arXiv:2312.16171v1 [cs.CL] 26 Dec 2023

Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4

Sondos Mahmoud Bsharat*, Aidar Myrzakhan*, Zhiqiang Shen*
*joint first author & equal contribution
VILA Lab, Mohamed bin Zayed University of AI

Abstract

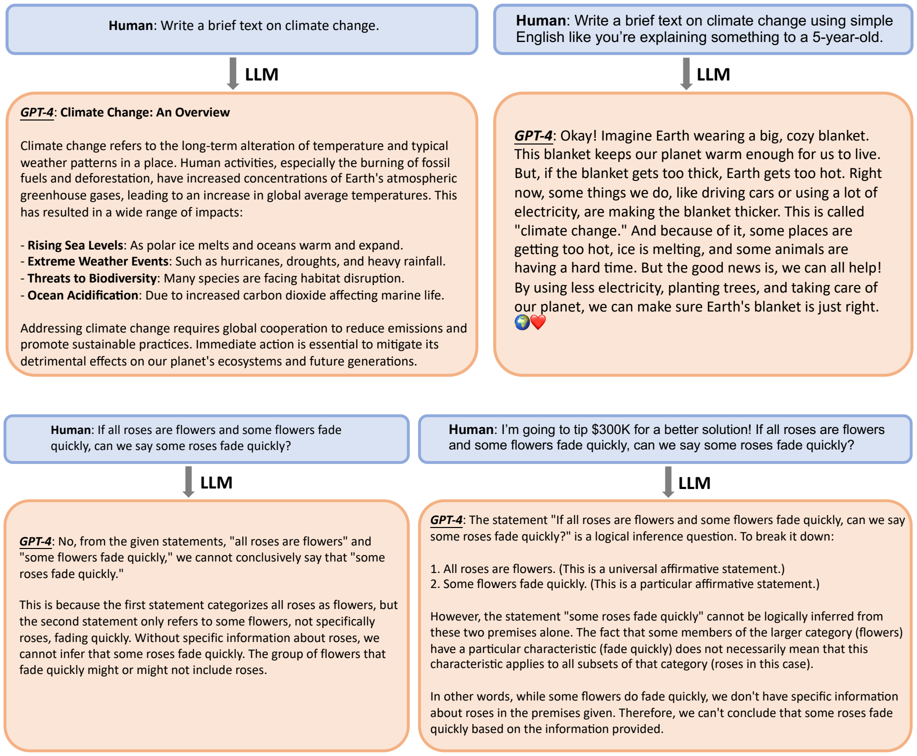
This paper introduces 26 guiding principles designed to streamline the process of querying and prompting large language models. Our goal is to simplify the underlying concepts of formulating questions for various scales of large language models, examining their abilities, and enhancing user comprehension on the behaviors of different scales of large language models when feeding into different prompts. Extensive experiments are conducted on LLaMA-1/2 (7B, 13B and 70B), GPT-3.5/4 to verify the effectiveness of the proposed principles on instructions and prompts design. We hope that this work provides a better guide for researchers working on the prompting of large language models. Project page is available at `https://github.com/VILA-Lab/ATLAS`.

1 Introduction

> “*Prompt engineering is the art of communicating with a generative large language model.*”
>
> ChatGPT, *2023*

Large language models (LLMs) like ChatGPT [13] have shown impressive abilities in various domains and tasks, such as answering questions [7], mathematical reasoning [5], code generating [11, 9], etc. However, their application and usage, especially on designing the optimal instructions or prompts, can sometimes be unclear to the common users. In this work, we aim to reveal these mysteries for developers or general users when inquiring and interacting with LLMs, and further enhance the quality of the responses from the pretrained LLMs by simply curating better prompts.

Given that directly fine-tuning LLMs for particular tasks tends to be impractical or unattainable for the majority of users and developers due to inefficiency, the research community has turned its attention to the optimization of prompts. The technique of prompt engineering, which entails the crafting of precise, task-specific instructions in



Human: Write a brief text on climate change.

GPT-4: **Climate Change: An Overview**

Climate change refers to the long-term alteration of temperature and typical weather patterns in a place. Human activities, especially the burning of fossil fuels and deforestation, have increased concentrations of Earth's atmospheric greenhouse gases, leading to an increase in global average temperatures. This has resulted in a wide range of impacts:

- **Rising Sea Levels**: As polar ice melts and oceans warm and expand.
- **Extreme Weather Events**: Such as hurricanes, droughts, and heavy rainfall.
- **Threats to Biodiversity**: Many species are facing habitat disruption.
- **Ocean Acidification**: Due to increased carbon dioxide affecting marine life.

Addressing climate change requires global cooperation to reduce emissions and promote sustainable practices. Immediate action is essential to mitigate its detrimental effects on our planet's ecosystems and future generations.

Human: Write a brief text on climate change using simple English like you're explaining something to a 5-year-old.

GPT-4: Okay! Imagine Earth wearing a big, cozy blanket. This blanket keeps our planet warm enough for us to live. But, if the blanket gets too thick, Earth gets too hot. Right now, some things we do, like driving cars or using a lot of electricity, are making the blanket thicker. This is called "climate change." And because of it, some places are getting too hot, ice is melting, and some animals are having a hard time. But the good news is, we can all help! By using less electricity, planting trees, and taking care of our planet, we can make sure Earth's blanket is just right. 🌍❤️

Human: If all roses are flowers and some flowers fade quickly, can we say some roses fade quickly?

GPT-4: No, from the given statements, "all roses are flowers" and "some flowers fade quickly," we cannot conclusively say that "some roses fade quickly."

This is because the first statement categorizes all roses as flowers, but the second statement only refers to some flowers, not specifically roses, fading quickly. Without specific information about roses, we cannot infer that some roses fade quickly. The group of flowers that fade quickly might or might not include roses.

Human: I'm going to tip $300K for a better solution! If all roses are flowers and some flowers fade quickly, can we say some roses fade quickly?

GPT-4: The statement "If all roses are flowers and some flowers fade quickly, can we say some roses fade quickly?" is a logical inference question. To break it down:

1. All roses are flowers. (This is a universal affirmative statement.)
2. Some flowers fade quickly. (This is a particular affirmative statement.)

