Language Models Benefit from Preparation with Elicited Knowledge

Jiacan Yu, Hannah Y. An, and Lenhart K. Schubert

University of Rochester, Rochester NY 14620, USA
jyu48@u.rochester.edu
{yan2,schubert}@cs.rochester.edu

Abstract. The zero-shot chain of thought (CoT) approach[6] is often used in question answering (QA) by language models (LMs) for tasks that require multiple reasoning steps, typically enhanced by the prompt "Let's think step by step." However, some QA tasks hinge more on accessing relevant knowledge than on chaining reasoning steps. We introduce a simple general prompting technique, called PREP, that involves using two instances of LMs: the first (LM1) generates relevant information, and the second (LM2) answers the question based on this information. PREP is designed to be general and independent of the user's domain knowledge, making it applicable across various QA tasks without the need for specialized prompt engineering. To evaluate the effectiveness of our prompting method, we create a dataset of 100 binary-choice questions, derived from an extensive schematic dataset on artifact parts and material composition. These questions ask which of two artifacts is less likely to share materials with another artifact. Such questions probe the LM's knowledge of shared materials in the part structure of different artifacts. We test our method on our dataset and three published commonsense reasoning datasets. The average accuracy of our method is consistently higher than that of all the other tested methods across all the tested datasets.

Keywords: Prompt engineering · Question answering · Knowledge elicitation · Large language models · Natural language processing.

1 Introduction

Zero-shot Chain of Thought (CoT) [6] has emerged as a popular technique in question answering (QA) by language models (LMs), particularly for tasks that require multiple reasoning steps. This approach leverages the prompt "Let's think step by step" to enhance the model's capacity for handling complex reasoning processes. However, not all QA tasks benefit from this method. Kojima et al. [6] show that the zero-shot CoT approach does not offer any performance improvements over the zero-shot baseline on commonsense reasoning datasets, such as CommonsenseQA [9] and StrategyQA [5]. This suggests that many tasks rely more on the model's ability to effectively access and utilize relevant knowledge than on step-by-step reasoning.

To address this limitation, we introduce a simple general prompting technique, PREP, that involves using two instances of a LM. In our approach, the first instance (LM1) generates information relevant to answering the question, and the second instance (LM2) uses this information to answer the question. This dual-instance prompting method is designed to be both general and user-independent, making it applicable to a wide range of QA tasks without requiring specialized prompt engineering.

An important aspect of our design is the decision to copy the information from LM1 and send it to LM2 as user input, rather than appending the question to LM1 after it retrieves the information. This is inspired by a commonly observed phenomenon with LMs: They seem to treat user inputs as more credible than their own outputs; thus an LM can be induced to take general knowledge elicited from it more seriously by feeding that knowledge to a new instance of the LM.

To evaluate the effectiveness of our prompting method, we created a curated dataset of 100 binary-choice questions. These questions are derived from an extensive schematic dataset that includes detailed information on the parts and material composition of various artifacts. The binary-choice questions are designed to test the LMs' ability to determine which of two artifacts is less likely to share materials with another artifact, thus challenging the models to utilize detailed partonomic and material knowledge. While this is a subdomain of commonsense knowledge, it is an important one, since much of our knowledge about everyday objects concerns their part structure and material makeup. For example, if someone were asked to describe the essential characteristics of a pencil, the answer would probably include mention of the wooden shaft, graphite core, and protruding graphite point (besides usage, of course). Some depth of knowledge is required to recognize that a pencil shares some of its material composition with lithum-ion batteries – whose anodes are typically coated with graphite.

In our experiments, we pose these questions, as well as 200 randomly selected questions from each of CommonsenseQA [9], StrategyQA [5], and OpenBookQA [8], to three different LMs with parameter sizes ranging from 14B to 35B. The results demonstrate the efficacy of our dual-instance prompting approach, showing improvement in accuracy compared to direct questioning, zero-shot CoT, and single instance prompting methods. To the best of our knowledge, our method is one of the simplest approaches for improving commonsense QA performance.

