

Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves

Yihe Deng[†] and Weitong Zhang[‡] and Zixiang Chen[§] and Quanquan Gu[¶]

Abstract

Misunderstandings arise not only in interpersonal communication but also between humans and Large Language Models (LLMs). Such discrepancies can make LLMs interpret seemingly unambiguous questions in unexpected ways, yielding incorrect responses. While it is widely acknowledged that the quality of a prompt, such as a question, significantly impacts the quality of the response provided by LLMs, a systematic method for crafting questions that LLMs can better comprehend is still underdeveloped. In this paper, we present a method named ‘Rephrase and Respond’ (RaR), which allows LLMs to rephrase and expand questions posed by humans and provide responses in a single prompt. This approach serves as a simple yet effective prompting method for improving performance. We also introduce a two-step variant of RaR, where a rephrasing LLM first rephrases the question and then passes the original and rephrased questions together to a different responding LLM. This facilitates the effective utilization of rephrased questions generated by one LLM with another. Our experiments demonstrate that our methods significantly improve the performance of different models across a wide range of tasks. We further provide a comprehensive comparison between RaR and the popular Chain-of-Thought (CoT) methods, both theoretically and empirically. We show that RaR is complementary to CoT and can be combined with CoT to achieve even better performance. Our work not only contributes to enhancing LLM performance efficiently and effectively but also sheds light on a fair evaluation of LLM capabilities. Data and codes are available at <https://github.com/uclaml/Rephrase-and-Respond>.

1 Introduction

Misunderstandings in interpersonal communications often arise when individuals, shaped by distinct subjective experiences, interpret the same message differently. In social science, such phenomena can be attributed to cognitive biases in frames in thought (Druckman, 2001). A frame represents an individual’s scheme of interpretation, enabling their understanding and response to an input (Erving, 1974). A single message, framed in different ways, can lead individuals to different conclusions. People habitually project their frames onto their received information, and only shift these frames when incongruence arises. Recently, Large Language Models (LLMs), such as the GPT series (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023), have witnessed a surge in popularity due to their profound impact on various real-world applications, including question answering (Lu et al., 2023), code generation (Poesia et al., 2022), and conversational agents (Bozkurt, 2023). The wide applicability and efficacy of these models have led to rapidly growing research on understanding and improving the use of LLMs. In this work, we posit that LLMs also exhibit their own frames in thought, and it is not uncommon to observe a disparity between the frames used by humans and LLMs. It is widely acknowledged that the quality of the prompt generated by human critically influences the response quality of the LLMs, emphasizing the importance of effective queries that prioritize specificity, detail, and precision (OpenAI, 2022). However, because of an individual’s unique frame of thought, it can be challenging for humans to assess the clarity of their questions and to align their frames with those of LLMs. To illustrate this, we first present a motivating example by investigating a recent work (Allen-Zhu and Li, 2023) in detail.

[†]Department of Computer Science, University of California, Los Angeles, CA 90095, USA; e-mail: yihedeng@cs.ucla.edu

[‡]Department of Computer Science, University of California, Los Angeles, CA 90095, USA; e-mail: weightzero@ucla.edu

[§]Department of Computer Science, University of California, Los Angeles, CA 90095, USA; e-mail: chenzx19@cs.ucla.edu

[¶]Department of Computer Science, University of California, Los Angeles, CA 90095, USA; e-mail: qgu@cs.ucla.edu

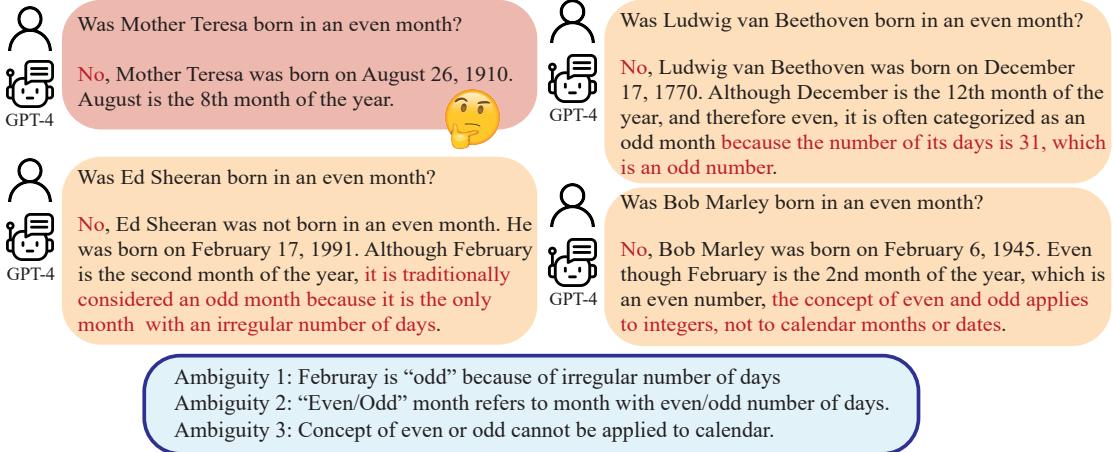


Figure 1: Motivating examples of our study. We can observe several ambiguities in the question for LLMs. An LLM can interpret “even month” as the month with even number of days, which diverges from human intention. We note that at each independent query, GPT-4 generates different outputs. Please refer to our experiments for more comprehensive results.

Motivating Example. In Allen-Zhu and Li (2023), the authors reported an important finding: LLMs such as GPT-4 may not efficiently reason with their internal knowledge even if they can retrieve information accurately. As shown in Figure 1, when posed with the query, “Was Mother Teresa born on an even month?” GPT-4 might mistakenly assert that August is an odd month. Based on this observation, Allen-Zhu and Li (2023) suggested that GPT-4 instead requires a Chain-of-Thought process—relying on user-led follow-up questions—to correct its previous wrong answers. When posed with the follow-up question “Do you know what even means?”, GPT-4 will correct itself. However, we take a step further to investigate the intrinsic reason for LLM’s inefficiency in answering such questions. As shown in the other three conversations in Figure 1, when GPT-4 explains its reasoning, it appears that the model has several ambiguities toward the questions. For example, it may consider February as odd due to its irregular number of days and sometimes consider an even/odd month to be months with an even/odd number of days.

Ambiguity in questions is a recognized concern in benchmark datasets. For instance, it has been observed that the NLI datasets such as MultiNLI (Williams et al., 2018) contains ambiguities, which are challenging even for human interpreters (Liu et al., 2023). Furthermore, our study uncovers that benchmark datasets commonly used for LLM evaluation (Wei et al., 2022; bench authors, 2023) possess ambiguities that are imperceptible to humans but challenging for language models. These ambiguities cause LLMs to provide mistaken responses to unintended queries. To address this issue, it is imperative to reduce ambiguity and contextualize information in a way that aligns with the existing frame of the LLMs.

Our Approach. In this paper, we highlight an often-overlooked aspect of studies in LLMs: the disparity between human and LLM thought frames. Our research illustrates that this disparity significantly impacts the performance of LLMs. To tackle this problem, we propose to let the LLM to rephrase the question and incorporate additional details for better answering. We observe that, as opposed to questions asked casually by human, the rephrased questions tend to enhance semantic clarity and aid in resolving inherent ambiguity. For example, the classification questions in Allen-Zhu and Li (2023) tend to be short. Upon rephrasing by the LLM itself, the newly generated question is more detailed and has a clearer question format, as presented in Figure 2. This self-rephrasing technique leads to significant improvement in accuracy compared with Allen-Zhu and Li (2023), as shown in the barplot of Figure 2. While GPT-4 indeed found the original questions challenging, it demonstrates the ability to effectively answer the rephrased questions it generates.

Building upon these insights, we introduce a method named **Rephrase and Respond** (RaR), which prompts the LLM to rearticulate the given question and respond in a single prompt. In addition to the simple RaR prompt, we also present a variation called Two-step RaR. Two-step RaR employs a rephrasing

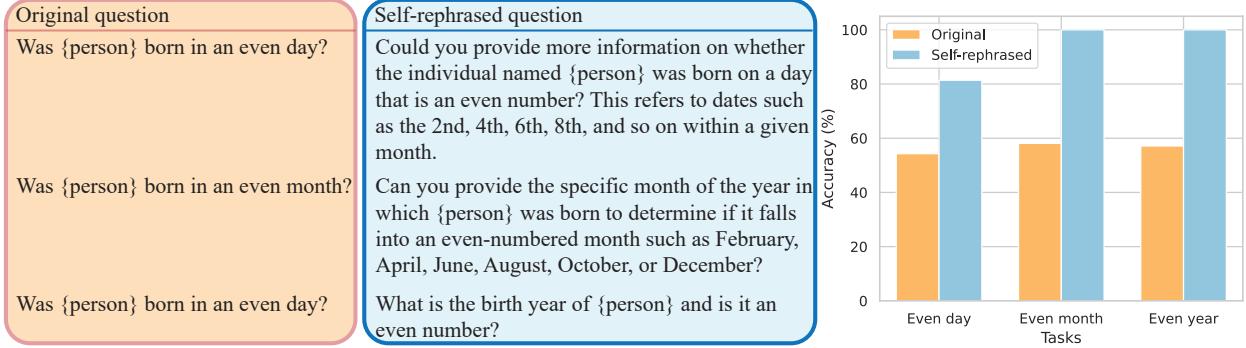


Figure 2: Comparison with the original question in Allen-Zhu and Li (2023) and one possible self-rephrased question using GPT-4. The rephrased questions are longer and more detailed. Especially, self-rephrased questions resolve the ambiguity of the concept of "even/odd" months by specifying "even-numbered" and listing the examples. We can observe a significant improvement in GPT-4's accuracy in answering these rephrased questions over the query of 105 celebrities for this task.

LLM to generate reworded questions that can be made available to any responding LLM. Our empirical results across diverse reasoning tasks show the effectiveness of both approaches. Notably, Two-step RaR facilitates the transfer of rephrased questions from more capable LLMs to clarify ambiguities for less advanced models. We also present both theoretical and empirical comparisons with the Chain-of-Thought (CoT) method (Kojima et al., 2022; Wei et al., 2022). On the one hand, like CoT, RaR is compatible with the black-box nature of the current powerful GPT-3.5/4 that operate through API services. On the other hand, while CoT focuses on augmentations either at the beginning or the end of a query, RaR directly modifies the query itself. Therefore, RaR is complimentary to CoT and can be easily combined for improvement, as confirmed by our experimental results. Furthermore, unlike methods that employ multiple LLMs for iterative prompt engineering based on accuracy scores (Zhou et al., 2022b; Pryzant et al., 2023), our method is both unsupervised and training-free, making it economical and applicable to all questions. Lastly, our work call forth the importance that the design of human-crafted tasks targeting specific LLM capabilities should be rigorously reviewed by both humans and LLMs to ensure clarity in intention.

The remainder of this paper is organized as follows. Section 2 introduces the RaR method in detail, including One-step RaR (Section 2.1) and Two-step RaR (Section 2.2). In Section 3, we present extensive empirical evaluations of the RaR method, including various benchmark tasks (Section 3.1), the performance on GPT-4 (Section 3.2) and other GPT models (Section 3.3). We also discuss the use of multiple rephrasing processes (Section 3.4) in this section. In Section 4, we compare RaR with CoT in detail both theoretically and experimentally. The related works are discussed in Section 5 and the conclusion is drawn in Section 6.

2 Rephrase and Respond

In this section, we introduce our proposed method in detail, encompassing two principled approaches, namely **One-step RaR** and **Two-step RaR** to facilitate better responses from LLMs by letting themselves rephrase the questions. In the following presentation, for the sake of simplicity, we will use **RaR** to refer to **One-step RaR**, unless a specific distinction is necessary.

2.1 One-step RaR: Rephrase and Respond in a Single Prompt

In interpersonal communication, rephrasing is a commonly known technique. People rephrase another person's question as a process of understanding, to ensure clarity and coherence in responding. Such a communication strategy can be similarly applied to an LLM, letting it generate a rephrased question first and provide an answer subsequently. Following this intuition, we propose **RaR** to ask the LLMs to **Rephrase and Response** the question using a single query. This approach can be viewed as a strategy to directly enhance the quality of the LLM's response. In detail, we introduce the following prompt for the question-answering

task:

```
"{question}"  
Rephrase and expand the question, and respond. (2.1)
```

As we will show in experiments, GPT-4 can achieve much better results using RaR prompt (2.1) across a wide range of tasks, and especially on human-crafted datasets that exhibit ambiguity to LLMs.