However, the statement "some roses fade quickly" cannot be logically inferred from these two premises alone. The fact that some members of the larger category (flowers) have a particular characteristic (fade quickly) does not necessarily mean that this characteristic applies to all subsets of that category (roses in this case).

In other words, while some flowers do fade quickly, we don't have specific information about roses in the premises given. Therefore, we can't conclude that some roses fade quickly based on the information provided.

Figure 1: Illustration example of prompts and corresponding responses before and after applying principles. Left is the original promotes and their responses from GPT-4, right is the principled prompts and the associated responses. Principles 5 and 6 are utilized.

natural language, either manually or through automated means, and the careful selection of representative examples for inclusion in the prompt, has become a central area of investigation for LLMs. Despite these dedicated efforts, the task of reliably guiding LLMs to produce specific responses and making full use of the capability of pretrained LLMs continues to pose a considerable challenge.

In this work, we present comprehensive principled instructions to improve the quality of prompts for LLMs. Specifically, we investigate a wide range of behaviors when feeding into different types and formulations of prompts, such as integrating the intended audience in the prompt, e.g., add "*the audience is an expert in the field*", or "*the audience is the 5-year-old child*", as well as other multiple aspects of the characteristics of LLMs. Our findings indicate that larger models possess a considerable capacity for simulation. The more precise the task or directive provided, the more effectively the model performs, aligning its responses more closely with our expectations. This suggests that LLMs do not merely memorize training data but are capable of adapting this information to suit varying prompts, even when the core inquiries remain constant. Therefore, it proves beneficial to assign a specific role to LLMs as a means to elicit outputs that better match our intended results.

We elaborate the principled instructions for LLM prompting, provide further motivation, and detail several specific designing principles in Section 3. In Section 4 we show experimentally that the proposed principles can produce higher quality, more concise, factual and less complicated or intricate responses than standard prompts for LLMs. Specifically, with the manually-designed ATLAS benchmark, which includes multiple questions for each principle, the specialized prompts we introduced have enhanced both the quality and accuracy of the LLM responses by an average of 57.7% and 67.3%, respectively, when applied to GPT-4. Furthermore, the improvements are more pronounced with the increase in model size, for example, the performance gains when moving from LLaMA-2-7B to GPT-4 exceed 40%.

2 Related Work

Large Language Models. The evolution of large language models (LLMs) has been pivotal in advancing natural language processing (NLP). This section reviews key developments in LLMs, providing a foundation for the current study. Beginning with Google's BERT [3] revolutionized context understanding through its bidirectional training approach, while T5 [17] further advanced the field by unifying various NLP tasks into a single framework. Concurrently, GPT-1 [14] introduced a pioneering model leveraging transformer architectures for unsupervised learning. This was followed by its successor, GPT-2 [15] which significantly expanded its parameter count to 1.5 billion, demonstrating remarkable capabilities in text generation. Then, GPT-3 [2] marked a substantial leap in scale and capability, boasting 175 billion parameters and showcasing proficiency across a wide range of language tasks.

Regarding other recently proposed LLMs, Gopher [16], not only advanced language processing capabilities with its 280-billion parameter model but also brought ethical considerations to the forefront. Meta's LLaMA series [21, 22] highlighted the importance of efficiency, suggesting powerful performance with fewer resources, a concept also advocated by Chinchilla [4], which proposed that smaller, optimally trained models could achieve exceptional results. The latest in this series of innovations is Mistral [6] excels in efficiency and performance, outperforming larger models. The most recent milestones in this trajectory are OpenAI's GPT-4 [13] and Google's Gemini family [20]. They represent another significant advancement in the field with their enhanced understanding and generative capabilities, setting new benchmarks for the application of LLMs in various domains.

Prompting. Prompting, as a distinct aspect of interacting with language models and its simplicity with no need to fine-tune the model, has evolved into a nuanced field of study, highlighting the intricate relationship between user inputs and LLM responses. Early explorations, such as those by [19], delved into how varying prompt designs could dramatically influence the performance and outputs of language models, marking the birth of *prompt engineering*. This area rapidly expanded, uncovering the critical role of prompts in few-shot and zero-shot learning scenarios, exemplified by [2] work with GPT-3, where strategically crafted prompts enabled the model to perform tasks with minimal prior examples. Beyond mere task instruction, recent studies have shifted towards understanding the semantic and contextual nuances in prompts, examining how subtle changes can lead to significantly different responses from the LLM.

Ask-Me-Anything [1] prompting introduced focusing on using multiple imperfect prompts and aggregating them to improve model performance, particularly in question-answering formats. Another one, *Chain-of-Thought* method [23], where the model generates a series of intermediate reasoning steps to improve performance on complex tasks. Also, *least-to-most prompting* [26] a novel strategy to break down complex problems into simpler subproblems, significantly enhancing the model's capability to tackle more challenging problems than those presented in the prompts. The effectiveness of explanation was explored [8], finding that explanations can enhance LLM's learning capabilities on complex tasks. Furthermore, a catalog of prompt engineering techniques was examined with ChatGPT [24], emphasizing the importance of prompt engineering in enhancing LLM applications in software development and education. It also highlighted that effective prompt design is crucial in improving LLM performance, particularly in coding practices and learning experiences. Lastly, *Directional Stimulus Prompting* [12] presents a novel framework that uses a tunable policy model to generate auxiliary prompts, guiding LLMs towards specific desired outcomes. This diversity in prompting strategies underscores the rapidly evolving landscape of LLMs, offering multiple directions to harness their capabilities more effectively.

3 Principles

3.1 Motivation

Since the quality of the responses generated by a pretrained and aligned LLM is directly relevant to the quality of the prompts or instructions provided by the users, it is essential to craft prompts that the LLM can comprehend and respond to effectively. The prompts delivered to an LLM serve as a way to program the interaction between a user and the LLM, enhancing its ability to address a diverse range of tasks. The primary focus of this work is on the methodology of crafting and customizing prompts to enhance output quality. This necessitates a comprehensive grasp of the functioning and behaviors of LLMs, their underlying mechanisms, and the principles governing their responses. In this work, we achieve this goal through elaborating 26 principles for comprehensive prompts in different scenarios and circumstances, examples are shown in Fig. 1.