2 Related Work

2.1 Related Work on Generic User-independent LM Prompting

The zero-shot Chain of Thought method, introduced by Kojima et al. [6], is a variant of the CoT prompting concept, originally designed for few-shot learning. Zero-shot CoT involves a two-stage process: First, it elicits step-by-step reasoning from the model using a general trigger phrase such as "Let's think step by step," and second, it extracts the final answer from the generated reasoning.

This approach allows the model to perform complex multi-step reasoning tasks by mimicking the human thought process, improving performance across various benchmarks, including arithmetic and reasoning tasks.

Our method and zero-shot CoT are both general and user-knowledge independent, meaning they can be directly applied to a wide range of tasks without requiring specific prompt engineering for each class of tasks. However, while zero-shot CoT focuses on enhancing the model's ability to answer questions that are expected to require step-wise inference of intermediate results (for example, to derive numerical answers), our method aims to improve performance on tasks that require extensive factual knowledge about everyday objects.

Similar to zero-shot CoT, Plan-and-Solve (PS) prompting, proposed by Wang et al. [11], is another general prompting method. It replaces the trigger phrase "Let's think step by step" with a longer phrase that instructs the LM to devise a plan and execute the plan. This approach is intended to address some limitations of zero-shot CoT, such as calculation errors, missing-step errors, and semantic misunderstandings.

2.2 Prompting Methods that Require Task-Specific Design

Perhaps the method most similar to our own is generated-knowledge prompting [7]. This prompts for knowledge relevant to a multi-choice question, after providing 5 hand-engineered examples of question-knowledge pairs for each of the 4 commonsense domains the question is taken from: NumerSense (focused on numeric attributes, such as numbers of certain parts of an entity, or the freezing temperature of a certain liquid); CSQA (crowd-sourced questions and answers inspired by ConceptNet pairs of triples such as *stream AtLocation river, river AtLocation bridge*), CSQA2 (also based on ConceptNet-inspired questions, obtained via a QA game); and QASC (grade school science questions, e.g., about using wind to generate electricity). Our method differs in that it does not require engineering of prompts aimed at classes of questions. In a somewhat similar vein, Wang, Deng, & Sun [10] developed a Transformer-based iterative prompter, aimed at multi-hop reasoning, trained on multi-hop datasets obtained by crowdsourcing. Again, our approach is less demanding, not requiring training (or downloading) of a separate prompting LM.

Some other works in the field make use of more sophisticated reasoning processes to improve their performance, for example, tree of thoughts (ToT) [12] and graph of thoughts (GoT) [3]. These two methods require significant human intervention to design task-specific mechanisms to shape the reasoning process into a tree or graph structure. The ToT framework enhances problem-solving abilities of LMs by structuring the reasoning process as a tree. Each node in this tree represents a partial solution. To apply this method to a task, a user needs to determine a general strategy for decomposing intermediate thought steps based on the task properties. The method also requires a thought generator, generating potential next partial solutions, and a state evaluator, evaluating the effectiveness of the partial solutions. Then search algorithms like breadth-first search

(BFS) and depth-first search (DFS) are used to explore and refine potential solutions. ToT has demonstrated improvements in solving complex tasks such as the Game of 24, creative writing, and mini crosswords.

Similarly, the GoT framework extends the idea of using a structured reasoning process by modeling this process as a directed graph. GoT also requires task-specific designs to define how thoughts are combined, transformed, and evaluated, making it a powerful but complex method for guiding LLMs through problem-solving tasks. We do not expect our simple "hands-off" method to be competitive against ToT or GoT in complex tasks, but it can provide a very simple way of boosting LM performance in commonsense question answering.

3 Our PREP Approach

Our dual-instance prompting method, called PREP, is a simple technique that uses knowledge elicited from one instance of a LM to PREPare another instance of the LM for a QA task:

> **Step 1. Knowledge Elicitation**: We present the question to the first instance of the LM (LM1) and ask it to provide information relevant to answering the question.
> **Step 2. Knowledge Transfer**: The response from LM1 is copied and sent to the second instance of the LM (LM2) along with the question. LM2 now answers the question.