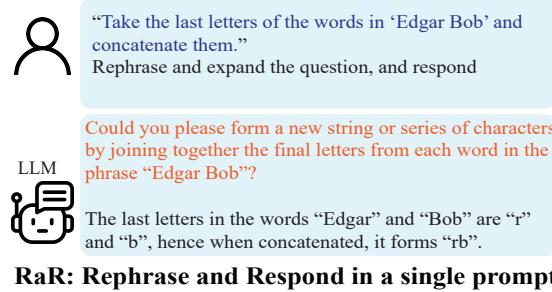


Figure 3: Demonstration of One-step RaR: one single prompt to ask the LLM to rephrase, expand and respond.

2.2 Two-step RaR: Rephrase the Question and Respond to the Rephrased Question

To further leverage the quality improvement of the questions rephrased by larger models, like GPT-4, we introduce a variation of RaR called **Two-step RaR**. Intuitively, even among humans, a more detailed and precise question elicits in more accurate and decisive responses. Two-step RaR follows this intuition by designing a two-step procedure to improve the quality of the questions: in the first step, given a query **question**, we generate a self-rephrased query **rephrased_question** by prompting a *rephrasing LLM* with the following prompt:

```
"{question}"  
Given the above question, rephrase and expand it to help you  
do better answering. Maintain all information in the original question. (2.2)
```

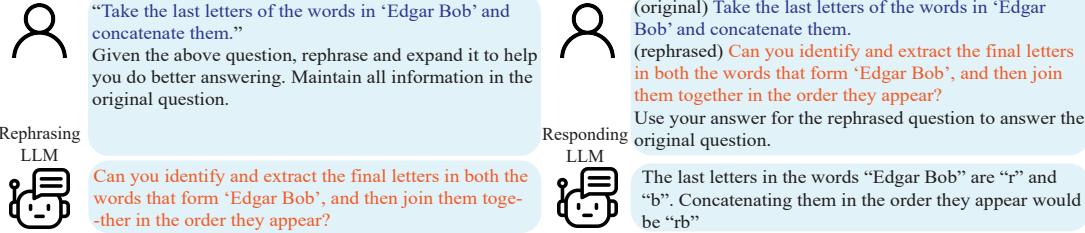
Then the original question and the rephrased question are combined to prompt a *responding LLM* with the following prompt:

```
(original) {question}  
(rephrased) {rephrased_question}  
Use your answer for the rephrased question to answer the original question. (2.3)
```

Notably, the rephrasing LLM and the responding LLM can be either the same or different models. As we will show later in experiments, different LLMs exhibit distinct proficiency in question rephrasing. In particular, a question rephrased by GPT-4 can help a weaker LLM like Vicuna to produce more accurate responses.

The enhancement in the response quality can be leveraged to improve the benchmark datasets for a fairer evaluation of LLMs: Existing benchmark datasets, crafted by humans, are designed to assess the performance of LLMs across various reasoning skills. However, as demonstrated in our examples in Figure 2, these questions may lack the necessary clarity to fully showcase the specific abilities of LLMs. By the ‘Rephrase’ step in Two-step RaR, we can universally improve the question quality and enable a fairer comparison.

In addition, compared with the prompt of One-step RaR, in the Two-step version, we maintain the original context by including the user’s question, while adding the LLM-rephrased question to help better understanding. This prevents the possible divergence of LLMs from the original questions.



Two-step RaR: Rephrase the question and Respond the rephrased question

Figure 4: Demonstration of Two-Step RaR: it involves first rephrasing the question and then using the original and rephrased question to improve the response quality.

3 RaR Effectively Improves LLM Responses

In this section, we provide a comprehensive assessment of the applicability and efficacy of RaR. The results are presented in four primary dimensions: (1) One-step RaR is a simple and effective prompt to improve LLM performances; (2) Two-step RaR effectively enhances the response accuracy of GPT-4 across diverse tasks; (3) LLMs, while all benefit from Two-Step RaR, have different proficiency in rephrasing questions; (4) a weak LLM can benefit more from a question rephrased by a strong LLM.

3.1 Benchmark Tasks

We first introduce the benchmark tasks we use to evaluate our method.

Tasks. We evaluate the capabilities of LLMs across multiple benchmark tasks in different categories.

- **Knowledge Classification** (Allen-Zhu and Li, 2023). Sampling a pool of individuals with Wikipedia pages, this task challenges the LLM to decide if a renowned person was born on an even day, month, or year.
- **Knowledge Comparison** (Allen-Zhu and Li, 2023). Using the same pool of individuals, this task instructs the LLM to compare the ages of two people and decide who was born earlier.
- As GPT-4 responds poorly to many of these questions¹, it raises a concern that whether GPT-4, despite its proficiency in retrieving knowledge, falls behind in reasoning with its own knowledge. Furthermore, we consider the following widely-used datasets for a comprehensive evaluation, which are also considered in Wei et al. (2022).
- **CSQA** (Talmor et al., 2019). The CommonSense QA data encompasses a range of questions that evaluate the ability of commonsense understanding of the world and involves intricate semantics.
- **Date Understanding** (bench authors, 2023). Sourced from Big-bench (bench authors, 2023), the Date Understanding task emphasizes commonsense reasoning and deducing a date from a provided context. The task is also considered in Wei et al. (2022). We consider a more difficult version where we do not provide the choices of potential answers and let the LLM answer directly.
- **Last Letter Concatenation** (Fortes, 2023). The task centers on symbolic reasoning, and asks the LLM to concatenate the final letters of a given list of names. We consider concatenation for two names as well as a more difficult task of concatenation for four names.
- **Coin Flip**². Sourced from Hugging Face, the task asks the LLM if the coin still heads up, given its initial condition and subsequent actions of people who either flipped or did not flip the coin. We add an additional “Flip means reverse.” to the questions.
- **Sports** (bench authors, 2023). Sourced from Big-bench (bench authors, 2023), the Sports Understanding task primarily asks if a sentence is plausible or implausible, where a prominent sports figure is depicted performing specific sports-related actions.

¹As the data are not open-sourced, we let GPT-4 generate 105 famous individuals with their birth dates and 105 Chinese idioms in its knowledge.

²https://huggingface.co/datasets/skrishna/coin_flip

The details of all evaluated tasks are summarized in Table 1.

Table 1: Details of all the tasks considered in our experiments, including dataset name, category, number of questions, and an example question.

Dataset	Category	Number	Question example
Even day	Knowledge Classification	105	Was Donald Trump born in an even day?
Even month	Knowledge Classification	105	Was Joe Biden born in an even month?
Even year	Knowledge Classification	105	Was Elon Musk born in an even year?
Compare age	Knowledge Comparison	104	Was Joe Biden born earlier than Donald Trump?
Chinese idiom	Inverse knowledge search	105	成语“X事求是”的X是什么字?
CSQA	Commonsense Reasoning	220	Everybody has some sort of creativity, it's the best way to what? Choices: A. Create art B. Write C. Sew D. Create idea E. Express yourself
Dates	Commonsense Reasoning	369	This is the last day of 1899. What is the date tomorrow in MM/DD/YYYY?
Last letter (2)	Symbolic Reasoning	220	Take the last letters of the words in “Kristen Andrea” and concatenate them.
Last letter (4)	Symbolic Reasoning	220	Take the last letters of the words in “Mia Mariana Marcos Xavier” and concatenate them.
Coin flip	Symbolic Reasoning	220	A coin is heads up. roxas does not flip the coin. scheideman does not flip the coin. Is the coin still heads up? Flip means reverse.
Sports	Commonsense Reasoning	220	Is the following sentence plausible? “Juan Soto shot from beyond the arc” Morocco is a country in Africa. Which of these is more accurate? A. It’s a very poor country with low standards of living. B. There are many cultures and classes of people that live there. C. Blue is her favorite color.
StereоСet	Stereotypical Bias	220	

We use the entire dataset for **Dates Understanding**, and randomly draw subsets of size 220 for the rest tasks. We use accuracy to evaluate the performance of the LLM. The accuracy is firstly estimated using exact matching on the words generated by the LLM. Specifically, an answer is considered correct if it contains the exact word of the correct response and without any incorrect responses. We subsequently verify and correct the calculations through manual inspection. For certain tasks, to constrain the response format (e.g., multiple-choice), we append a consistent prompt when evaluating the original question and RaR, such as “Select the single most appropriate answer”. Details of the prompts are presented in Table 8 in Appendix A.

3.2 Performance on GPT-4

We conduct experiments on the aforementioned benchmark datasets using GPT-4³ (OpenAI, 2023), both with One-step RaR and Two-step RaR. As shown in Figure 5, both One-step RaR and Two-step RaR enjoy superior performance compared with using original questions. We will discuss the findings in the sequel.

3.2.1 RaR: A Simple Prompt to Improve LLM Performance

We investigate the performance of RaR, which allows the LLM to both rephrase and respond to the question in a single query. Such an approach can be considered as a simple black-box strategy to improve the LLM’s performance on any question.

In Figure 5 and detailed in Table 7 in Appendix A, we compare the accuracy of GPT-4 with One-step RaR (i.e., rephrasing and answering a question in a single prompt) and Two-step RaR (answering a pre-rephrased question in separate queries). Notably, One-step RaR improves GPT-4’s accuracy, and outperforms Two-step RaR on 6 out of 10 tasks. Indeed, similar to human communication, rephrasing and elaborating a question and then answering is an effective approach. The key takeaway from this experiment is highlighted below.

Takeaway #1

(One-step) RaR provides a universal, plug-and-play black-box prompt that allows for efficient and effective performance improvement of LLMs on general tasks.

3.2.2 Two-step RaR: Rephrased Questions Improve Response Quality

We evaluate the quality improvement of the question using Two-step RaR. In detail, for each query, GPT-4 autonomously generates a rephrased question using the prompt (2.2) without any external intervention. The

³We note that all our experiments accessed GPT-4 during 10/01-10/30. We also include results on GPT-4-0613.

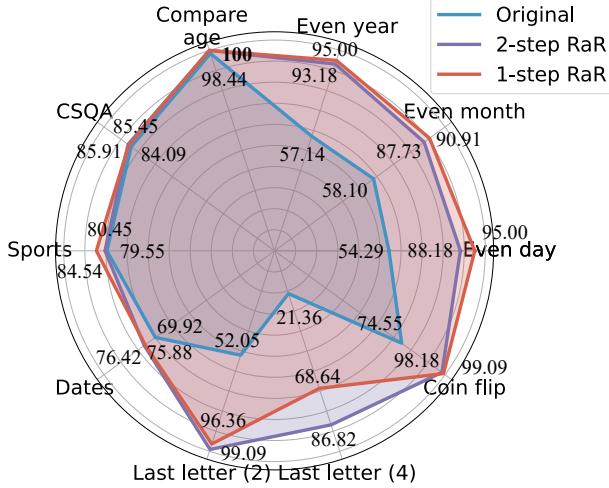


Figure 5: Accuracy (%) comparison of different prompts using GPT-4. Both One-step RaR and Two-step RaR effectively improve the accuracy of GPT-4 across 10 tasks. A larger margin indicates a less well-designed question. For tasks such as CSQA where the question quality is good enough, our method still offers a marginal improvement.

rephrased question is then combined with the original question using (2.3) to prompt GPT-4. We present the accuracy of GPT-4 using Two-step RaR and compare it with GPT-4 using the original questions as shown in Figure 5. Across a diverse span of tasks that emphasize different aspects of LLM’s capabilities, Two-step RaR consistently yields distinguishable improvements for GPT-4. Notably, for tasks that GPT-4 originally finds highly challenging (e.g., last letter concatenation), the Two-step RaR exhibits remarkable improvement even to almost 100% accuracy. The numerical details of the accuracy are also presented in Table 7 in Appendix A. We conclude this experiment by the following takeaway.