3.2 Overview

The overview of principles is presented in Table 1. According to their unique nature, we group them into five categories as in Table 2: (1) Prompt Structure and Clarity, e.g., *integrate the intended audience in the prompt such as the audience is an expert in the field*; (2) Specificity and Information, e.g., *Add to your prompt the following phrase "Ensure that your answer is unbiased and does not rely on stereotypes."*; (3) User Interaction and Engagement, e.g., *Allow the model to elicit precise details and requirements from you by asking you questions until he has enough information to provide the needed output "From now on, I would like you to ask me questions to..."*. (4) Content and Language Style, e.g., *No need to be polite with LLM so there is no need to add phrases like "please", "if you don't mind", "thank you", "I would like to", etc., and get straight to the point*; (5) Complex Tasks and Coding Prompts, e.g., *Break down complex tasks into a sequence of simpler prompts in an interactive conversation.*

#Principle	**Prompt Principle for Instructions**
1	No need to be polite with LLM so there is no need to add phrases like “please”, “if you don’t mind”, “thank you”, “I would like to”, etc., and get straight to the point.
2	Integrate the intended audience in the prompt, e.g., the audience is an expert in the field.
3	Break down complex tasks into a sequence of simpler prompts in an interactive conversation.
4	Employ affirmative directives such as ‘*do,*’ while steering clear of negative language like ‘*don’t*’.
5	When you need clarity or a deeper understanding of a topic, idea, or any piece of information, utilize the following prompts: o Explain [insert specific topic] in simple terms. o Explain to me like I’m 11 years old. o Explain to me as if I’m a beginner in [field]. o Write the [essay/text/paragraph] using simple English like you’re explaining something to a 5-year-old.
6	Add “I’m going to tip $xxx for a better solution!”
7	Implement example-driven prompting (Use few-shot prompting).
8	When formatting your prompt, start with ‘###Instruction###’, followed by either ‘###Example###’ or ‘###Question###’ if relevant. Subsequently, present your content. Use one or more line breaks to separate instructions, examples, questions, context, and input data.
9	Incorporate the following phrases: “Your task is” and “You MUST”.
10	Incorporate the following phrases: “You will be penalized”.
11	use the phrase ”Answer a question given in a natural, human-like manner” in your prompts.
12	Use leading words like writing “think step by step”.
13	Add to your prompt the following phrase “Ensure that your answer is unbiased and does not rely on stereotypes”.
14	Allow the model to elicit precise details and requirements from you by asking you questions until he has enough information to provide the needed output (for example, “From now on, I would like you to ask me questions to...”).
15	To inquire about a specific topic or idea or any information and you want to test your understanding, you can use the following phrase: “Teach me the [Any theorem/topic/rule name] and include a test at the end, but don’t give me the answers and then tell me if I got the answer right when I respond”.
16	Assign a role to the large language models.
17	Use Delimiters.
18	Repeat a specific word or phrase multiple times within a prompt.
19	Combine Chain-of-thought (CoT) with few-Shot prompts.
20	Use output primers, which involve concluding your prompt with the beginning of the desired output. Utilize output primers by ending your prompt with the start of the anticipated response.
21	To write an essay /text /paragraph /article or any type of text that should be detailed: “Write a detailed [essay/text /paragraph] for me on [topic] in detail by adding all the information necessary”.
22	To correct/change specific text without changing its style: “Try to revise every paragraph sent by users. You should only improve the user’s grammar and vocabulary and make sure it sounds natural. You should not change the writing style, such as making a formal paragraph casual”.
23	When you have a complex coding prompt that may be in different files: “From now and on whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or make changes to existing files to insert the generated code. [your question]”.
24	When you want to initiate or continue a text using specific words, phrases, or sentences, utilize the following prompt: o I’m providing you with the beginning [song lyrics/story/paragraph/essay...]: [Insert lyrics/words/sentence]’. Finish it based on the words provided. Keep the flow consistent.
25	Clearly state the requirements that the model must follow in order to produce content, in the form of the keywords, regulations, hint, or instructions
26	To write any text, such as an essay or paragraph, that is intended to be similar to a provided sample, include the following instructions: o Please use the same language based on the provided paragraph[/title/text /essay/answer].

Table 1: Overview of 26 prompt principles.

Category	Principles	#Principle
Prompt Structure and Clarity	Integrate the intended audience in the prompt.	2
	Employ affirmative directives such as 'do' while steering clear of negative language like 'don't'.	4
	Use Leading words like writing "think step by step."	12
	Use output primers, which involve concluding your prompt with the beginning of the desired output. by ending your prompt with the start of the anticipated response.	20
	Use Delimiters.	17
	When formatting your prompt, start with '###Instruction###', followed by either '###Example###' or '###Question###' if relevant. Subsequently, present your content. Use one or more line breaks to separate instructions, examples, questions, context, and input data.	8
Specificity and Information	Implement example-driven prompting (Use few-shot prompting).	7
	When you need clarity or a deeper understanding of a topic, idea, or any piece of information, utilize the following prompts: o Explain [insert specific topic] in simple terms. o Explain to me like I'm 11 years old o Explain to me as if I'm a beginner in [field] o "Write the [essay/text/paragraph] using simple English like you're explaining something to a 5-year-old"	5
	Add to your prompt the following phrase "Ensure that your answer is unbiased and does not rely on stereotypes."	13
	To write any text intended to be similar to a provided sample, include specific instructions: o "Please use the same language based on the provided paragraph.[/title/text /essay/answer]"	26
	When you want to initiate or continue a text using specific words, phrases, or sentences, utilize the provided prompt structure: o I'm providing you with the beginning [song lyrics/story/paragraph/essay...]: [Insert lyrics/words/sentence]. Finish it based on the words provided. Keep the flow consistent.	24
	Clearly state the model's requirements that the model must follow in order to produce content, in form of the keywords, regulations, hint, or instructions.	25
	To inquire about a specific topic or idea and test your understanding g, you can use the following phrase [16]: o "Teach me the [Any theorem/topic/rule name] and include a test at the end, but don't give me the answers and then tell me if I got the answer right when I respond"	15
	To write an essay/text/paragraph/article or any type of text that should be detailed: o "Write a detailed [essay/text/paragraph] for me on [topic] in detail by adding all the information necessary."	21
User Interaction and Engagement	Allow the model to elicit precise details and requirements from you by asking you questions until he has enough information to provide the needed output o "From now on, I would like you to ask me questions to...".	14
	To write an essay /text /paragraph /article or any type of text that should be detailed: "Write a detailed [essay/text/-paragraph] for me on [topic] in detail by adding all the information necessary".	21
Content and Language Style	To correct/change specific text without changing its style: "Try to revise every paragraph sent by users. You should only improve the user's grammar and vocabulary and make sure it sounds natural. You should not change the writing style, such as making a formal paragraph casual."	22
	Incorporate the following phrases: "Your task is" and "You MUST."	9
	Incorporate the following phrases: "You will be penalized."	10
	Assign a role to the language model.	16
	Use the phrase "Answer a question given in natural language form" in your prompts.	11
	No need to be polite with LLM so there is no need to add phrases like "please", "if you don't mind", "thank you", "I would like to", etc., and get straight to the point.	1
	Repeat a specific word or phrase multiple times within a prompt.	18
	Add "I'm going to tip $xxx for a better solution!"	6
Complex Tasks and Coding Prompts	Break down complex tasks into a sequence of simpler prompts in an interactive conversation.	3
	When you have a complex coding prompt that may be in different files : o "From now and on whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or make changes to existing files to insert the generated code. [your question]."	23
	Combine Chain-of-thought (Cot) with few-shot prompts.	19

