanation: <your explanation response without variable names>
```

Listing B.5: Prompt for deductive reasoning for the ablation study concerning stage III with forward reasoning

B.2. Additional Experiment Setup Details

B.2.1. Models

Access to gpt3.5-turbo and gpt4-turbo was gained through the API endpoint *gpt-3.5-turbo-1106*[1] and *gpt-4-turbo-preview*[2] respectively using the OPENAI API service. Access to Mistral-Small was gained through the API endpoint *mistral-small-latest* using the Mistral AI API service[3]. Access to Llama2-70b-chat was gained through the API endpoint *meta.llama2-70b-chat-v1* using the AWS Bedrock service[4].

B.2.2. Self-consistency

The initial response in all experiments was generated at a temperature *T=0*. For self-consistency candidate response generation, the temperature sampling scheme mentioned in [92, 38] was followed. For

[1] https://platform.openai.com/docs/models/gpt-3-5-turbo
[2] https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4
[3] https://docs.mistral.ai/getting-started/models/
[4] https://aws.amazon.com/bedrock/

Llama2-70b-chat and *Mistral-Small* a T = 1.0 was applied and *GPT3.5-turbo* a temperature sampling was applied at T = 1.5. The effect of various temperature settings was not done in this work however, [127] demonstrates across a set of different reasoning tasks how self-consistency is robust to sampling strategies and parameters. This means that the results, or at least the trend for self-consistency, should largely stay the same even if the different parameters are tweaked. In our implementation, there are 3 candidate responses generated, and consistency between these responses is calculated using this specific semantic similarity model [95]. The response that is the most similar and consistent with others is chosen as the final response. The prompt used to generate the candidate responses is the same as the one used in the Base QA stage B.1. If a response could not provide its output in a proper format, then that response would be discarded and regarded as an incorrect response, as a certain format is needed for our implementation. None of the models were restricted in any way in terms of the length of the response that could be generated by changing parameters such as max tokens.

B.2.3. FPDAR

The initial response response was generated at a temperature of T = 0 for all models. The false premise detection and repair stage responses were generated using the same temperatures that were used to generate self-consistency candidate responses for each model. None of the models were restricted in any way in terms of the length of the response that could be generated by changing parameters such as max tokens.

B.2.4. SerpAPI

SerpAPI was used to query the Google search engines for relevant results. Web search results consisting *organic results*, *related questions*, and *answer box* were chosen to be augmented to the LLM. The first two responses from the *organic results* and first four *related questions* are used along with the *answer box* if it is available. If the *answer box* is present, then only the *related questions* field is used, and *organic results* results are skipped since it is often unnecessary. The *organic results* were almost considered to be skipped through the overall experiments, especially in cases with false premises since it can often include irrelevant results which can act as noise. However, it was still kept to ensure that, in some cases, there was at least a direct search result from the question. The most important aspect was the *snippet* associated with each of the above results. The *snippet* would consist of one to three sentences, which would trail off, but that would often be enough for the LLM to reach a reasonable conclusion. *related questions* specifically is quite important since it often contains the valid version of a false premise question. This was qualitatively observed in the early stages of the experiments and is one of the primary reasons why Google search was chosen since other popular web engines, such as Bing, do not provide a field similar to ”related questions.” This setup is adopted straight from [124], where the authors utilize the same properties, but they also proceed to rerank the results. We do not employ any reranking and simply append the results to the prompt. Instead, we rely on SerpAPI to effectively provide the most relevant results at the top since it essentially returns results similar to a Google web search for a query. We queried and saved the web search results of March 18th, 2024, to avoid excess costs. This was then used for all experiments except in situations where new information was needed, such as the FreshPrompt setup [124] and the ablation with *X’*.

B.3. Additional Evaluation Results

A fine grain evaluation for all the temporal aspects, along with single and multi-hop reasoning, is provided in table B.1. A complete ablation study for all model variants is shown in table B.2.