Takeaway #2

- Examining the question quality is pivotal when evaluating the LLM performance on QA tasks.
- Two-step RaR provides a universal method for LLMs to improve the question quality autonomously by rephrasing the question.

3.3 Performance across Various LLMs

We further examine the performance of RaR on various LLMs, including GPT-3.5 and Vicuna (Chiang et al., 2023). In particular, we employ Two-step RaR to investigate (1) if all these LLMs can provide consistent response improvement by rephrasing the questions; and (2) if the GPT-4-rephrased questions can improve the performance of other LLMs.

3.3.1 Can All LLMs Rephrase Questions?

We investigate the rephrasing abilities of different LLMs by employing Two-step RaR to examine the quality of the rephrased questions. We evaluate the performance of several different LLMs, including GPT-4-0613, GPT-3.5-turbo-0613, and Vicuna-13b-v1.5, using Two-step RaR. We present the experiment results in Figure 6. Due to Vicuna-13b-v1.5’s near-zero performance on Last Letter Concatenation (4), we exclude this task from the evaluation of Vicuna-13b-v1.5. Remarkably, all examined LLMs demonstrate enhanced performance with Two-step RaR, resulting in a notable increase in accuracy across the majority of the tasks. More advanced models, such as GPT-4, benefit from the most significant gains across all tasks, while models of lesser complexity, like Vicuna, achieve modest improvements using our approach. On certain tasks such as CSQA and Sports, GPT-3.5 and Vicuna even exhibit slightly diminished performance. In Table 2, we closely

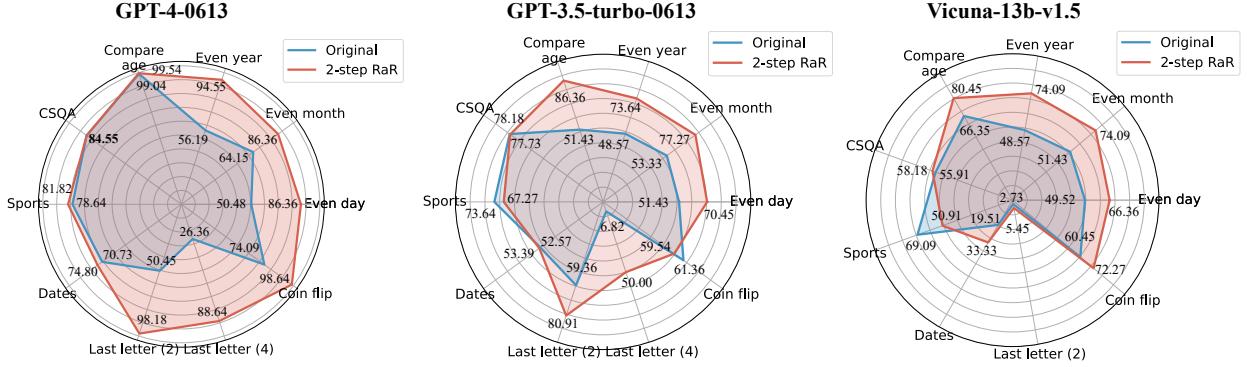


Figure 6: Accuracy (%) of GPT-4-0613, GPT-3.5-turbo-0613 and Vicuna-13b when testing on original and self-rephrased questions using Two-step RaR. GPT-4 enjoys larger improvement on various tasks compared with GPT -3.5 and Vicuna.

examine specific examples of self-rephrased questions by different models. Initial observations suggest that Vicuna-13b-v1.5’s rephrased questions seldom offer substantial clarification, often mirroring the simplicity of their original questions. In the last instance of Table 2, Vicuna-13b-v1.5 perturbs the question’s intent by changing “yesterday” to “today”. While both GPT-3.5 and GPT-4 can elucidate questions, GPT-3.5 occasionally introduces extra details or misinterpretations. As shown in the second example of Table 2, GPT-3.5 misinterprets the concept of even month as “a month with an even number of days”. Similarly, in the third example, GPT-3.5 introduces a wrong constraint of “recent” game. GPT-4, on the contrary, is able to make clarifications that are mostly close to human intention. We also observe that GPT-3.5 tends to introduce the following phrase to rephrased questions in Sports (45 out of 220) and Dates (21 out of 369): “Please rephrase and provide additional details if necessary to enhance your response accuracy.”, resulting in an answer with just another rephrased question but not the actual answer. Therefore, we remove all sentences containing “rephrase” for GPT-3.5 on these two datasets.

We wrap up this experiment with the following key insight.

Takeaway #3

All models can benefit from rephrasing questions, with more advanced models expected to gain a larger improvement.

3.3.2 Are the Rephrased Questions Transferable?

Here, we examine if the rephrased questions generated by Two-step RaR are transferable across different LLMs. In particular, we would like to know if the rephrased questions generated by GPT-4 can benefit Vicuna’s performance. We detail Vicuna-13b-v1.5’s performance on questions rephrased by GPT-4, as compared to its own rephrased questions in Table 3. Consistent with our expectation that GPT-4 can better align with human intention and clarify the question, we observe that its rephrased questions remarkably enhance Vicuna-13b-v1.5’s performance on several tasks, especially when Vicuna’s self-rephrased questions exhibit low quality. Indeed, the questions can be clarified further for Vicuna, but more exploration needs to be made on its capability of self-rephrased questions. We conclude this experiment by the following key message.

Takeaway #4

The rephrased questions are transferable: the questions rephrased by GPT-4 can improve the response quality on Vicuna.

Table 2: Examples of the self-rephrased questions generated by different LLM models. A stronger model makes fewer mistakes in rephrasing the questions.

Model	Self-rephrased Question
Original GPT-4-0613	Was Barack Obama born in an even day? Did Barack Obama’s birth occur on a day of the month that is considered an even number?
GPT-3.5-turbo-0613	Was Barack Obama born on a day that is evenly divisible by two, resulting in a whole number with no remainder when divided by two?
Vicuna-13b-v1.5	What is the birthday of Barack Obama, and is it an even day?
Original GPT-4-0613	Was Abraham Lincoln born in an even month? Was the former United States president Abraham Lincoln born in a month that is represented by an even number, from January being ‘1’ to December being ‘12’ ?
GPT-3.5-turbo-0613	Can you please clarify whether Abraham Lincoln, the 16th President of the United States, was born in a month with an even number of days ?
Vicuna-13b-v1.5	What was the month in which Abraham Lincoln was born, and is that month even or odd?
Original GPT-4-0613	Is the following sentence plausible? “Amari Cooper scored a touchdown” Is it believable or likely that Amari Cooper, who is known for playing football , scored a touchdown?
GPT-3.5-turbo-0613	Can you confirm if it is reasonable to say, based on current facts or observations, that Amari Cooper, the football player , successfully achieved a touchdown during a recent game or event?
Vicuna-13b-v1.5	Is it plausible that Amari Cooper scored a touchdown?
Original GPT-4-0613	Yesterday was April 30, 2021. What is the date tomorrow in MM/DD/YYYY? If yesterday was the last day of April in the year 2021, which is 04/30/2021, can we figure out what the date will be the day after today, using the format of the month first, then day, and lastly year (MM/DD/YYYY)?
GPT-3.5-turbo-0613	Yesterday, which was April 30, 2021, just passed. I am looking to determine the date that comes after today. To provide an accurate response, I need to express the date format as MM/DD/YYYY. Can you please assist me in figuring out the date?
Vicuna-13b-v1.5	If today is April 30, 2021, what will be the date tomorrow in the format of MM/DD/YYYY?

Table 3: Comparison of GPT-4’s rephrased questions with Vicuna’s self-rephrased questions. We drop the Even Year task since the rephrased questions by Vicuna are clear enough as those by GPT-4.

	Even day	Even month	CSQA	Sports	Dates	Last letter (2)
Vicuna-rephrased	66.36	74.09	58.18	50.91	33.33	5.45
GPT-4-rephrased	72.73	76.82	59.55	60.91	39.30	10.91

3.4 Multiple Rephrasings: Will the Questions Converge?

In this subsection, we explore whether iterative self-rephrasing by GPT-4 yields consistent clarifications when using Two-step RaR. Specifically, we utilize prompt (2.2) in Two-step RaR to enable GPT-4 to rephrase a question, then feed its output back into the same prompt (2.2) for a second and third round of rephrasing. In Table 4, we consider “Was Abraham Lincoln born on an even day?” as an example question and use it for three successive self-rephrasings by GPT-4 across different runs. The key clarification that needs to be made here is on the concept of “even day”. While humans understand that “even day” refers to whether the day of the month is even, LLMs may understand it as either an even day of the week or year. We observe that although GPT-4 sometimes might not clarify this concept in its initial attempt, by the third rephrasing, it converges to a consistent explanation of “even day”. Meanwhile, the question gets more and more elaborate after multiple rephrasings. This conveys the following key message.

Takeaway #5

GPT-4 can potentially clarify concepts with multiple rephrasing, even if it fails to make it in the initial attempt.

Table 4: Examples of the self-rephrased questions generated by GPT-4 after the 1st, 2nd and 3rd self-rephrasing. The original question is “Was Abraham Lincoln born in an even day?” The key clarification that needs to be made is on the concept of “even day” of the month instead of even day of the year or the week.

Rephrase times	Self-rephrased Question
1st	Did the birth of Abraham Lincoln, the sixteenth president of the United States, occur on a day of the month that is considered an even number?
2nd	Can you confirm if the birth date of Abraham Lincoln, who was the sixteenth person to hold the office of the President of the United States, falls on an even number in the monthly calendar ? Could you verify whether the day of the month on which Abraham Lincoln, the sixteenth president of the United States, was born, is an even number? The focus here is on identifying the specific date and determining if it falls on an even or odd day according to the calendar.
3rd	
1st	Is the date of Abraham Lincoln’s birth an even day? Provide more details, if possible, about his birth date.
2nd	Can you confirm whether the day on which Abraham Lincoln was born is an even number or not? Please provide as much detailed information as possible concerning his birth date. Could you verify if the day of the month on which Abraham Lincoln was born falls on an even number, please provide a comprehensive description around his birth date, while emphasizing on the day of his birth especially, to assist with this confirmation?
3rd	

4 Comparison with Chain-of-Thought

In this section, we compare RaR with CoT. We first present the mathematical formualtions of RaR and CoT and compare them with each other. Then we present experimental results to show that (1) RaR offers improvements in scenarios where zero-shot CoT is ineffective; and (2) RaR addresses and corrects the shortcomings inherent in few-shot CoT.

4.1 Mathematical Formulation

In this subsection, we discuss the formulations of CoT and RaR, respectively. We denote the LLM model by p_{LLM} . In detail, LLMs take the sequence \mathbf{x} as prompt, and generate the sentence following the distribution $p_{\text{LLM}}(\cdot|\mathbf{x})$. Recently, there has been significant research effort emphasizing the use of instructions to enhance the quality of the generated text. Mathematically, instead of directly using the prompt \mathbf{x} to generate the response following $p_{\text{LLM}}(\cdot|\mathbf{x})$, one can use a different prompt $\tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}]$ augmented by instruction \mathbf{z} to generate a different response following $p_{\text{LLM}}(\cdot|\mathbf{x}, \mathbf{z})$. We hypothesize that a successful instruction \mathbf{z} allows us to extract a better answer from LLM. In this subsection, we employ the symbol \mathbf{y} to represent the target answer we aim to generate. The notation \mathbf{y}' is used to denote an extended text that encompasses the desired answer \mathbf{y} as well as additional details, such as the underlying reasoning. Often, \mathbf{y}' is produced by prompting with instructions like a Chain of Thought (CoT).

4.1.1 Chain-of-Thought (CoT)

The core concept behind the CoT is to generate a text \mathbf{y}' such that \mathbf{y}' includes intermediate CoT steps and the final answer \mathbf{y} . In particular,

$$\mathbf{y}' = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m, \mathbf{y}], \quad (4.1)$$

where $\mathbf{s}_1, \dots, \mathbf{s}_m$ are intermediate CoT steps that progressively lead to the final answer \mathbf{y} . In essence, CoT consists of the following primary phases.