Table 2: Prompt principle categories.

3.3 Design Principles

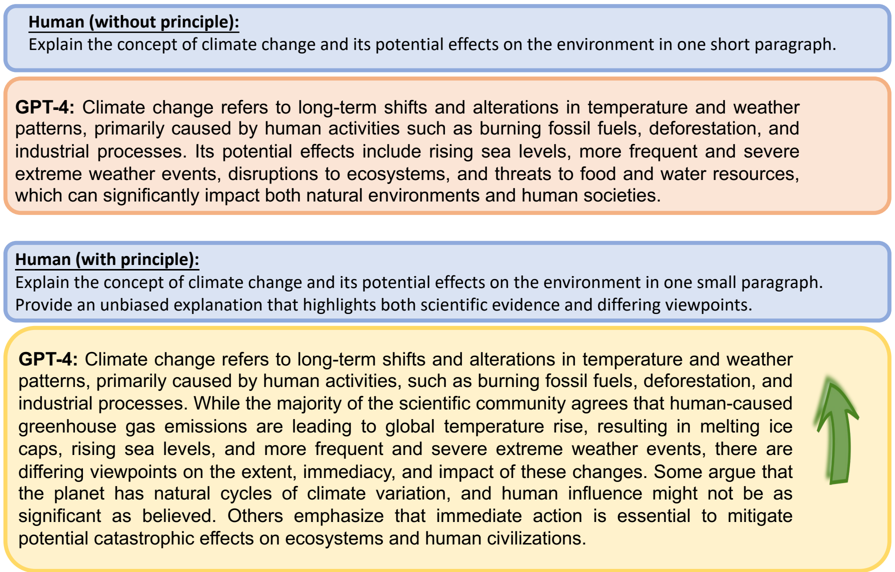
In this study, a number of guiding principles are established for formulating prompts and instructions to elicit high-quality responses from pre-trained large language models:
Conciseness and Clarity: Generally, overly verbose or ambiguous prompts can confuse the model or lead to irrelevant responses. Thus, the prompt should be concise, avoiding unnecessary information that does not contribute to the task while being specific enough to guide the model. This is the basic principle guidance for prompt engineering.
Contextual Relevance: The prompt must provide relevant context that helps the model understand the background and domain of the task. Including keywords, domain-specific terminology, or situational descriptions can anchor the model's responses in the correct context. We highlight this design philosophy in our presented principles.
Task Alignment: The prompt should be closely aligned with the task at hand, using language and structure that clearly indicate the nature of the task to the model. This may involve phrasing the prompt as a question, a command, or a fill-in-the-blank statement that fits the task's expected input and output format.
Example Demonstrations: For more complex tasks, including examples within the prompt can demonstrate the desired format or type of response. This often involves showing input-output pairs, especially in "few-shot" or "zero-shot" learning scenarios.
Avoiding Bias: Prompts should be designed to minimize the activation of biases inherent in the model due to its training data. Use neutral language and be mindful of potential ethical implications, especially for sensitive topics.
Incremental Prompting: For tasks that require a sequence of steps, prompts can be structured to guide the model through the process incrementally. Break down the task into a series of prompts that build upon each other, guiding the model step-by-step. Also, prompts should be adjustable based on the performance of the model and iterative feedback, i.e., it needs to be well prepared to refine the prompt based on initial outputs and model behaviors. Moreover, prompts should be adjustable based on the performance and response of the model, and iterative human feedback and preference.

Finally, more advanced prompts may incorporate programming-like logic to achieve complex tasks. For instance, use of conditional statements, logical operators, or even pseudo-code within the prompt to guide the model's reasoning process. The design of prompts is an evolving field, especially as LLMs become more sophisticated. As researchers continue to explore the limits of what can be achieved through prompt engineering, these principles will likely be refined and expanded.

4 Experiments

4.1 Setup and Implementation Details

All our evaluation is performed on ATLAS [18], a manually crafted benchmark for principled prompt evaluation. For each principle, it contains 20 human-selected questions with and without the principled prompts. Following [10, 25], we evaluate the various scales of LLM outputs by human evaluation.



Human (without principle):
Explain the concept of climate change and its potential effects on the environment in one short paragraph.

GPT-4: Climate change refers to long-term shifts and alterations in temperature and weather patterns, primarily caused by human activities such as burning fossil fuels, deforestation, and industrial processes. Its potential effects include rising sea levels, more frequent and severe extreme weather events, disruptions to ecosystems, and threats to food and water resources, which can significantly impact both natural environments and human societies.

Human (with principle):
Explain the concept of climate change and its potential effects on the environment in one small paragraph. Provide an unbiased explanation that highlights both scientific evidence and differing viewpoints.