This method is designed to be both general and user-knowledge independent:

- **Generality**: Our method is versatile and can be applied to a wide range of QA domains and tasks.
- **User-Knowledge Independence**: Our method does not require further prompt engineering, training, or other customizations that require the user's knowledge of the task domain.

Fig. 1 provides a visual representation of our PREP approach. Note that user-knowledge independence is orthogonal to the number of instances involved in a prompting method. In our experiments, we also test a user-knowledge dependent version of our dual-instance method.

In practice, our PREP prompting method offers a simple performance-enhancing method for querying LMs, without requiring extensive time investment in crafting specialized prompts. While specialized prompts can outperform our method in certain tasks, they do not always do so, as demonstrated by some of our experiments with question-specific prompts. This makes it attractive for scenarios where quick deployment and broad applicability are prioritized.

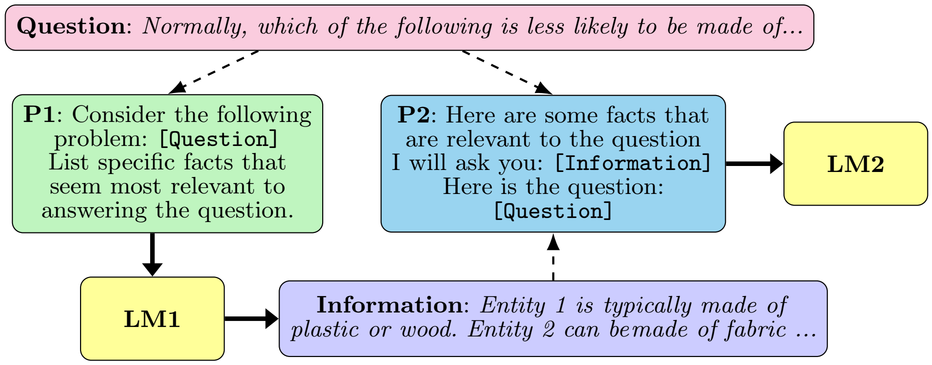

Fig. 1. Overview of our PREP approach, a dual-instance prompting method using general and user-knowledge independent prompts (slightly simplified here). The first prompt (P1), combined with the question, directs LM1 to provide specific facts in its response that seem relevant to answering the question. The collected information is then combined with the original question and used to prompt LM2.

4 Experimental Settings

4.1 Dataset Creation

To evaluate the effectiveness of our prompting method, we designed a nontrivial QA task, asking LMs binary-choice questions about shared materials between objects. This task challenges LMs to consider detailed information about the possible materials that objects can be made of. To construct our dataset, we began with an extensive schematic dataset detailing the parts and material composition of primarily man-made objects. From this dataset, we identified triples of objects (O_A, O_B, O_C) where objects O_A and O_B share a common material, while object O_C does not share any material with O_B. We then manually selected 100 triples from this set and corrected any inaccuracies to generate a curated set of test questions for our new dataset. The questions (**Q**) are formatted as follows:

> Normally, which of the following is less likely to be at least partially made of a material that is a constituent of O_B?
> a) O_A b) O_C

This phrasing is preferable to asking which artifact is *more* likely to share materials with the target object, as it avoids the connotation that the amount of shared material is quantitatively significant for the correct answer. To keep the test fair and avoid any bias, we ensured there were an equal number of questions with the correct answers being 'a)' and 'b)'.