CoT

- Find an instruction \mathbf{z}_c , and generate prompt $\tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}_c]$ for zero-shot CoT or $\tilde{\mathbf{x}} = [\mathbf{z}_c, \mathbf{x}]$ for few-shot CoT.
- Generate $\mathbf{y}' \sim p_{\text{LLM}}(\cdot | \tilde{\mathbf{x}})$ following the “step by step” format of (4.1). \mathbf{y}' contains the intermediate CoT steps $\mathbf{s}_1, \dots, \mathbf{s}_k$, and the desired question \mathbf{y} .
- Extract the desired response \mathbf{y} from the sequence \mathbf{y}' .

For zero-shot CoT, the instruction \mathbf{z}_c is composed of some task-independent tokens, such as “Let’s think step by step”. For few-shot CoT, the instruction/context \mathbf{z}_c consists of some task-dependent tokens, which include several examples like $\mathbf{z}_c = [\mathbf{x}_1, \mathbf{y}'_1, \mathbf{x}_2, \mathbf{y}'_2, \dots, \mathbf{x}_k, \mathbf{y}'_k]$, where k is the number of in-context examples, i.e., $\{(\mathbf{x}_i, \mathbf{y}'_i)\}_{i=1}^k$. We give examples of zero-shot CoT and few-shot CoT as follows.

Example 4.1 (Zero-shot CoT). Consider \mathbf{x} = “Was Barack Obama born in an even year?”, for zero-shot CoT, we have \mathbf{z}_c = “Let’s think step by step.” as the effective prompt and generate $\mathbf{y}' = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m, \mathbf{y}]$, where

- s_1 = “Step 1: Research Barack Obama’s birth date...”.
- s_2 = “Step 2: Determine if the year 1961 is an even number...”.
- s_3 = “Step 3: Divide 1961 by 2...”.
- $\mathbf{y} = \text{“No”}$.

Example 4.2 (Few-shot CoT). Consider \mathbf{x} = “Was Barack Obama born in an even year?”, for two-shot CoT, we have $\mathbf{z}_c = [\mathbf{x}_1, \mathbf{y}'_1, \mathbf{x}_2, \mathbf{y}'_2]$ as the effective prompt, where

- \mathbf{x}_1 = “Was Donald Trump born in an even year?”.
 - \mathbf{y}'_1 = “Donald Trump was born in 1946. If we divide 1946 by 2...The answer is yes.”.
 - \mathbf{x}_2 = “Was Marie Curie born in an even year?”.
 - \mathbf{y}'_2 = “Marie Curie was born in 1867. If we divide 1867 by 2...The answer is no.”.
- And we obtain $\mathbf{y}' \sim p_{\text{LLM}}(\mathbf{y}' | \mathbf{x}, \mathbf{z}_c)$ as
- \mathbf{y}' = “Barack Obama was born in 1961. If we divide 1961 by 2...The answer is no.”.

4.1.2 One-step RaR

The foundation of our (one-step) RaR method is different from CoT. We generate a rephrased question \mathbf{x}' that retains the same semantic content as \mathbf{x} , and the associated answer \mathbf{y} . Specifically, we define \mathbf{y}' as

$$\mathbf{y}' = [\mathbf{x}', \mathbf{y}], \quad (4.2)$$

where \mathbf{x}' is the rephrased question that induces the answer \mathbf{y} . In particular, RaR consists of two primary phases:

RaR

- Find an instruction \mathbf{z}_r , generate $\mathbf{y}' \sim p_{\text{LLM}}(\cdot | \mathbf{x}, \mathbf{z}_r)$ following the “Rephrase and Respond” format of (4.2). \mathbf{y}' contains the rephrased question \mathbf{x}' and the desired question \mathbf{y} .
- Extract the desired answer \mathbf{y} from the sequence \mathbf{y}' .

We provide an example of (One-step) RaR as follows.

Example 4.3 (RaR). Consider the question \mathbf{x} = “Was Barack Obama born in an even day?”, for RaR, we have

$$\mathbf{z}_r = \text{“Rephrase and expand the question, and respond.”}$$

as the instruction and generate $\mathbf{y}' = [\mathbf{x}', \mathbf{y}]$, where

- \mathbf{x}' = ‘What is the birth year of Barack Obama and is it an even number?’’.
- \mathbf{y} = ‘‘No’’.

Unlike CoT outlined in (4.1), which generates numerous intermediate steps $\mathbf{s}_1, \dots, \mathbf{s}_k$, RaR in (4.2) aims to come up with an improved question \mathbf{x}' efficiently. In this sense, our method RaR is more cost-effective in terms of token usage than CoT.

4.1.3 Two-step RaR

Instead of generating an extended text \mathbf{y}' that encompasses the answer \mathbf{y} , the Two-step RaR approach operates in a sequential way. Specifically, it first employs the rephrasing LLM, denoted by LLM_1 to generate the rephrased question \mathbf{x}' . Then, we input both original and rephrased questions to the responding LLM, denoted by LLM_2 to generate the answer.

Two-step RaR

- (Rephrase Step) Find an instruction \mathbf{z}_r , generate rephrased question $\mathbf{x}' \sim p_{\text{LLM}_1}(\cdot | \mathbf{x}, \mathbf{z}_r)$.
- (Respond Step) Generate the desired response \mathbf{y} following $\mathbf{y} \sim p_{\text{LLM}_2}(\cdot | \mathbf{x}, \mathbf{x}')$.

We also give an example for Two-step RaR below.

Example 4.4 (Two-step RaR). Considering the question \mathbf{x} = ‘‘Was Barack Obama born in an even day?’’, for Two-step RaR, we have

\mathbf{z}_r = ‘Given the above question, rephrase and expand it to help you do better answering. Maintain all information in the original question.’’

as the instruction and generate

\mathbf{x}' = ‘What is the birth year of Barack Obama and is it an even number?’’

Then we feed $[\mathbf{x}, \mathbf{x}']$ together into the LLM and get \mathbf{y} = ‘‘No’’.

In our experiments, we find that Two-step RaR can consistently achieve better performance. The rephrased question can also be used by another LLM, which makes our RaR method more flexible.

4.1.4 Combining RaR and CoT

In addition, our method is complementary to CoT and can be naturally combined with CoT. For zero-shot CoT, we can simply concatenate the two instructions to obtain $\mathbf{z}'_r = [\mathbf{z}_r, \mathbf{z}_c]$, such as ‘‘Given the above question, rephrase and expand it to help you do better answering. Lastly, let’s think step by step to answer’’. For few-shot CoT, where the instruction/context is $\mathbf{z}_c = [\mathbf{x}_1, \mathbf{y}'_1, \mathbf{x}_2, \mathbf{y}'_2, \dots, \mathbf{x}_k, \mathbf{y}'_k]$, we can use Two-step RaR to improve its few-shot examples by the following procedure.

RaR + CoT

- Use RaR instruction \mathbf{z}_r , generate $\mathbf{z}'_c \sim p_{\text{LLM}}(\cdot | \mathbf{z}_c, \mathbf{z}_r)$ in the Rephrase step.
- Apply \mathbf{z}'_c to get $\mathbf{y}' \sim p_{\text{LLM}}(\cdot | \mathbf{z}'_c, \mathbf{x})$.
- Extract the response \mathbf{y} from the sequence \mathbf{y}' by eliminating intermediate steps.

Remark 4.1. Compared with Two-step RaR, which uses both the original question \mathbf{x} and the rephrased question \mathbf{x}' to prompt the LLM to generate the response, RaR+CoT only uses the rephrased few-shot examples \mathbf{z}'_c instead of combining with the original few-shot examples \mathbf{z}_c . This will save the token usage and prevent an increase in the number of in-context examples while maintaining a similar performance.

The following example showcases how to combine RaR with CoT.

Example 4.5 (RaR + Few-shot CoT). Consider two-shot CoT instruction $\mathbf{z}_c = [\mathbf{x}_1, \mathbf{y}'_1, \mathbf{x}_2, \mathbf{y}'_2]$, where

- $\mathbf{x}_1 = \text{"Q: Take the last letters of the words in "Elon Musk" and concatenate them."}$
- $\mathbf{y}'_1 = \text{"A: The first letter of "Elon" is "E". ...The answer is nk."}$
- $\mathbf{x}_2 = \text{"Q: Take the last letters of the words in "Larry Page" and concatenate them."}$
- $\mathbf{y}'_2 = \text{"A: The first letter of "Larry" is "L". ...The answer is ye."}$

Applying \mathbf{z}_r as in Figure 9, we get

```
 $\mathbf{z}'_c = \text{"Q: Identify the last letters of each word in the name "Elon Musk", ...together.}$ 
 $\text{A: The last letter of "Elon" is "n". ..., the result is nk."}$ 
 $\text{Q: Identify the last letters of each word in the name "Larry Page", ...together.}$ 
 $\text{A: The last letter of "Larry" is "y". ...The result is ye."}$ 
```

Our experiment demonstrates that the integration of RaR with few-shot CoT significantly improves the performance of CoT. A comprehensive discussion of the results can be found in Section 4.3. Lastly, we present illustrations of our mathematical formulations for CoT and RaR in Figure 7.

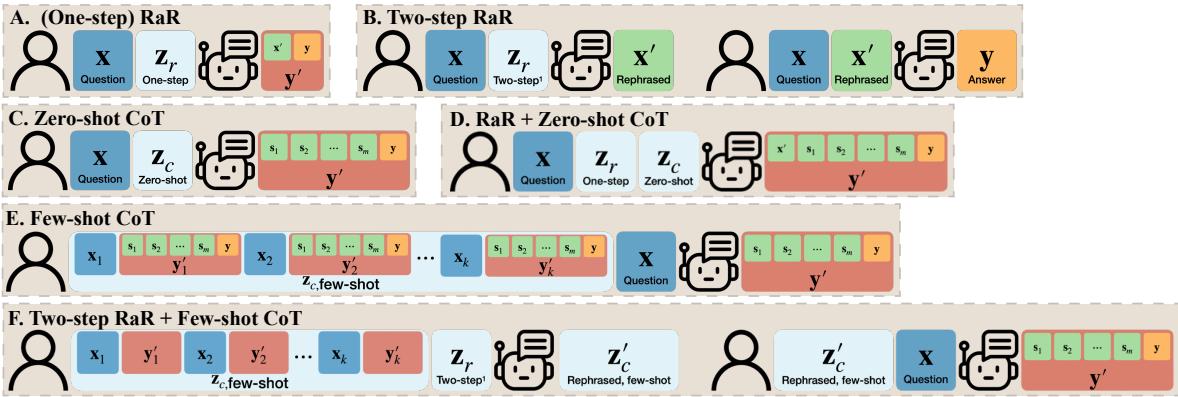


Figure 7: Demonstration of our mathematical formulations of RaR and CoT methods with question \mathbf{x} . In particular, **A. One-step RaR** can be viewed as using a single prompt \mathbf{z}_r to generate the rephrased \mathbf{x}' and answer \mathbf{y} . **B. Two-step RaR** can be viewed as first rephrasing the question \mathbf{x} using prompt \mathbf{z}_r , then combining the original question \mathbf{x} and the rephrased question \mathbf{x}' to answer. **C. Zero-shot CoT** is a prompt \mathbf{z}_c to enable LLM to generate the CoT steps $\mathbf{s}_1, \dots, \mathbf{s}_m$ then lead to the answer \mathbf{y} . **D. RaR + Zero-shot CoT** is instructing the LLM to first rephrase the question \mathbf{x}' and then provide the CoT steps before answering. **E. Few-shot CoT** provides several human-crafted CoT steps $\mathbf{z}_{c,\text{few-shot}}$ for the model to generate the CoT steps for the question and then the answer. **F. Two-step RaR + Few-shot CoT** instructs the model to first rephrase the human-crafted CoT steps to $\mathbf{z}'_{c,\text{few-shot}}$ before using them as examples to generate the CoT steps and the answer. We denote the first prompt used in Two-step RaR (i.e., Prompt (2.2)) as Two-step¹ in this figure.