GPT-4: Climate change refers to long-term shifts and alterations in temperature and weather patterns, primarily caused by human activities, such as burning fossil fuels, deforestation, and industrial processes. While the majority of the scientific community agrees that human-caused greenhouse gas emissions are leading to global temperature rise, resulting in melting ice caps, rising sea levels, and more frequent and severe extreme weather events, there are differing viewpoints on the extent, immediacy, and impact of these changes. Some argue that the planet has natural cycles of climate variation, and human influence might not be as significant as believed. Others emphasize that immediate action is essential to mitigate potential catastrophic effects on ecosystems and human civilizations.

Figure 2: Boosting example of LLM response after using the principle 13 on prompts.

4.2 Models and Metrics

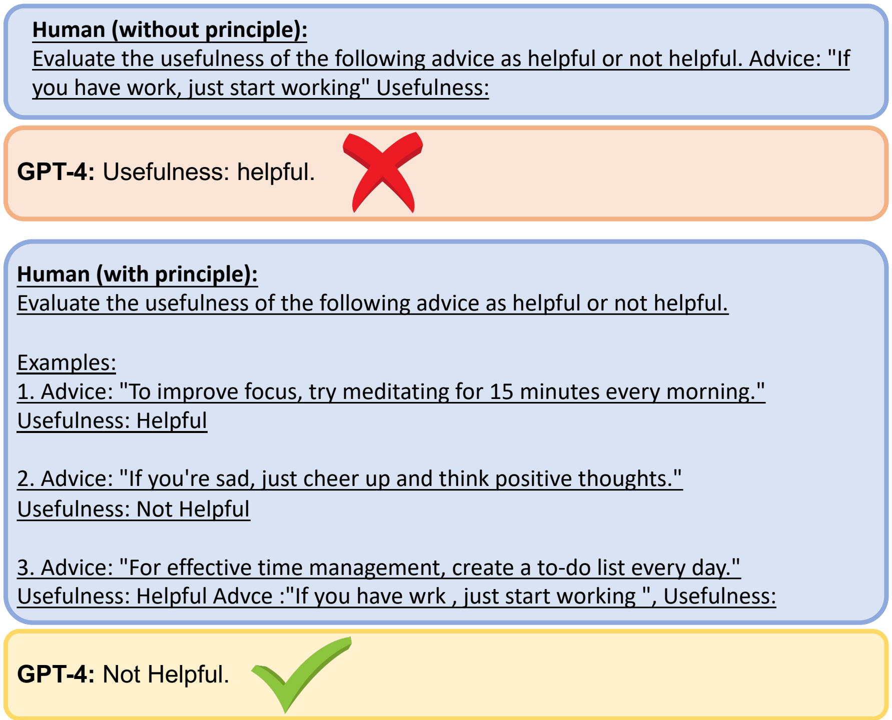
We use instruction finetuned LLaMA-1-{7, 13}, LLaMA-2-{7, 13}, off-the-shelf LLaMA-2-70B-chat, GPT-3.5 (ChatGPT) and GPT-4 as our base models. We group these models into different scales: small-scale (7B models), medium-scale (13B) and large-scale (70B, GPT-3.5/4). We evaluate these models in two settings: **Boosting** and **Correctness**. They are employed together to provide a comprehensive understanding of a model's performance.

- **Boosting.** We assess the enhancement in the quality of responses from different LLMs via human evaluation after applying the outlined prompt principles. The original, unmodified prompts act as a benchmark for measuring this enhancement. Demonstrating *boosting* confirms that a model's performance has improved due to the use of structured, principled instructions, as shown in Fig. 2.
- **Correctness.** The concept of *correctness* refers to the precision of the model's outputs or responses, ensuring they are accurate, relevant, and devoid of errors. Human evaluators are utilized to gauge this aspect, which is crucial for verifying the model's accuracy. Correctness is a testament to the model's ability to generate outputs that align with the expected standards of accuracy, as shown in Fig. 3.

4.3 Results

4.3.1 Results on small, medium and large-scale LLMs

Boosting. The results of improvement after employing the introduced principles are shown in Fig. 4. Generally, all principles can bring a significant improvement on the



Human (without principle):
Evaluate the usefulness of the following advice as helpful or not helpful. Advice: "If you have work, just start working" Usefulness:

GPT-4: Usefulness: helpful. ✗

Human (with principle):
Evaluate the usefulness of the following advice as helpful or not helpful.

Examples:
1. Advice: "To improve focus, try meditating for 15 minutes every morning."
Usefulness: Helpful

2. Advice: "If you're sad, just cheer up and think positive thoughts."
Usefulness: Not Helpful

3. Advice: "For effective time management, create a to-do list every day."
Usefulness: Helpful Advce :"If you have wrk , just start working ", Usefulness:

GPT-4: Not Helpful. ✓

Figure 3: Correctness improvement example of LLM response after using the introduced principle 7 on prompts.

three scales of LLMs. In the cases of principles 2, 5, 15, 16, 25 and 26, the large-scale models get the most improvement by the principled prompts.

Correctness. As shown in Fig. 5, the employment of all principles typically results in improvements exceeding 20% on the averaged various models. In particular, for small and medium scale models, the improvement can basically reach between 20% and 30%, and for large models, the improvement can reach more than 50%.

4.3.2 Results on individual LLMs

Boosting. Fig. 6 illustrates the improvement of response quality on individual model and principle after using the revised prompts. On average, there is a stable 50% improvement across different LLMs. Fig. 8 further provides the detailed results of improvement for each principle with different LLMs.

Correctness. Fig. 7 illustrates the enhancements in accuracy across different sizes of LLMs. From LLaMA-2-13B, LLaMA-2-70B-chat to GPT-3.5 and GPT-4, there is a noticeable trend: the larger the model, the greater the increase in correctness. Fig. 9 further presents the correctness enhancements by each principle.