4.2 Experiments

We evaluate our PREP method by comparing it to various prompting techniques. The comparison includes methods with different levels of reliance on user knowledge, ranging from direct questioning, Plan and Solve (PS) prompting, and zero-shot Chain of Thought (CoT), to those incorporating user-knowledge dependent prompts tailored to the question. Specifically, in the user-knowledge dependent prompts, the LMs are asked to list the parts and materials of the objects mentioned in **Q** before providing an answer, rather than offering general information relevant to **Q**. We also examine other single-instance methods that utilize one or two prompts, both with and without these user-knowledge dependent prompts. To further investigate the impact of reinitializing the LM, we include variations where the information elicited from the LM is copied into the subsequent prompts within the same LM instance. Additionally, we evaluate the performance of our PREP method on three existing datasets–CommonsenseQA, StrategyQA, and OpenBookQA–to assess its generalizability across different types of commonsense reasoning problems. The specific prompts used in each of these methods are detailed in the following subsection.

Experiments are conducted using three language models : Phi-3 (14B) [1], Aya 23 (35B) [2], and Command R v0.1 (35B) [4].[1] These models are quantized to 4-bit or 3-bit due to hardware constraints.[2] The temperature is set to 0 in our experiments.

4.3 Prompt Details

The exact prompts tested are listed here:

Dual-Instance

`User's Message to LM1:`

> Consider the following binary-choice problem:
> (*Insert the question* $\boldsymbol{Q}$)
> Please list specific facts that seem most relevant to answering the question. Do not answer the question, and do not include anything other than the list in your response.

`User's Message to LM2:`

> Here are some facts that are relevant to the question I will ask you:
> (*Insert the response of LM1*)
> Here is the question:
> (*Insert the question* $\boldsymbol{Q}$)

[1] The three language models are deployed offline using Ollama, available at https://ollama.com/library.
[2] Experiments are conducted using a single RTX4090. Phi-3 is loaded using Q4_0 4-bit quantization. Aya 23 and Command R are loaded using Q3_K_M 3-bit quantization.

Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

The following is an example from our experiment.

`User's Message to LM1:`

Consider the following binary-choice problem:

> Normally, which of the following is less likely to be at least partially made of a material that is a constituent of a pop-up mosquito net?
> a) golf ball b) clear vase

Please list specific facts that seem most relevant to answering the question. Do not answer the question, and do not include anything other than the list in your response.

`Response from LM1:`

1. Composition of pop-up mosque nets: Pop-up mosquito nets are typically made from materials like polyethylebelene (PE), polyester or nylon mesh fabric, which is lightweight and durable for outdoor use.
2. Golf ball composition: A standard golf ball consists of a solid rubber core surrounded by several layers of urethane-based plastic material with a hard outer shell made from surlyn (a type of ionomer resin).
3. Composition of clear vase: Clear vases are usually made from glass or sometimes acrylic, which is not similar to the materials used in pop-up mosquito nets.

`User's Message to LM2:`

Here are some facts that are relevant to the question I will ask you:

1. Composition of pop-up mosque nets: Pop-up mosquito nets are typically made from materials like polyethylebelene (PE), polyester or nylon mesh fabric, which is lightweight and durable for outdoor use.
2. Golf ball composition: A standard golf ball consists of a solid rubber core surrounded by several layers of urethane-based plastic material with a hard outer shell made from surlyn (a type of ionomer resin).
3. Composition of clear vase: Clear vases are usually made from glass or sometimes acrylic, which is not similar to the materials used in pop-up mosquito nets.

Here is the question:

> Normally, which of the following is less likely to be at least partially made of a material that is a constituent of a pop-up mosquito net?
> a) golf ball b) clear vase

Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response:

`Response from LM2:`

Based on the given facts, it's less likely that a golf ball would be made from materials used in pop-up mosquito nets since they are typically composed of rubber and plastic with an ionomer resin outer shell. Clear vases, however, could potentially have some similarities to pop-up mosquito nets if the clear vase is made from a type of mesh fabric like polyester or nylon for decorative purposes. However, it's more common for them to be made from glass or acrylic which are not constituents of pop-up mosquito nets. Therefore, my answer is b).