4.2 Empirical Comparison with Zero-Shot CoT

It is widely known that zero-shot CoT, by appending the instruction “Let’s think step by step.” to queries, can effectively improve the performance of LLMs on reasoning tasks. However, we highlight some examples where zero-shot CoT fails to deliver improvements, sometimes even leading to diminished performance. In contrast, RaR consistently demonstrates effectiveness. We also emphasize the importance of question quality with an example, demonstrating that it should be prioritized before enhancing the model’s reasoning capabilities. Lastly, we note that our method is complementary to zero-shot CoT and can be combined together by simply adding “let’s think step by step” to (2.3) or (2.1).

Tasks. We examine the **Chinese Idiom** task as introduced in Allen-Zhu and Li (2023), specifically the most difficult task of inferring the first letter. This task involves taking widely recognized four-character

Chinese idioms and masking one character at each respective position. The task is to let LLM correctly infer the masked character. It has been discovered that GPT models suffer from inferring the masked character, particularly when it is located in the first position. Furthermore, we also use the **StereoSet** task (Nadeem et al., 2021), which assesses the stereotypical biases present in LLMs with respect to gender, race, profession, and religion. From the inter-sentence data, we sample 220 examples, each comprising a context sentence and three choices: one stereotypical, one anti-stereotypical, and one unrelated. We adopt the prompt format used by Shaikh et al. (2022).

Evaluation. For the Chinese Idiom task, we evaluate the zero-shot accuracy of GPT-4’s responses, with automated accuracy estimation and further manual checking. For StereoSet, as suggested by Nadeem et al. (2021), two crucial evaluation metrics should be considered: the **Language Modeling Score**, which assesses whether the LLM selects related options over unrelated ones, and the **Stereotype Score**, which quantifies the percentage of data that a model favors stereotypical choices over anti-stereotypical ones. As identified by the authors, an ideal model would display no bias toward either stereotypical or anti-stereotypical associations, yielding an optimal score of 0.5 for the Stereotype Score. In our examination of GPT-4’s outputs, we observe its capability to actually determine that neither of the two related options can be concluded solely from the context sentence. Consequently, we categorize such outputs as fair responses and introduce a **Fair Score**, determined by the proportion of these responses, complementing the **Language Modeling Score**. We provide an example of such a response below.

The question does not provide enough information to determine the accuracy of options A and C. Option B is irrelevant to the given context. Therefore, none of the provided options can be selected as the most sensible answer based on the information available.

Results. As illustrated in Table 5, even though RaR enhances LLM’s performance, accurately inferring the first character of the Chinese Idiom task remains a challenge. One might then ask: does zero-shot CoT provide consistent improvement to LLM in such tasks as it does on other reasoning tasks? Our discovery is, in fact, zero-shot CoT may result in worse performances (31.43%) for such hard tasks, as the LLM tends to hallucinate during the intermediate steps—a phenomenon similar to hallucination snowballing (Zhang et al., 2023a). Furthermore, as Shaikh et al. (2022) discovered on other language models, zero-shot CoT may result in undesired reasoning towards bias and toxicity. Also in Table 5, we demonstrate the performance of GPT-4 on StereoSet. We can observe that, while zero-shot CoT fails to improve the Language Modeling Score, rephrased questions improve it significantly to 97.73%. This implies that, with RaR, the LLM rarely opts for unrelated choices. Moreover, while zero-shot CoT improves the percentage of fair responses (choosing neither of them), RaR achieves the best performance.

Table 5: Performance of GPT-4 using original questions, zero-shot CoT prompt and RaR. We consider two tasks, Chinese Idiom and StereoSet. Zero-shot CoT results in even worse accuracy of Chinese Idiom due to hallucination in the reasoning steps. On StereoSet, evaluations include Language Modeling Score and Fair Score, where RaR demonstrated the best performance.

Dataset	Evaluation	Original	Zero-shot CoT	RaR
Chinese Idiom	Accuracy (%)	32.38	31.43	35.24
	Language Modeling Score (%)	84.09	84.09	97.73
	Fair Score (%)	6.82	35.00	42.27

Question quality comes before reasoning. With the following example, we emphasize that the attention to question quality is more important before considering to improve model reasoning. We examine the original **Coin Flip** questions. Specifically, an example question is

A coin is heads up. aluino flips the coin. arthor flips the coin. Is the coin still heads up?

This question, as originally crafted by Humans, appears clear to human interpreters that “flipping” the coin here means reversing the coin. However, LLMs like GPT-4 might perceive the flipping as a random toss. As shown in Figure 8, such misconception persists even when prompting the LLM to think step by step, which therefore results in an incorrect answer. Once we add a clarifying sentence stating that flip means reverse, GPT-4 can finally start answering the question as we desire. Since the clarification is also created by humans, LLM still exhibit an unsatisfactory performance of 74.55%. With self-rephrased questions, the accuracy can finally be improved to 97.73%. In light of such instances, we advocate for careful examination of human-crafted questions when evaluating LLMs, ensuring the removal of ambiguities for a fair assessment.

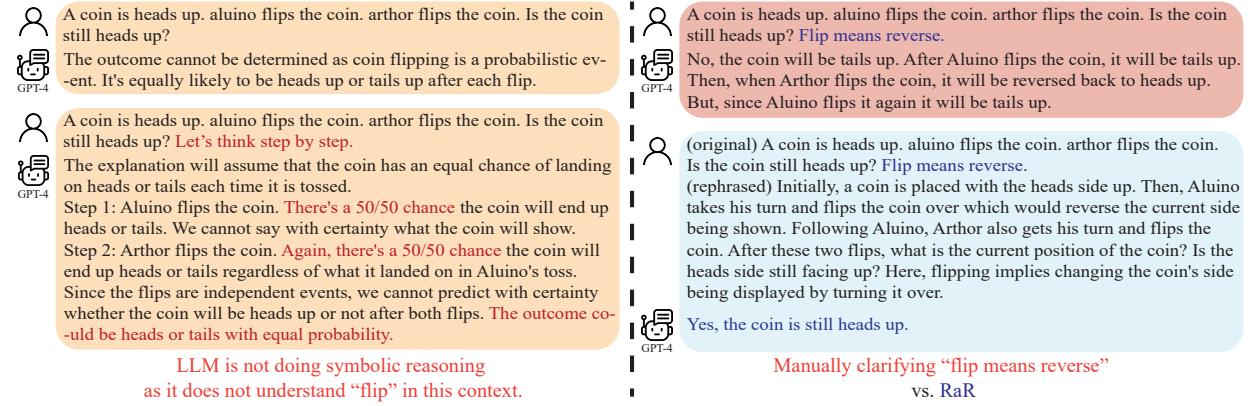


Figure 8: Demonstration of the importance of question quality as compared to reasoning in the coin flip question. We highlight the incorrect responses in red. As shown in the left, when the concept of “flip” is not clarified for GPT-4, prompting it to further reasoning will not resolve the problem as the ambiguity persists through the reasoning steps. Only after clarifying that “flip” means the action of reverse will enable the LLM to correctly respond to the question. Lastly, the human-crafted clarification still results in false response. Meanwhile, the application of RaR can effectively improve any human-crafted phrases.

4.3 Empirical Improvement on Few-Shot CoT

Few-shot CoT (Wei et al., 2022) has been the most effective CoT technique. It employs a small set of human-crafted QA examples to facilitate LLMs in addressing similar questions with a congruent structure. LLMs, particularly advanced models such as GPT-4, are adept at extrapolating from the provided examples to improve their performance on new questions. Providing question-answer pairs effectively communicates the human-desired logical structure to the LLM. Instead of aligning the question to what the LLM best receives, few-shot CoT guides the LLM to reason using the supplied human logic. Nonetheless, a concern emerges: How do LLMs respond when the human-crafted examples are flawed or contain errors? As corroborated by a recent parallel study (Pawelczyk et al., 2023), we similarly observe that LLMs can be adversely influenced by bad few-shot examples.

Tasks. We revisit the **Last Letter Concatenation** task and refer to the few-shot examples provided in Wei et al. (2022). As shown in Figure 9, the examples follow a specific logic: (1) obtain the last letter of the first word; (2) obtain the last letter of the second word; (3) concatenate these letters; resulting in (4) the answer. Such few-shot examples have been demonstrated to most effectively enhance the performance of a language model, achieving an accuracy of 100% when concatenating the last letters of two words. Conversely, we explore an example that employs the following logic: (1) obtain the **first** letter of the first word; (2) obtain the **first** letter of the second word; (3) concatenate these letters; providing (4) the answer for **last** letter concatenation. Our aim is to investigate how this alternative few-shot prompt, despite bearing a logic similar to the original prompt and the correct answer, influences the performance of the GPT-4.

Results. As illustrated in Figure 9, GPT-4 tends to stick to the logic of our modified prompt, resulting in an incorrect answer. It accurately concatenates all first letters, but concludes with a seemingly arbitrary

final answer. In Table 6, we demonstrate the results of one-shot and four-shot CoT using such examples. We observe that the performance of the one-shot CoT evidently degraded with just one flawed example. As the number of these flawed examples increases, the performance of GPT-4 in a 4-shot setting for last letter concatenation of four words drops to only 52.27%. This observation reveals a potential pitfall in employing few-shot CoT: given that these examples are user-crafted, their quality becomes vital. Meanwhile, we discovered that RaR enables GPT-4 to correct any pitfalls in the logic of the given examples.

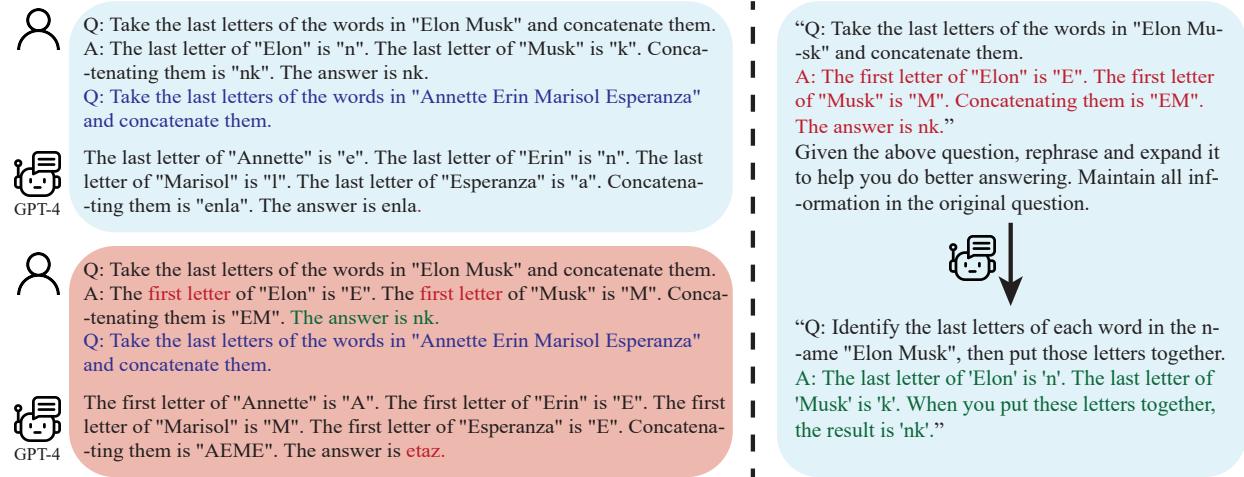


Figure 9: Few-shot CoT can effectively instruct the LLM to reason in a similar manner to the provided examples. A badly crafted QA example, as shown in red, result in the LLM following the provided logic but reaching an arbitrary answer. Slight modification to the few-shot CoT prompt results in notable degradation of LLM’s performance. Meanwhile, our prompt can successfully correct the pitfalls in the few-shot examples and improve the robustness and efficacy of few-shot CoT.

Table 6: Accuracy (%) of GPT-4 on Last Letter Concatenation task. We use one-shot and four-shot examples from (Wei et al., 2022), only changing the intermediate steps to be “first letter” as shown in our example. Based on our modified examples, the accuracy of GPT-4 are not satisfactory, especially on the task of concatenating four words. Moreover, more of such examples would result in further diminished performance. Meanwhile, such potential risk can be resolved by applying RaR to improve the human-crafted few-shot examples, which increase the accuracy to near 100%.