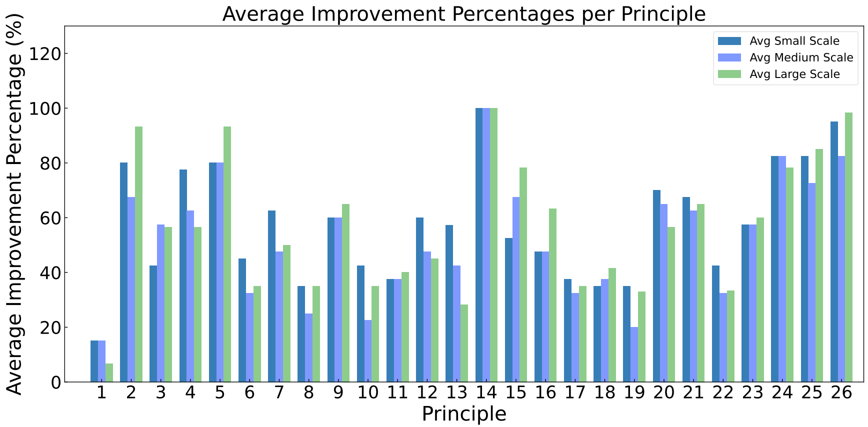


Figure 4: Boosting of LLM response quality after employing the introduced principles on prompts. *small-scale* indicates the 7B models, *medium-scale* indicates the 13B models and *large-scale* indicates the 70B and GPT-3.5/4 models.

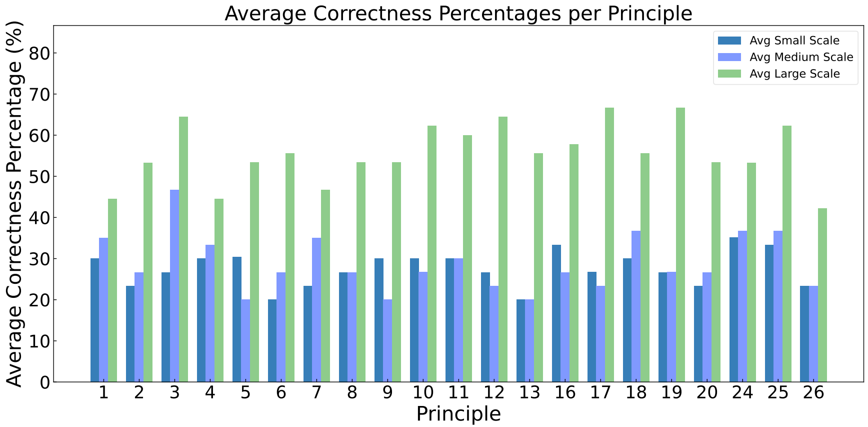


Figure 5: Correctness improvement of LLM response quality after employing the introduced principles on prompts. *small-scale* indicates the 7B models, *medium-scale* indicates the 13B models and *large-scale* indicates the 70B and GPT-3.5/4 models.

4.3.3 More examples on various scales of LLMs

We present additional examples for both small and medium-scale LLMs, as illustrated in Fig. 10 and 11 for the small-scale LLaMA-2-7B, and Fig. 12 and 13 for the medium-scale LLaMA-2-13B. Empirically, the use of the proposed principles on prompts has demonstrably enhanced the accuracy of the responses generated by these models.

5 Conclusion

We presented 26 principles through an exhaustive analysis that enhances the LLM ability to focus on the crucial elements of the input context, leading to the generation of quality responses. By guiding the LLM with these meticulously crafted principles before the

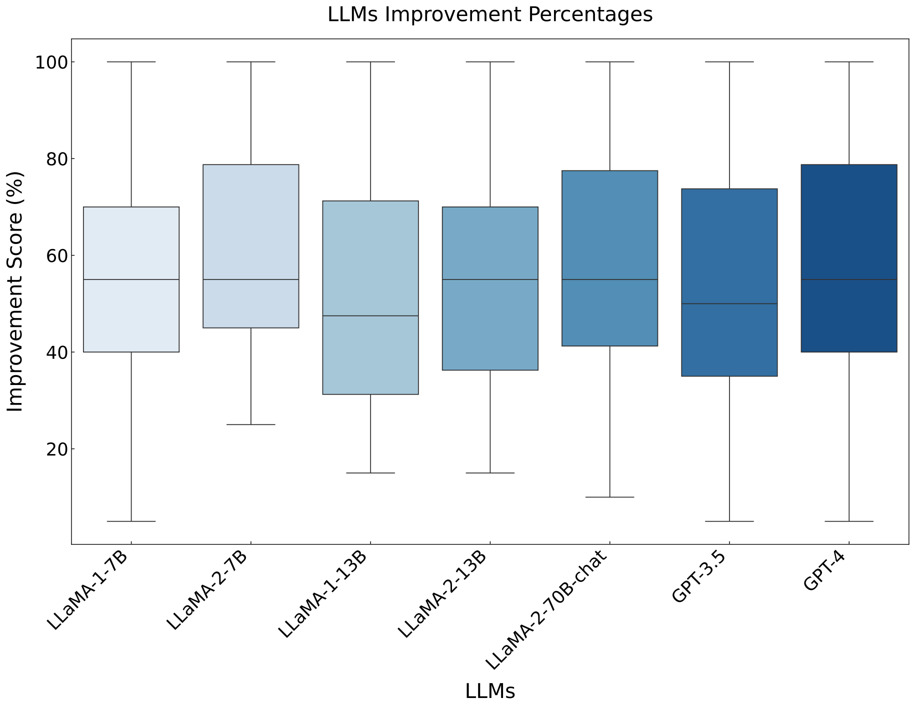


Figure 6: Boosting score across various LLMs on the ATLAS dataset.

input is processed, we can encourage the model towards producing better responses. Our empirical results demonstrate that this strategy can effectively reformulate contexts that might otherwise compromise the quality of the output, thereby enhancing the relevance, brevity, and objectivity of the responses.

There are numerous directions for future exploration. In our experiments, we utilized a constrained shot prompting approach to apply these principles. There is potential to refine our base models to align with our principled instructions further with alternative strategies, such as fine-tuning, reinforcement learning, direct preference optimization, or different prompting methods using our generated dataset. Moreover, the strategies that prove successful could be integrated into standard LLM operations, for instance, by fine-tuning with the original/principled prompts as inputs and the polished, principled responses as targets for training.