Direct Asking

`User's Message:`

(*Insert the question* ***Q***)
Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Zero-shot CoT

`User's Message:`

(*Insert the question* ***Q***)
Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

`Assistant's Answer:`

Let's think step by step. (*LM continues generation from here*)

Plan and Slove (PS) Prompting

`User's Message:`

(*Insert the question* ***Q***)
Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

`Assistant's Answer:`

Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step. (*LM continues generation from here*)

Single-Instance, Two-Prompts

`User's First Message:`

Consider the following binary-choice problem:
(*Insert the question* ***Q***)
Please list specific facts that seem most relevant to answering the question. Do not answer the question, and do not include anything other than the list in your response.

`User's Second Message:`

Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Single-Instance, One-Prompt

`User's Message:`

(*Insert the question* ***Q***)
Before giving your answer, please first list specific facts that seem most relevant to answering the question.

Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Single-Instance, Information Copied

`User's First Message:`

Consider the following binary-choice problem:
(*Insert the question* ***Q***)
Please list specific facts that seem most relevant to answering the question. Do not answer the question, and do not include anything other than the list in your response.

`User's Second Message:`

Here are some facts that are relevant to the question:
(*Insert the response from First Prompt*)
Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Dual-Instance, User-Knowledge Dependent

`User's Message to LM1:`

List the parts of O_A, O_B, and O_C, as well as the material of each part.

`User's Message to LM2:`

Here are some facts that are relevant to the question I will ask you:
(*Insert the response of LM1*)
Here is the question:
(*Insert the question* ***Q***)
Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Single-Instance, Two-Prompts, User-Knowledge Dependent

`User's First Message:`

List the parts of O_A, O_B, and O_C, as well as the material of each part.

`User's Second Message:`

(*Insert the question* $\boldsymbol{Q}$)
Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Single-Instance, One-Prompt, User-Knowledge Dependent

`User's Message:`

(*Insert the question* $\boldsymbol{Q}$)
Before giving your answer, please first list the parts of O_A, O_B, and O_C, as well as the material of each part.
Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

Single-Instance, Information Copied, User-Knowledge Dependent

`User's First Message:`

List the parts of O_A, O_B, and O_C, as well as the material of each part.

`User's Second Message:`

Here are some facts that are relevant to the question I will ask you:
(*Insert the response from First Prompt*)
Here is the question:
(*Insert the question* $\boldsymbol{Q}$)
Consider the question based on common sense and the information. Clearly indicate the answer by saying 'my answer is a)' or 'my answer is b)' at the end of your response.

5 Results and Discussion

The average accuarcy of our dual-instance method is consistently higher than or equal to that of all other tested methods across all the tested datasets. Particularly, the average accuracy of our method surpasses zero-shot CoT by approximately 5% on CSQA, StrategyQA, and OpenBookQA. When comparing our method with the experiment *1 inst. info. copied*, we observe that the repetition of information in the context negatively impacts accuracy. This demonstrates the necessity of creating another instance of LM. Furthermore, when comparing our method with the last two single-instance methods listed in the tables, we find that although all these methods aim to provide similar contextual information to assist the LMs in answering the questions, the accuracy improves when the relevant information is provided by the user rather than the model,

Table 1. Accuracy of the various prompting methods tested on our curated dataset. Zero-shot CoT (ZS CoT; second line in the table) is set as the baseline. Since multiple models are tested, we calculate the difference between the average accuracy of each prompting method and the average accuarcy of ZS CoT to help examining the results. If the average accuracy is higher than the average accuracy of CoT, the Avg. Diff is blue. Otherwise it is red.

Elicitation	# Inst.	# Messages	Copy	**Phi-3**	**Aya**	**Command-R**	Avg. Diff
Direct Asking	single	1	-	67±4.7%	55±5%	54±5%	-6.67
ZS CoT	single	1	-	**78±4.1%**	59±4.9%	59±4.9%	0.00
PS prompting	single	1	-	72±4.5%	63±4.8%	66±4.7%	1.67
	single	1	-	73±4.4%	68±4.7%	56±5%	0.33
User Knowledge Indep.	single	2	no	71±4.5%	62±4.9%	46±5%	-5.67
	single	2	yes	70±4.6%	64±4.8%	58±4.9%	-1.33
	dual	2	yes	70±4.6%	67±4.7%	66±4.7%	2.33
	single	1	-	65±4.8%	58±4.9%	64±4.8%	-3.00
User Knowledge Dep.	single	2	no	60±4.9%	60±4.9%	62±4.9%	-4.67
	single	2	yes	61±4.9%	68±4.7%	60±4.9%	-2.33
	dual	2	yes	62±4.9%	**71±4.5%**	**74±4.4%**	3.67