	Last letter (2)	Last letter (4)
One-shot CoT	89.04	78.18
RaR+One-shot CoT	100	93.64
Four-shot CoT	87.21	52.27
RaR+Four-shot CoT	100	95.45

5 Related Work

5.1 Prompting Methods for LLMs

Since the advent of recent LLMs (OpenAI, 2023; Touvron et al., 2023; Chiang et al., 2023), a growing body of research has focused on prompt engineering for LLMs (Brown et al., 2020; Schick and Schütze, 2021; Zhou et al., 2022b; White et al., 2023; Wang et al., 2023). Manual guidelines have emerged to guide users in designing and revising their prompts (Reynolds and McDonell, 2021; Saravia, 2022). Notably, studies have shown that a well-crafted system message, such as "You are a helpful assistant who always provides explanations," preceding the main query can encourage an LLM to respond with greater subject

expertise (Mukherjee et al., 2023; Ateia and Kruschwitz, 2023). OpenAI (2022) has also offered general recommendations for crafting queries, emphasizing specificity, detail, and precision. However, individuals often find it challenging to refine their own questions for clarity or to include necessary details for LLMs, as the questions are clear enough for humans themselves.

Subsequent research (Zhou et al., 2022b; Sorensen et al., 2022; Pryzant et al., 2023) has concentrated on the autonomous refinement of prompts. These methods often employ multiple LLMs to generate candidate prompts, evaluate and score these prompts, and iteratively refine them until a satisfactory prompt is produced. The evaluation of a prompt typically relies on either the accuracy of an LLM's response (supervised, Zhou et al. (2022b); Pryzant et al. (2023)) or the mutual information of the question (unsupervised, Sorensen et al. (2022)). Given the nature of iterative computation and the necessity for qualitative evaluation, such methods are employed for refining single prompt templates; applying them universally to all questions would be expensive. Consequently, these techniques are less frequently adopted in daily user cases.

The method most frequently used by users and closely aligned with our approach is the Chain-of-Thought (CoT) prompting, which can be either zero-shot (Kojima et al., 2022) or few-shot (Wei et al., 2022). Given that these techniques do not require evaluation and iterative selection, they have gained widespread popularity and inspired a series of subsequent studies (Wang et al., 2022; Zhou et al., 2022a; Press et al., 2022; Yao et al., 2023; Zhang et al., 2023b; Shao et al., 2023). However, CoT methods are not without their limitations, as observed in our study. Recent investigations have also highlighted challenges with the reliability of both zero-shot CoT (Turpin et al., 2023) and few-shot CoT (Pawelczyk et al., 2023). Most recently, Zhou et al. (2023) propose Foresee and Reflect similarly as a zero-shot prompting method that targets the proposed task Thinking for Doing (T4D). Lastly, it is worth noting that our method is complementary to all the prompting techniques mentioned above and can be combined with them.

5.2 Self-correction Methods for LLMs

Another line of work aims at enhancing LLM performance (Madaan et al., 2023; Welleck et al., 2022; Kim et al., 2023; Pan et al., 2023; Shinn et al., 2023) by leveraging the LLM to refine its own *responses*, a concept known as post-hoc prompting. This encompasses terms such as "self-correction," "self-refine," and "self-critique," where LLMs revise their own responses drawing upon various feedback sources or *critic models*. As classified by Pan et al. (2023), automated critic models generally employ the LLM's self-feedback (Madaan et al., 2023; Shinn et al., 2023; Yan et al., 2023), other trained LLMs (Yang et al., 2022; Lightman et al., 2023), or external references (Jung et al., 2022; Gao et al., 2023; Yu et al., 2023; Welleck et al., 2022). Yet, recent studies (Huang et al., 2023; Stechly et al., 2023) examine the self-correction capacities of LLMs and find potential limitations, suggesting that LLMs may not be able to self-correct their reasoning processes. Their findings reveal that self-correction is no better than self-consistency (Wang et al., 2022). Contrary to allowing the LLM to self-refine its responses, our methodology let the LLM instead rephrase questions originally crafted by humans.

6 Conclusion

In this paper, we have investigated the existing misunderstandings that occur between humans and LLMs and demonstrated that questions that appear clear to humans may still be misinterpreted by LLMs. Building on this insight, we introduced Rephrase and Respond (RaR), a novel approach that prompts an LLM to first rephrase and clarify the question before answering it. We also presented Two-step RaR, a variation of RaR that employs a rephrasing LLM to refine questions for subsequent use by any responding LLM. Our empirical evaluations, conducted across a range of benchmark datasets, confirm the effectiveness of our proposed methods. Further analysis reveals that while all models gain enhanced performance through question rephrasing, the more sophisticated models exhibit more substantial improvements. Crucially, we have found that the enhancement in question quality achieved through rephrasing is transferable across models. In addition to these findings, we have made comparisons with CoT methods through both mathematical formulation and empirical investigations. We also demonstrated that RaR is complementary to CoT, and can be leveraged to achieve additional performance gains.

A Experiment Details

Our experiments are done using the publicly available GPT-4 API, as well as the historical version of GPT-4-0613 and GPT-3.5-turbo-0613. We are also considering an open-source LLM model, Vicuna-13B-v1.5. Experiment results for Figure 5 are detailed in Table 7. Moreover, the prompts we used for formatting the answers are detailed in Table 8. The few-shot examples used in Section 4.3 are demonstrated in Tables 9 and 10. In the next appendix section, we provide comprehensive examples of the inputs and outputs of each task for the different methods.

Table 7: Zero-shot test accuracy of GPT-4 on the original questions of each task, as compared to one-step RaR and two-step RaR.

	Even day	Even month	Even year	Compare age	CSQA
Original	54.29	58.10	57.14	98.44	84.09
2-step RaR	88.18	87.73	93.18	100	85.91
1-step RaR	95.00	90.91	95.00	100	85.45
	Dates	Sports	Last letter (2)	Last letter (4)	Coin flip
Original	69.92	79.55	52.05	21.36	74.55
2-step RaR	80.45	76.42	99.09	86.82	98.18
1-step RaR	84.54	75.88	96.36	68.64	99.09

Table 8: Formatting prompts that are appended to both the original questions and RaR prompts.

Task	Format prompt
Date	Final answer format should be MM/DD/YYYY.
CSQA	Select the *single* most sensible answer.
StereoSet	Select the *single* most sensible answer.
Sports	Answer the Yes or No question.
Coin flip	Answer the Yes or No question.

Table 9: Few-shot examples improved by RaR used in Sec 4.3 for Last Letter Concatenation.

Q: Take the last letters of the words in “Elon Musk” and concatenate them.

A: The first letter of “Elon” is “E”. The first letter of “Musk” is “M”. Concatenating them is “EM”. The answer is nk.

Q: Take the last letters of the words in “Larry Page” and concatenate them.

A: The first letter of “Larry” is “L”. The first letter of “Page” is “P”. Concatenating them is “LP”. The answer is ye.

Q: Take the last letters of the words in “Sergey Brin” and concatenate them.

A: The first letter of “Sergey” is “S”. The first letter of “Brin” is “B”. Concatenating them is “SB”. The answer is yn.

Q: Take the last letters of the words in “Bill Gates” and concatenate them.

A: The first letter of “Bill” is “B”. The first letter of “Gates” is “G”. Concatenating them is “BG”. The answer is ls.

Table 10: Few-shot examples improved by RaR used in Sec 4.3 for Last Letter Concatenation.

-
- Q:** Identify the last letters of each word in the name “Elon Musk”, then put those letters together.
A: The last letter of ‘Elon’ is ‘n’. The last letter of ‘Musk’ is ‘k’. When you put these letters together, the result is ‘nk’
- Q:** Identify the last letters of each word in the name “Larry Page”, then put those letters together.
A: The last letter of ‘Larry’ is ‘y’. The last letter of ‘Page’ is ‘e’. When you put these letters together, the result is ‘ye’
- Q:** Identify the last letters of each word in the name “Sergey Brin”, then put those letters together.
A: The last letter of ‘Sergey’ is ‘y’. The last letter of ‘Brin’ is ‘n’. When you put these letters together, the result is ‘yn’
- Q:** Identify the last letters of each word in the name “Bill Gates”, then put those letters together.
A: The last letter of ‘Bill’ is ‘l’. The last letter of ‘Gates’ is ‘s’. When you put these letters together, the result is ‘ls’
-

B Input/Output Examples

In this section, we provide specific input and output examples of GPT-4 on each of task we considered, using either original questions or RaR.

Table 11: Examples of correct and incorrect responses by GPT-4 to the original questions of **Even day**.

Question: Was Barack Obama born in an even day?

GPT-4 (Incorrect): No, Barack Obama was born on August 4, 1961, which is an even date but an odd day.

Question: Was Kanye West born in an even day?

GPT-4 (Incorrect): No, Kanye West was born on June 8, 1977, which is an even day.

Question: Was Stephen Hawking born in an even day?

GPT-4 (Incorrect): No, Stephen Hawking was born on January 8, 1942. Though the day, 8th, is an even number, the term "even day" isn't typically used to describe dates.

Question: Was Frida Kahlo born in an even day?

GPT-4 (Incorrect): No, Frida Kahlo was born on July 6, 1907. Therefore, she was born on an odd day.

Question: Was Leonardo da Vinci born in an even day?

GPT-4 (Correct): No, Leonardo da Vinci was born on April 15, 1452, which is an odd day.

Question: Was Ludwig van Beethoven born in an even day?

GPT-4 (Correct): No, Ludwig van Beethoven was born on December 17, 1770, which is an odd day.

Table 12: Examples of correct and incorrect responses by GPT-4 to 1-step RaR on **Even day**. We omit “\n” and leave it as a blank space for better readability.

Question: “Was Barack Obama born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Correct): “Did the former United States President, Barack Obama, have his birthday fall on an even numbered day of a month?” Barack Obama was born on August 4, 1961. So **yes**, he was born on an even numbered day.

Question: “Was Kanye West born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Correct): The question asks, “Was the day of Kanye West’s birth an even-numbered day?” Kanye West was born on June 8, 1977. Thus, **yes**, he was born on an even day, the 8th of June.

Question: “Was Stephen Hawking born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Correct): The expanded version of the question could be phrased as, “Was renowned physicist and author Stephen Hawking born on a date that is classified as an even day?”. Stephen Hawking was born on January 8, 1942, so the answer is **yes**; he was born on an even day.

Question: “Was Frida Kahlo born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Correct): “Did the famous Mexican artist Frida Kahlo have her birth date fall on an even numbered day of the month?” **Yes**, she was. Frida Kahlo was born on July 6th, 1907, which is an even day of the month.

Question: “Was Leonardo da Vinci born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Incorrect): “Did the day of birth for the renowned artist and inventor, Leonardo da Vinci, fall on an even-numbered day?” Yes, Leonardo da Vinci was born on April 15, 1452, which is an odd-numbered day.

Question: “Was Ludwig van Beethoven born in an even day?” Rephrase and expand the question, and respond.

GPT-4 (Correct): “Was the celebrated composer Ludwig van Beethoven born on a day that falls under an even-numbered date?” Ludwig van Beethoven was born on December 17, 1770. Since 17 is an odd number, the answer is **no**, Beethoven was not born on an even day.

Table 13: Examples of correct and incorrect responses by GPT-4 to 1-step RaR on **Even day**. We omit “\n” and leave it as a blank space for better readability.

Question: (original) Was Barack Obama born in an even day? (rephrased) Was the 44th President of the United States, Barack Obama, born on a day of the month that is classified as an even number? Use your answer for the rephrased question to answer the original question.

GPT-4 (Incorrect): No, Barack Obama was not born on an even day. He was born on August 4, 1961.

Question: (original) Was Kanye West born in an even day? (rephrased) Can you tell me if the day of the month on which Kanye West was born falls on an even number? Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): Kanye West was born on June 8, 1977. Since 8 is an even number, the answer to both questions is **yes**.

Question: (original) Was Stephen Hawking born in an even day? (rephrased) Stephen Hawking, did his birth take place on a day that falls into the category of even numbers? Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): Stephen Hawking was born on January 8, 1942. Therefore, the answer to both the original and rephrased questions is **“yes.”**

Question: (original) Was Frida Kahlo born in an even day? (rephrased) Was the famous artist Frida Kahlo born on a day of the month that is an even number? Use your answer for the rephrased question to answer the original question.