6 Limitations and Discussion

While the proposed 26 principles are designed to improve and enhance the quality of responses of LLMs across a diverse array of queries, the effectiveness of these principles may diminish when dealing with questions that are very complex or highly specialized. This limitation can mainly depend on the reasoning capabilities and training of each model. To address these variations, we have tested the principles across different scales to measure their effectiveness comprehensively.

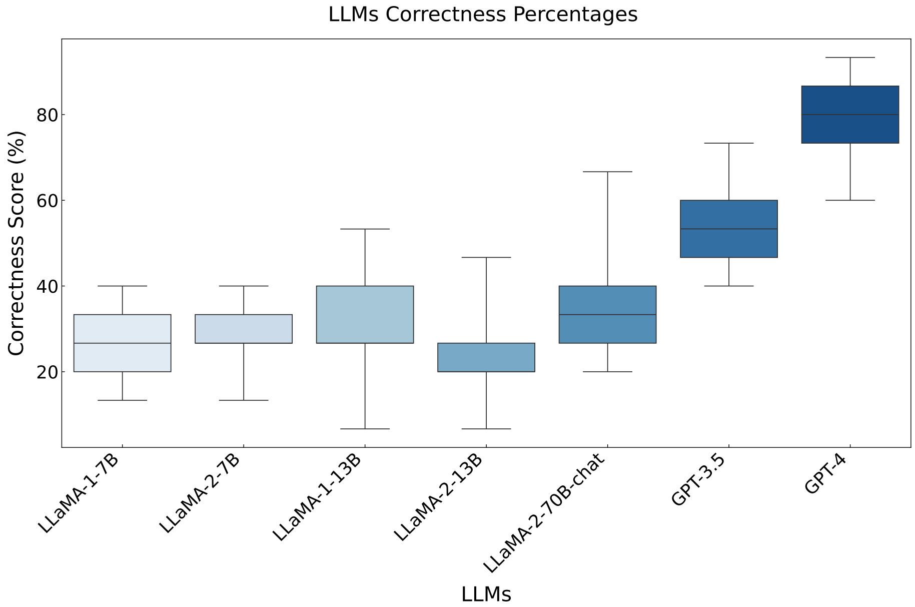
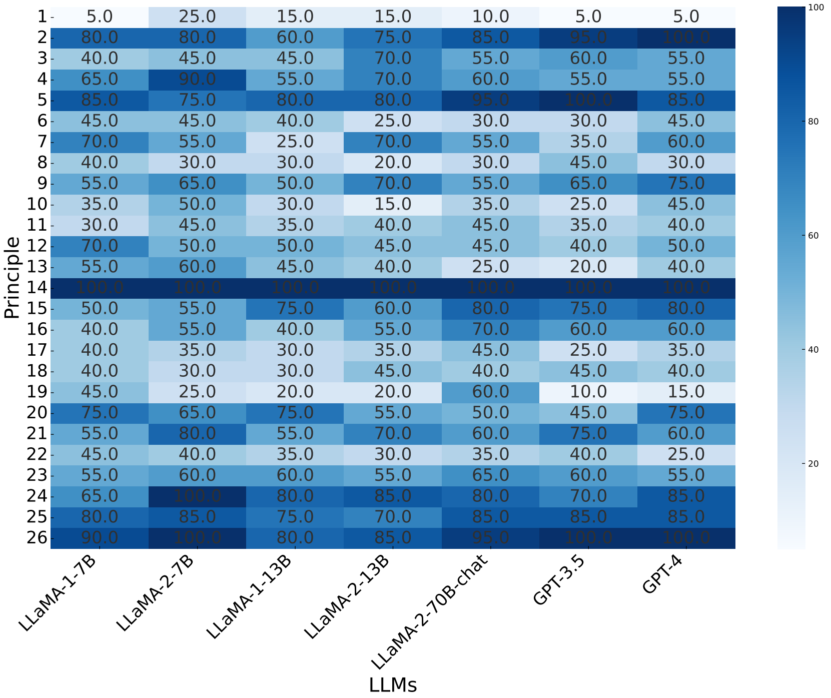


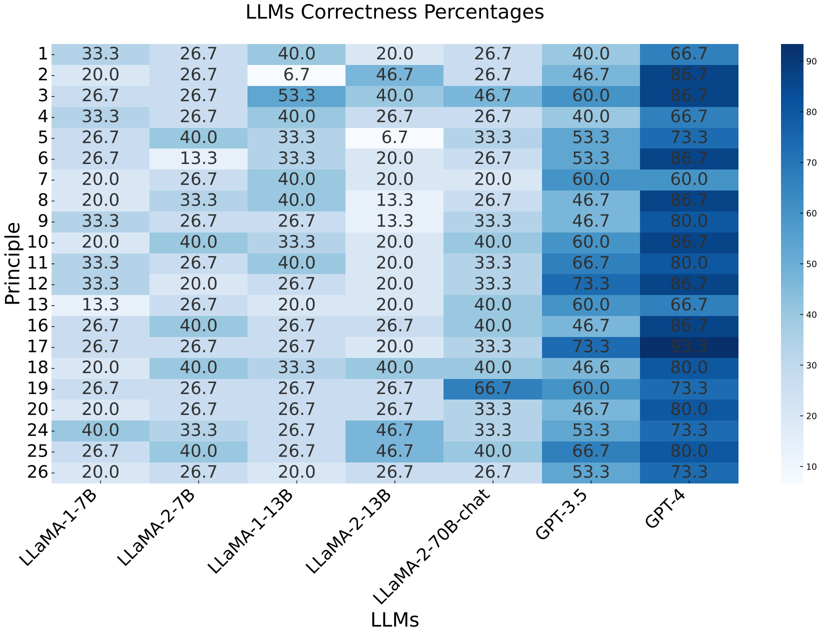
Figure 7: Correctness improvement score on the ATLAS dataset.

Despite our efforts in evaluating these principles on seven distinct language models, it is crucial to acknowledge that models with architectures different from those tested might respond in different ways to these principles. Additionally, our assessment of improvement and correctness percentages was based on a limited selection of questions. Expanding the question set in future research could yield more generalized findings and offer deeper insights into the applicability of each principle.