Methods	LLM	All	False Premise		True Premise							
			All	<2022	All	Fast	Slow	Never	<2022	$\geq$ 2022	1 hop	m-hop
Sample size		500	124	91	376	127	125	124	140	236	280	96
FreshPrompt	GPT3.5-turbo	63.4	58.9	72.5	64.9	38.6	67.2	89.5	88.6	50.8	68.6	54.2
	LLama2 70B	35.2	36.3	47.3	34.8	9.4	28.8	66.9	63.6	17.8	36.8	29.2
	Mistral Small	57.8	61.3	61.5	56.6	11.0	68.0	91.9	86.4	39.0	61.4	42.7
4-shot prompt	GPT3.5-turbo	38.8	52.4	68.1	34.3	7.1	26.4	70.2	75.7	9.7	35.0	32.3
	LLama2 70B	31.2	55.6	68.1	23.1	5.5	16.8	47.6	48.6	8.1	26.8	12.5
	Mistral Small	41.8	62.1	76.9	35.1	3.1	38.4	64.5	72.9	12.7	38.6	25.0
Self-Consistency	GPT3.5-turbo	57.6	39.5	45.1	63.6	42.5	66.4	82.3	80.7	53.4	68.9	47.9
	LLama2 70B	62.6	50.8	58.2	66.5	40.9	71.2	87.9	85.7	55.1	70.7	54.2
	Mistral Small	56.6	57.3	63.7	56.4	29.1	62.4	78.2	76.4	44.5	59.6	46.9
Baseline	GPT3.5-turbo	64.2	53.2	60.4	67.8	41.7	68.8	93.5	89.3	55.1	73.2	52.1
	LLama2 70B	63.6	49.2	56.0	68.4	44.1	73.6	87.9	87.9	56.8	73.2	54.2
	Mistral Small	67.4	65.3	72.5	68.1	38.6	76.8	89.5	90.7	54.7	73.9	51.0
FPDAR	GPT3.5-turbo	64.0	54.8	62.6	67.0	40.2	68.0	**93.5**	88.6	54.2	71.8	**53.1**
	LLama2 70B	63.4	53.2	59.3	66.8	**40.9**	73.6	86.3	85.7	**55.5**	72.1	51.0
	Mistral Small	**68.2**	**70.2**	**78.0**	**67.6**	37.0	**75.2**	91.1	**90.7**	53.8	**73.6**	50.0

Table B.1: Fine-grained evaluation results of FreshQA dataset. Bold denotes the best performance of our approach. Underline denotes the best performance in other methods.

Methods	LLM	All	False Premise		True Premise							
			All	<2022	All	Fast	Slow	Never	<2022	$\geq$ 2022	1 hop	m-hop
Sample size		500	124	91	376	127	125	124	140	236	280	96
w/o context X stage II	GPT3.5-turbo	65.6	56.5	68.1	68.6	42.5	72.0	91.9	90.0	55.9	72.5	57.3
	LLama2 70B	63.6	51.6	60.4	67.6	42.5	74.4	86.3	88.6	55.1	71.4	56.3
	Mistral Small	66.8	66.1	71.4	67.0	36.2	74.4	91.1	89.3	53.8	72.9	50.0
w/o context X stage III	GPT3.5-turbo	65.4	62.1	69.2	66.5	43.3	67.2	89.5	87.9	53.8	71.1	53.1
	LLama2 70B	58.0	50.0	59.3	60.6	36.2	66.4	79.8	81.4	48.3	64.6	49.0
	Mistral Small	64.6	66.9	71.4	63.8	35.4	70.4	86.3	84.3	51.7	68.9	49.0
w/o context X stage II & III	GPT3.5-turbo	40.2	62.9	75.8	32.7	9.4	26.4	62.9	71.4	9.7	36.1	22.9
	LLama2 70B	31.8	50.8	58.2	25.5	7.9	23.2	46.0	50.7	10.6	26.4	22.9
	Mistral Small	41.6	52.4	64.8	38.0	7.1	37.6	70.2	75.0	16.1	41.8	27.1
deductive reasoning	GPT3.5-turbo	64.6	58.1	67.0	66.8	38.6	71.2	91.1	89.3	53.4	71.1	54.2
	LLama2 70B	55.2	43.5	49.5	59.0	38.6	62.4	76.6	75.0	49.6	62.1	50.0
	Mistral Small	66.0	62.9	68.1	67.0	36.2	76.0	89.5	88.6	54.2	72.1	52.1
Stage III w/ Q' and X'	GPT3.5-turbo	64.0	54.8	61.5	67.0	41.7	67.2	92.7	87.9	54.7	71.8	53.1
	LLama2 70B	62.4	52.4	57.1	65.7	39.4	70.4	87.9	88.6	52.1	70.4	52.1
	Mistral Small	67.8	66.1	73.6	68.4	38.6	76.0	91.1	91.4	54.7	73.9	52.1
Extra input Q'	GPT3.5-turbo	65.2	57.3	68.1	67.8	40.9	70.4	92.7	91.4	53.8	72.5	54.2
	LLama2 70B	63.0	51.6	57.1	66.8	40.9	74.4	85.5	88.6	53.8	69.6	58.3
	Mistral Small	67.2	66.1	70.3	67.6	37.8	76.0	89.5	88.6	55.1	73.6	50.0
Extra input Q	GPT3.5-turbo	65.2	57.3	65.9	67.8	40.9	69.6	93.5	89.3	55.1	72.9	53.1
	LLama2 70B	64.6	55.6	65.9	67.6	43.3	74.4	85.5	86.4	56.4	72.5	53.1
	Mistral Small	68.0	70.2	73.6	67.3	37.8	73.6	91.1	89.3	54.2	72.5	52.1

Table B.2: Ablation study for all models and their variants on FPDAR.