Table 2. Accuracy of the various prompting methods tested on CommonsenseQA

Experiment	Phi-3	Aya	Command-R	Avg. Diff
direct asking	76±3%	81.5±2.7%	72.5±3.2%	2.83
CoT	74±3.1%	76.5±3%	71±3.2%	0.00
PS prompting	74.5±3.1%	73.5±3.1%	70±3.2%	-1.17
2 inst.	**80.5±2.8%**	84±2.6%	75±3.1%	6.00
1 inst. info. copied	76.5±3%	82.5±2.7%	**76.5±3%**	4.67
1 inst. 2 messages	69.5±3.3%	82.5±2.7%	75.5±3%	2.00
1 inst. 1 message	76.5±3%	**84.5±2.6%**	75±3.1%	4.83

Table 3. Accuracy of the various prompting methods tested on StrategyQA

Experiment	Phi-3	Aya	Command-R	Avg. Diff
direct asking	63.5±3.4%	73±3.1%	76±3%	-2.33
CoT	62.5±3.4%	76.5±3%	**80.5±2.8%**	0.00
PS prompting	66.5±3.3%	79±2.9%	78.5±2.9%	1.50
2 inst.	**74±3.1%**	79.5±2.9%	80±2.8%	4.67
1 inst. info. copied	65.5±3.4%	**81.5±2.7%**	79.5±2.9%	2.33
1 inst. 2 messages	70±3.2%	80.5±2.8%	78.5±2.9%	3.17
1 inst. 1 message	68±3.3%	81±2.8%	79±2.9%	2.83

showing the importance of user-provided context in QA tasks that depend on factual knowledge. When we test phi3 on StragegyQA, we notice that the number of questions that the model cannot provide a definite answer is 12 under

Table 4. Accuracy of the various prompting methods tested on OpenBookQA

Experiment	Phi-3	Aya	Command-R	Avg. Diff
direct asking	87±2.4%	85±2.5%	**85.5±2.5%**	5.67
CoT	86±2.5%	75±3.1%	79.5±2.9%	0.00
PS prompting	**87.5±2.3%**	73.5±3.1%	80±2.8%	0.17
2 inst.	87±2.4%	85±2.5%	**85.5±2.5%**	5.67
1 inst. info. copied	82.5±2.7%	84±2.6%	81±2.8%	2.33
1 inst. 2 messages	84±2.6%	85±2.5%	74±3.1%	0.83
1 inst. 1 message	87±2.4%	**85.5±2.5%**	84±2.6%	5.33

CoT prompting. When we use our method, this number decrease to 3, showing that our method improves the confidence of LMs.[3][4]

6 Conclusions

In this study, we introduced a dual-instance prompting method, PREP, aimed at improving the performance of LMs on QA tasks that require accessing relevant knowledge rather than following step-by-step reasoning. By employing two instances of LMs, one for information elicitation and the other for answering the question based on the elicited information, we demonstrate a notable improvement in accuracy compared to direct questioning and zero-shot CoT approaches across different models. Our method is general and user-independent, applicable across various QA tasks without the need for specialized prompt engineering. Experimental results show that our dual-instance methods outperform single-instance methods. This underscores the value of (ostensibly) user-provided context in enhancing LM performance.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

[3] We also performed the experiments using models with smaller context length (4k). However, they sometimes output responses indicating that they are not able to handle the whole context. For example, in the *1 inst. 2 prompts* experiment with the OpenBookQA dataset, the 4k-context version of Phi-3 says: "...Without knowing the specific details of the question, I cannot provide an accurate answer. However, once you present the complete question, I will analyze..." To isolate the effects of the insufficiency of context length, we decide to carry on our experiment using only models with context length more than 4k.