GPT-4 (Incorrect): Frida Kahlo was born on July 6, 1907. Thus, she was not born on an even day.

Question: (original) Was Leonardo da Vinci born in an even day? (rephrased) Was the historical figure Leonardo da Vinci, renowned for his artistic and scientific achievements, born on a day that falls under an even number? Use your answer for the rephrased question to answer the original question.

GPT-4 GPT-4 (Correct): Leonardo da Vinci was born on April 15, 1452. 15 is an odd number, so he was **not** born on an even day.

Question: (original) Was the celebrated composer Ludwig van Beethoven born on a day that falls under an even-numbered date? (rephrased) Did the birth of the famous composer Ludwig van Beethoven fall on a day of the month that is considered an even number? Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): Ludwig van Beethoven was born on December 17, 1770. 17 is an odd number, not an even one. Therefore, the answer to both the original and rephrased question is **no**, Ludwig van Beethoven was not born on an even day.

Table 14: Examples of correct and incorrect responses by GPT-4 to the original questions of **Coin flip**.

Question: A coin is heads up. aluino flips the coin. arthor flips the coin. Is the coin still heads up? Flip means reverse. Answer the Yes or No question.

GPT-4 (Incorrect): No

Question: A coin is heads up. jumaimah flips the coin. buity flips the coin. Is the coin still heads up? Flip means reverse. Answer the Yes or No question.

GPT-4 (Incorrect): No

Question: A coin is heads up. kinta does not flip the coin. renit does not flip the coin. Is the coin still heads up? Flip means reverse. Answer the Yes or No question.

GPT-4 (Correct): Yes

Question: A coin is heads up. brandarius does not flip the coin. gitta flips the coin. Is the coin still heads up? Flip means reverse. Answer the Yes or No question.

GPT-4 (Correct): No.

Table 15: Examples of correct and incorrect responses by GPT-4 using 1-step RaR on **Coin flip**.

Question: “A coin is heads up. aluino flips the coin. arthor flips the coin. Is the coin still heads up? Flip means reverse.” Rephrase and expand the question, and respond. Answer the Yes or No question.

GPT-4 (Correct): The coin started out being heads up. Then, Aluino flipped the coin, reversing its side. After that, Arthor also flipped the coin, reversing its side again. Is the coin facing heads up now? Considering the action of flipping the coin means to reverse its side, after two flips, the coin would indeed be back in its original position. So, **yes**, the coin is still heads up.

Question: “A coin is heads up. jumaimah flips the coin. buity flips the coin. Is the coin still heads up? Flip means reverse.” Rephrase and expand the question, and respond. Answer the Yes or No question.

GPT-4 (Correct): The coin is initially facing heads up. Then, it is flipped once by Jumaimah and again by Buity. After these two flips, is the coin still facing heads up? Taking into consideration that flipping the coin means to turn it over to its opposite side, the answer is **Yes**, the coin is still heads up after two flips.

Question: “A coin is heads up. kinta does not flip the coin. renit does not flip the coin. Is the coin still heads up? Flip means reverse.” Rephrase and expand the question, and respond. Answer the Yes or No question.

GPT-4 (Correct): Has the position of the coin, which was initially heads up, changed considering that neither Kinta nor Renit have flipped, implying reversed, the coin? The answer is **yes**, the coin is still heads up.

Question: “A coin is heads up. brandarius does not flip the coin. gitta flips the coin. Is the coin still heads up? Flip means reverse.” Rephrase and expand the question, and respond. Answer the Yes or No question.

GPT-4 (Correct): Rephrased and expanded question: Initially, the coin was facing heads up. The scenario suggests that brandarius doesn’t flip or reverse the coin but gitta does. Considering the definition of the word ‘flip’ as to reverse, we can understand that a coin flipped changes its position from head to tail or vice versa. So, would the coin maintain its initial position of being heads up after gitta flipped it? Answer: **No**.

Table 16: Examples of correct and incorrect responses by GPT-4 using 2-step RaR on **Coin flip**.

Question: (original) A coin is heads up. aluino flips the coin. arthor flips the coin. Is the coin still heads up? Flip means reverse. (rephrased) The initial state of a coin is that it is face up with heads. Then, Aluino performs an action of flipping the coin, which implies reversing its current face from heads to tails. Following this, Arthor takes his turn and also flips the coin. Now, can it be determined whether the coin reverts to its original state of being heads up after these sequential flips? Remember, in this context, flipping pertains to reversing the side that the coin is currently showing. Answer the Yes or No question.

GPT-4 (Correct): Yes.

Question: (original) A coin is heads up. jumaimah flips the coin. buity flips the coin. Is the coin still heads up? Flip means reverse. (rephrased) The initial position of the coin is heads up. Jumaimah proceeds to perform an action on the coin known as a flip, which changes the coin from its current position to its opposite position. Following this, Buity also conducts the same action on the coin as Jumaimah did, which is a flip. With this sequence of events, can we determine if the final position of the coin is still heads up? Remember, the term 'flip' is defined as reversing the current position of the coin. Answer the Yes or No question.

GPT-4 (Correct): Yes.

Question: (original) A coin is heads up. kinta does not flip the coin. renit does not flip the coin. Is the coin still heads up? Flip means reverse. (rephrased) The original question gives us a scenario wherein a coin is initially placed in a heads up position. It also states that two individuals, Kinta and Renit, have not interacted or manipulated the coin, specifically, they have not flipped it. To flip, as described, is to reverse the coin's present state. Considering these points, is the coin's position unchanged and is it still facing heads up? Answer the Yes or No question.

GPT-4 (Correct): Yes.

Question: (original) A coin is heads up. brandarius does not flip the coin. gitta flips the coin. Is the coin still heads up? Flip means reverse. (rephrased) The current state of a coin is that it is facing heads up. An individual named brandarius does not alter this state by flipping the coin. However, another individual named gitta does change the state of the coin by flipping it. A flip in this context refers to reversing the side of the coin that is facing up. Hence, is it possible that following gitta's action on the coin, it remains heads up? Answer the Yes or No question.

GPT-4 (Correct): No.

Table 17: Examples of correct and incorrect responses by GPT-4 to the original questions of **Last letter concatenation (2)**.

Question: Take the last letters of the words in “Beth Joel” and concatenate them.

GPT-4 (Incorrect): “hel”

Question: Take the last letters of the words in “Patrick Lorena” and concatenate them.

GPT-4 (Incorrect): kkna

Question: Take the last letters of the words in “Edgar Bob” and concatenate them.

GPT-4 (Correct): “rb”

Question: Take the last letters of the words in “Alexandra Betty” and concatenate them.

GPT-4 (Incorrect): ara

Question: Take the last letters of the words in “Paul Liz” and concatenate them.

GPT-4 (Incorrect): uliz

Question: Take the last letters of the words in “Sheila Alberto” and concatenate them.

GPT-4 (Correct): “ao”

Table 18: Examples of correct and incorrect responses by GPT-4 using 1-step RaR on **Last letter concatenation (2)**.

Question: “Take the last letters of the words in “Beth Joel” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Correct): Can you merge the last letters from each of the words in the name “Beth Joel” together? What would the resultant combination look like? The last letters from the words “Beth” and “Joel” are “h” and “l” respectively. When we merge them together, the resultant combination is “hl”.

Question: “Take the last letters of the words in “Patrick Lorena” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Correct): Could you please look at each word in the term “Patrick Lorena,” find the last letter of each word, and then join these letters together? Response: The last letters of “Patrick” and “Lorena” are “k” and “a”, respectively. When we concatenate them, we get “ka”.

Question: “Take the last letters of the words in “Edgar Bob” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Correct): Could you please form a new string or series of characters by joining together the final letters from each word in the phrase “Edgar Bob”? The last letters in the words “Edgar” and “Bob” are “r” and “b”, hence when concatenated, it forms “rb”.

Question: “Take the last letters of the words in “Alexandra Betty” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Correct): Could you take the final letters that are from both words in “Alexandra Betty”, combine these letters, and present them together in one word? The response would be “ay”.

Question: “Take the last letters of the words in “Paul Liz” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Incorrect): Can you take the final letters from each word in the phrase “Paul Liz” and put them together sequentially? Yes, by doing so, you get “ulz”.

Question: “Take the last letters of the words in “Sheila Alberto” and concatenate them.” Rephrase and expand the question, and respond.

GPT-4 (Correct): Can you identify the last letter from each word in the name “Sheila Alberto” and then join them together? In the name “Sheila Alberto,” the last letters of each word are “a” from Sheila and “o” from Alberto. If you concatenate or combine these letters, you get “ao”.

Table 19: Examples of correct and incorrect responses by GPT-4 using 2-step RaR on **Last letter concatenation (2)**.

Question: (original) Take the last letters of the words in “Beth Joel” and concatenate them. (rephrased) Can you extract the final letters from both “Beth” and “Joel”, and then combine them to form a new arrangement of letters, maintaining the original instruction to concatenate them? Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The last letter of “Beth” is “h” and the last letter of “Joel” is “l”. Concatenate them, the result is “hl”.

Question: (original) Take the last letters of the words in “Patrick Lorena” and concatenate them. (rephrased) Identify the final letters in each of the words in ‘Patrick Lorena’. After you’ve identified those, combine them all together, in the order they appeared, to form a new word or string of letters. Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The final letters of “Patrick” and “Lorena” are “k” and “a” respectively. When combined together, they form “ka”.

Question: (original) Take the last letters of the words in “Edgar Bob” and concatenate them.” Rephrase and expand the question, and respond. (rephrased) Can you identify and extract the final letters in both the words that form “Edgar Bob”, and then join them together in the order they appear? Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The last letters in the words “Edgar Bob” are “r” and “b”. Concatenating them in the order they appear would be “rb”.

Question: (original) Take the last letters of the words in “Alexandra Betty” and concatenate them. (rephrased) To answer the question, you would need to look at the words “Alexandra” and “Betty”. Identify the last letter in each of these words. Once you have these last letters, join them together in the order they appear in the original words. The question is asking you to perform these actions. Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The last letter of “Alexandra” is “a” and the last letter of “Betty” is “y”. When these are concatenated, or joined together, the result is “ay”.

Question: (original) Take the last letters of the words in “Paul Liz” and concatenate them. (rephrased) Identify the last letters of each word in the phrase “Paul Liz”. After identifying, combine these letters together to form a new word or string. This process is known as concatenation. In other words, you will merge the last characters from ‘Paul’ and ‘Liz’ without any spaces between them. Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The last letters of “Paul” and “Liz” are “l” and “z” respectively. Concatenating these letters would form “lz”.

Question: (original) Take the last letters of the words in “Sheila Alberto” and concatenate them. (rephrased) Identify the final letters in each of the words ‘Sheila’ and ‘Alberto’. After recognizing these, join or link them together in the same order they appear. This process is referred to as concatenation. Use your answer for the rephrased question to answer the original question.

GPT-4 (Correct): The last letter of “Sheila” is “a” and the last letter of “Alberto” is “o”. So, when concatenated, the resulting output is “ao”.

Table 20: Examples of correct and incorrect responses by GPT-4 to the original questions of **Sports**.

Question: Is the following sentence plausible? “Javier Zanetti earned a direct kick” Answer the Yes or No question.
GPT-4 (Incorrect): No
Question: Is the following sentence plausible? “Jayson Tatum nutmegged the defender” Answer the Yes or No question.
GPT-4 (Incorrect): Yes
Question: Is the following sentence plausible? “Sean Couturier passed the puck in the Stanley Cup” Answer the Yes or No question.
GPT-4 (Correct): Yes
Question: Is the following sentence plausible? “Tuukka Rask hit a double” Answer the Yes or No question.
GPT-4 (Correct): No
Question: Is the following sentence plausible? “Sam Darnold struck out” Answer the Yes or No question.
GPT-4 (Incorrect): Yes

Table 21: Examples of correct and incorrect responses by GPT-4 using One-step RaR on **Sports**.