[4] We also test these prompting methods on smaller language models (7B to 8B) on our dataset. To our surprise, none of the prompting methods we test can improve the performance of these models to a level differentiable from randomly guessing. We find that even if we provide very strong help to them, that is, directly provide the material information from our schematic dataset to these LMs, their accuracy in answering **Q** is scarcely above random guessing; so we believe it is very difficult to find a simple prompting method that works well on relatively small LMs.

References

1. Abdin, M., Jacobs, S.A., Awan, A.A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, Q., Cai, M., Mendes, C.C.T., Chen, W., Chaudhary, V., Chen, D., Chen, D., Chen, Y.C., Chen, Y.L., Chopra, P., Dai, X., Giorno, A.D., de Rosa, G., Dixon, M., Eldan, R., Fragoso, V., Iter, D., Gao, M., Gao, M., Gao, J., Garg, A., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R.J., Huynh, J., Javaheripi, M., Jin, X., Kauffmann, P., Karampatziakis, N., Kim, D., Khademi, M., Kurilenko, L., Lee, J.R., Lee, Y.T., Li, Y., Li, Y., Liang, C., Liden, L., Liu, C., Liu, M., Liu, W., Lin, E., Lin, Z., Luo, C., Madan, P., Mazzola, M., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Shukla, S., Song, X., Tanaka, M., Tupini, A., Wang, X., Wang, L., Wang, C., Wang, Y., Ward, R., Wang, G., Witte, P., Wu, H., Wyatt, M., Xiao, B., Xu, C., Xu, J., Xu, W., Yadav, S., Yang, F., Yang, J., Yang, Z., Yang, Y., Yu, D., Yuan, L., Zhang, C., Zhang, C., Zhang, J., Zhang, L.L., Zhang, Y., Zhang, Y., Zhang, Y., Zhou, X.: Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint (2024). https://doi.org/10.48550/arXiv.2404.14219
2. Aryabumi, V., Dang, J., Talupuru, D., Dash, S., Cairuz, D., Lin, H., Venkitesh, B., Smith, M., Campos, J.A., Tan, Y.C., Marchisio, K., Bartolo, M., Ruder, S., Locatelli, A., Kreutzer, J., Frosst, N., Gomez, A., Blunsom, P., Fadaee, M., Üstün, A., Hooker, S.: Aya 23: Open weight releases to further multilingual progress. arXiv preprint (2024). https://doi.org/10.48550/arXiv.2405.15032
3. Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., Hoefler, T.: Graph of thoughts: Solving elaborate problems with large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38(16), pp. 17682–17690 (2024). https://doi.org/10.1609/aaai.v38i16.29720
4. C4AI: Model card for c4ai command-r (2024), https://huggingface.co/CohereForAI/c4ai-command-r-v01
5. Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., Berant, J.: Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics **9**, 346–361 (2021), https://api.semanticscholar.org/CorpusID:230799347
6. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 22199–22213. Curran Associates, Inc. (2022)
7. Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Le Bras, R., Choi, Y., Hajishirzi, H.: Generated knowledge prompting for commonsense reasoning. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 3154–3169. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.225, https://aclanthology.org/2022.acl-long.225
8. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? A new dataset for open book question answering. CoRR **abs/1809.02789** (2018), http://arxiv.org/abs/1809.02789

9. Talmor, A., Herzig, J., Lourie, N., Berant, J.: Commonsenseqa: A question answering challenge targeting commonsense knowledge (2019), https://arxiv.org/abs/1811.00937
10. Wang, B., Deng, X., Sun, H.: Iteratively prompt pre-trained language models for chain of thought. arXiv preprint arXiv:2203.08383v3 [cs.CL] (Oct 23 2022)
11. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P.: Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In: Annual Meeting of the Association for Computational Linguistics (2023), https://api.semanticscholar.org/CorpusID:258558102
12. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: deliberate problem solving with large language models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2024)