Question: ‘Is the following sentence plausible? “Javier Zanetti earned a direct kick”’ Rephrase and expand the question, and respond. Answer the Yes or No question.
GPT-4 (Correct): “Is it plausible or possible that Javier Zanetti received a direct kick during a soccer game?” Yes, it is plausible.
Question: ‘Is the following sentence plausible? “Jayson Tatum nutmegged the defender”’ Rephrase and expand the question, and respond. Answer the Yes or No question.
GPT-4 (Incorrect): Could it be possible that Jayson Tatum managed to play a nutmeg move on a defender during a basketball game? Yes, it’s plausible as nutmegging is a commonly used trick in basketball and Jayson Tatum is known for his skills and tricks in the game.
Question: ‘Is the following sentence plausible? “Sean Couturier passed the puck in the Stanley Cup”’ Rephrase and expand the question, and respond. Answer the Yes or No question.
GPT-4 (Correct): Is it possible for the statement “Sean Couturier maradona’d the defender” to make sense or be meaningful? In context of sports language, particularly soccer, the term “maradona’d” could be used to describe a soccer player performing a maneuver similar to that of Diego Maradona, a renowned soccer player. Sean Couturier is a professional ice hockey player, and although the sports are different, it is plausible that he could carry out a maneuver in hockey analogous to one performed by Maradona in soccer, assuming “maradona’d” means he skillfully bypassed or tricked the defender in some way. So, yes, the sentence could be plausible.
Question: ‘Is the following sentence plausible? “Tuukka Rask hit a double”’ Rephrase and expand the question, and respond. Answer the Yes or No question.
GPT-4 (Correct): Is it possible that Tuukka Rask, a professional ice hockey player, could have hit a double in a game? No, Tuukka Rask is a professional ice hockey goaltender for the Boston Bruins. He would not be involved in a sport where he would hit a double, such as baseball. The phrase “hitting a double” isn’t applicable to ice hockey, making the sentence implausible.
Question: ‘Is the following sentence plausible? “Sam Darnold struck out”’ Rephrase and expand the question, and respond. Answer the Yes or No question.
GPT-4 (Correct): Is it reasonable or believable to say “Sam Darnold struck out”? No, it’s not plausible because Sam Darnold is a known professional football player, and the term “struck out” is used in baseball, not football.

Table 22: Examples of correct and incorrect responses by GPT-4 using Two-step RaR on **Sports**.

Question: (original) Is the following sentence plausible? “Javier Zanetti earned a direct kick” (rephrased) Is it believable or possible that Javier Zanetti secured a direct free kick in a match? Use your answer for the rephrased question to answer the original question. Answer the Yes or No question.

GPT-4 (Correct): Yes

Question: (original) Is the following sentence plausible? “Jayson Tatum nutmegged the defender” (rephrased) Is it believable that Jayson Tatum, a professional basketball player, could have performed a move where he passed the ball through a defender’s legs, also known as a ‘nutmeg’? Use your answer for the rephrased question to answer the original question. Answer the Yes or No question.

GPT-4 (Incorrect): Yes

Question: (original) Is the following sentence plausible? “Sean Couturier passed the puck in the Stanley Cup” (rephrased) Could it be possible that Sean Couturier passed the puck during a game in the Stanley Cup tournament? Use your answer for the rephrased question to answer the original question. Answer the Yes or No question.

GPT-4 (Correct): Yes

Question: (original) Is the following sentence plausible? “Tuukka Rask hit a double” (rephrased) Is it feasible that Tuukka Rask, presumably known for a non-baseball sport, hit a double in a baseball game? Use your answer for the rephrased question to answer the original question. Answer the Yes or No question.

GPT-4 (Correct): No

Question: (original) Is the following sentence plausible? “Sam Darnold struck out” (rephrased) Is it believable that Sam Darnold has struck out in a game? Please provide a detailed explanation supporting your answer without omitting any information from the initial question. Use your answer for the rephrased question to answer the original question. Answer the Yes or No question.

GPT-4 (Correct): No

References

- ALLEN-ZHU, Z. and LI, Y. (2023). Physics of language models: Part 3.2, knowledge manipulation.
- ATEIA, S. and KRUSCHWITZ, U. (2023). Is chatgpt a biomedical expert?—exploring the zero-shot performance of current gpt models in biomedical tasks. *arXiv preprint arXiv:2306.16108* .
- BENCH AUTHORS, B. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research* .
- BOZKURT, A. (2023). Generative artificial intelligence (ai) powered conversational educational agents: The inevitable paradigm shift. *Asian Journal of Distance Education* **18**.
- BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A. ET AL. (2020). Language models are few-shot learners. *Advances in neural information processing systems* **33** 1877–1901.
- CHIANG, W.-L., LI, Z., LIN, Z., SHENG, Y., WU, Z., ZHANG, H., ZHENG, L., ZHUANG, S., ZHUANG, Y., GONZALEZ, J. E., STOICA, I. and XING, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- DRUCKMAN, J. N. (2001). The implications of framing effects for citizen competence. *Political behavior* **23** 225–256.
- ERVING, G. (1974). Frame analysis: An essay on the organization of experience.
- FORTES, A. (2023). simple-dataset-generation.
- GAO, L., DAI, Z., PASUPAT, P., CHEN, A., CHAGANTY, A. T., FAN, Y., ZHAO, V., LAO, N., LEE, H., JUAN, D.-C. ET AL. (2023). Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- HUANG, J., CHEN, X., MISHRA, S., ZHENG, H. S., YU, A. W., SONG, X. and ZHOU, D. (2023). Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798* .
- JUNG, J., QIN, L., WELLECK, S., BRAHMAN, F., BHAGAVATULA, C., LE BRAS, R. and CHOI, Y. (2022). Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates.
- KIM, G., BALDI, P. and MCALLEER, S. (2023). Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491* .
- KOJIMA, T., GU, S. S., REID, M., MATSUO, Y. and IWASAWA, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems* **35** 22199–22213.
- LIGHTMAN, H., KOSARAJU, V., BURDA, Y., EDWARDS, H., BAKER, B., LEE, T., LEIKE, J., SCHULMAN, J., SUTSKEVER, I. and COBBE, K. (2023). Let's verify step by step.
- LIU, A., WU, Z., MICHAEL, J., SUHR, A., WEST, P., KOLLER, A., SWAYAMDIPTA, S., SMITH, N. A. and CHOI, Y. (2023). We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399* .
- LU, P., BANSAL, H., XIA, T., LIU, J., LI, C., HAJISHIRZI, H., CHENG, H., CHANG, K.-W., GALLEY, M. and GAO, J. (2023). Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts.
- MADAAN, A., TANDON, N., GUPTA, P., HALLINAN, S., GAO, L., WIEGREFFE, S., ALON, U., DZIRI, N., PRABHUMOYE, S., YANG, Y. ET AL. (2023). Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651* .

- MUKHERJEE, S., MITRA, A., JAWAHAR, G., AGARWAL, S., PALANGI, H. and AWADALLAH, A. (2023). Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707* .
- NADEEM, M., BETHKE, A. and REDDY, S. (2021). Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.
- OPENAI (2022). Best practices for prompt engineering with openai api. <https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api>
- OPENAI (2023). Gpt-4 technical report.
- PAN, L., SAXON, M., XU, W., NATHANI, D., WANG, X. and WANG, W. Y. (2023). Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188* .
- PAWELCZYK, M., NEEL, S. and LAKKARAJU, H. (2023). In-context unlearning: Language models as few shot unlearners.
- POESIA, G., POLOZOV, O., LE, V., TIWARI, A., SOARES, G., MEEK, C. and GULWANI, S. (2022). Synchromesh: Reliable code generation from pre-trained language models. *arXiv preprint arXiv:2201.11227*
- PRESS, O., ZHANG, M., MIN, S., SCHMIDT, L., SMITH, N. A. and LEWIS, M. (2022). Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350* .
- PRYZANT, R., ITER, D., LI, J., LEE, Y. T., ZHU, C. and ZENG, M. (2023). Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495* .
- RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., SUTSKEVER, I. ET AL. (2019). Language models are unsupervised multitask learners. *OpenAI blog* 1 9.
- REYNOLDS, L. and McDONELL, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI EA '21, Association for Computing Machinery, New York, NY, USA.
- SARAVIA, E. (2022). Prompt Engineering Guide. <https://github.com/dair-ai/Prompt-Engineering-Guide> .
- SCHICK, T. and SCHÜTZE, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.
- SHAIKH, O., ZHANG, H., HELD, W., BERNSTEIN, M. and YANG, D. (2022). On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. *arXiv preprint arXiv:2212.08061* .
- SHAO, Z., GONG, Y., SHEN, Y., HUANG, M., DUAN, N. and CHEN, W. (2023). Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618* .
- SHINN, N., CASSANO, F., BERMAN, E., GOPINATH, A., NARASIMHAN, K. and YAO, S. (2023). Reflexion: Language agents with verbal reinforcement learning.
- SORENSEN, T., ROBINSON, J., RYTTING, C., SHAW, A., ROGERS, K., DELOREY, A., KHALIL, M., FULDA, N. and WINGATE, D. (2022). An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- STECHLY, K., MARQUEZ, M. and KAMBHAMPATI, S. (2023). Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems.

- TALMOR, A., HERZIG, J., LOURIE, N. and BERANT, J. (2019). Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- TOUVRON, H., MARTIN, L., STONE, K., ALBERT, P., ALMAHAIROU, A., BABAEI, Y., BASHLYKOV, N., BATRA, S., BHARGAVA, P., BHOSALE, S. ET AL. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* .
- TURPIN, M., MICHAEL, J., PEREZ, E. and BOWMAN, S. R. (2023). Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting.
- WANG, J., SHI, E., YU, S., WU, Z., MA, C., DAI, H., YANG, Q., KANG, Y., WU, J., HU, H. ET AL. (2023). Prompt engineering for healthcare: Methodologies and applications. *arXiv preprint arXiv:2304.14670* .
- WANG, X., WEI, J., SCHUURMANS, D., LE, Q., CHI, E., NARANG, S., CHOWDHERY, A. and ZHOU, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* .
- WEI, J., WANG, X., SCHUURMANS, D., BOSMA, M., XIA, F., CHI, E., LE, Q. V., ZHOU, D. ET AL. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **35** 24824–24837.
- WELLECK, S., LU, X., WEST, P., BRAHMAN, F., SHEN, T., KHASHABI, D. and CHOI, Y. (2022). Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053* .
- WHITE, J., FU, Q., HAYS, S., SANDBORN, M., OLEA, C., GILBERT, H., ELNASHAR, A., SPENCER-SMITH, J. and SCHMIDT, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* .
- WILLIAMS, A., NANGIA, N. and BOWMAN, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji and A. Stent, eds.). Association for Computational Linguistics, New Orleans, Louisiana.
- YAN, H., SRIVASTAVA, S., TAI, Y., WANG, S. I., YIH, W.-T. and YAO, Z. (2023). Learning to simulate natural language feedback for interactive semantic parsing. *arXiv preprint arXiv:2305.08195* .
- YANG, K., TIAN, Y., PENG, N. and KLEIN, D. (2022). Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.
- YAO, S., YU, D., ZHAO, J., SHAFRAN, I., GRIFFITHS, T. L., CAO, Y. and NARASIMHAN, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601* .
- YU, W., ZHANG, Z., LIANG, Z., JIANG, M. and SABHARWAL, A. (2023). Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002* .
- ZHANG, M., PRESS, O., MERRILL, W., LIU, A. and SMITH, N. A. (2023a). How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534* .
- ZHANG, Z., ZHANG, A., LI, M., ZHAO, H., KARYPIS, G. and SMOLA, A. (2023b). Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923* .
- ZHOU, D., SCHÄRLI, N., HOU, L., WEI, J., SCALES, N., WANG, X., SCHUURMANS, D., CUI, C., BOUSQUET, O., LE, Q. V. ET AL. (2022a). Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

ZHOU, P., MADAAN, A., POTHARAJU, S. P., GUPTA, A., MCKEE, K. R., HOLTZMAN, A., PUJARA, J., REN, X., MISHRA, S., NEMATZADEH, A., UPADHYAY, S. and FARUQUI, M. (2023). How far are large language models from agents with theory-of-mind?

ZHOU, Y., MURESANU, A. I., HAN, Z., PASTER, K., PITIS, S., CHAN, H. and BA, J. (2022b). Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910* .