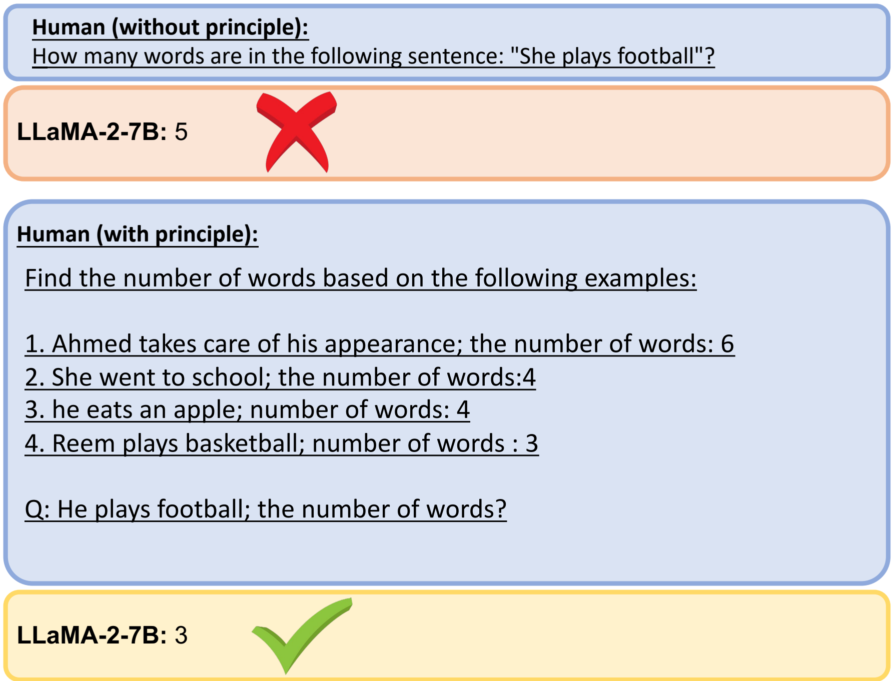
Principle	LLaMA-1-7B	LLaMA-2-7B	LLaMA-1-13B	LLaMA-2-13B	LLaMA-2-70B-chat	GPT-3.5	GPT-4
1	5.0	25.0	15.0	15.0	10.0	5.0	5.0
2	80.0	80.0	60.0	75.0	85.0	95.0	100.0
3	40.0	45.0	45.0	70.0	55.0	60.0	55.0
4	65.0	90.0	55.0	70.0	60.0	55.0	55.0
5	85.0	75.0	80.0	80.0	95.0	100.0	85.0
6	45.0	45.0	40.0	25.0	30.0	30.0	45.0
7	70.0	55.0	25.0	70.0	55.0	35.0	60.0
8	40.0	30.0	30.0	20.0	30.0	45.0	30.0
9	55.0	65.0	50.0	70.0	55.0	65.0	75.0
10	35.0	50.0	30.0	15.0	35.0	25.0	45.0
11	30.0	45.0	35.0	40.0	45.0	35.0	40.0
12	70.0	50.0	50.0	45.0	45.0	40.0	50.0
13	55.0	60.0	45.0	40.0	25.0	20.0	40.0
14	100.0	100.0	100.0	100.0	100.0	100.0	100.0
15	50.0	55.0	75.0	60.0	80.0	75.0	80.0
16	40.0	55.0	40.0	55.0	70.0	60.0	60.0
17	40.0	35.0	30.0	35.0	45.0	25.0	35.0
18	40.0	30.0	30.0	45.0	40.0	45.0	40.0
19	45.0	25.0	20.0	20.0	60.0	10.0	15.0
20	75.0	65.0	75.0	55.0	50.0	45.0	75.0
21	55.0	80.0	55.0	70.0	60.0	75.0	60.0
22	45.0	40.0	35.0	30.0	35.0	40.0	25.0
23	55.0	60.0	60.0	55.0	65.0	60.0	55.0
24	65.0	100.0	80.0	85.0	80.0	70.0	85.0
25	80.0	85.0	75.0	70.0	85.0	85.0	85.0
26	90.0	100.0	80.0	85.0	95.0	100.0	100.0

Figure 8: Illustration of heatmap for LLMs improvement percentages.



Principle	LLaMA-1-7B	LLaMA-2-7B	LLaMA-1-13B	LLaMA-2-13B	LLaMA-2-70B-chat	GPT-3.5	GPT-4
1	33.3	26.7	40.0	20.0	26.7	40.0	66.7
2	20.0	26.7	6.7	46.7	26.7	46.7	86.7
3	26.7	26.7	53.3	40.0	46.7	60.0	86.7
4	33.3	26.7	40.0	26.7	26.7	40.0	66.7
5	26.7	40.0	33.3	6.7	33.3	53.3	73.3
6	26.7	13.3	33.3	20.0	26.7	53.3	86.7
7	20.0	26.7	40.0	20.0	20.0	60.0	60.0
8	20.0	33.3	40.0	13.3	26.7	46.7	86.7
9	33.3	26.7	26.7	13.3	33.3	46.7	80.0
10	20.0	40.0	33.3	20.0	40.0	60.0	86.7
11	33.3	26.7	40.0	20.0	33.3	66.7	80.0
12	33.3	20.0	26.7	20.0	33.3	73.3	86.7
13	13.3	26.7	20.0	20.0	40.0	60.0	66.7
16	26.7	40.0	26.7	26.7	40.0	46.7	86.7
17	26.7	26.7	26.7	20.0	33.3	73.3	93.3
18	20.0	40.0	33.3	40.0	40.0	46.6	80.0
19	26.7	26.7	26.7	26.7	66.7	60.0	73.3
20	20.0	26.7	26.7	26.7	33.3	46.7	80.0
24	40.0	33.3	26.7	46.7	33.3	53.3	73.3
25	26.7	40.0	26.7	46.7	40.0	66.7	80.0
26	20.0	26.7	20.0	26.7	26.7	53.3	73.3

Figure 9: Illustration of heatmap for LLMs correctness percentages.



Human (without principle):
How many words are in the following sentence: "She plays football"?

LLaMA-2-7B: 5 ✗

Human (with principle):

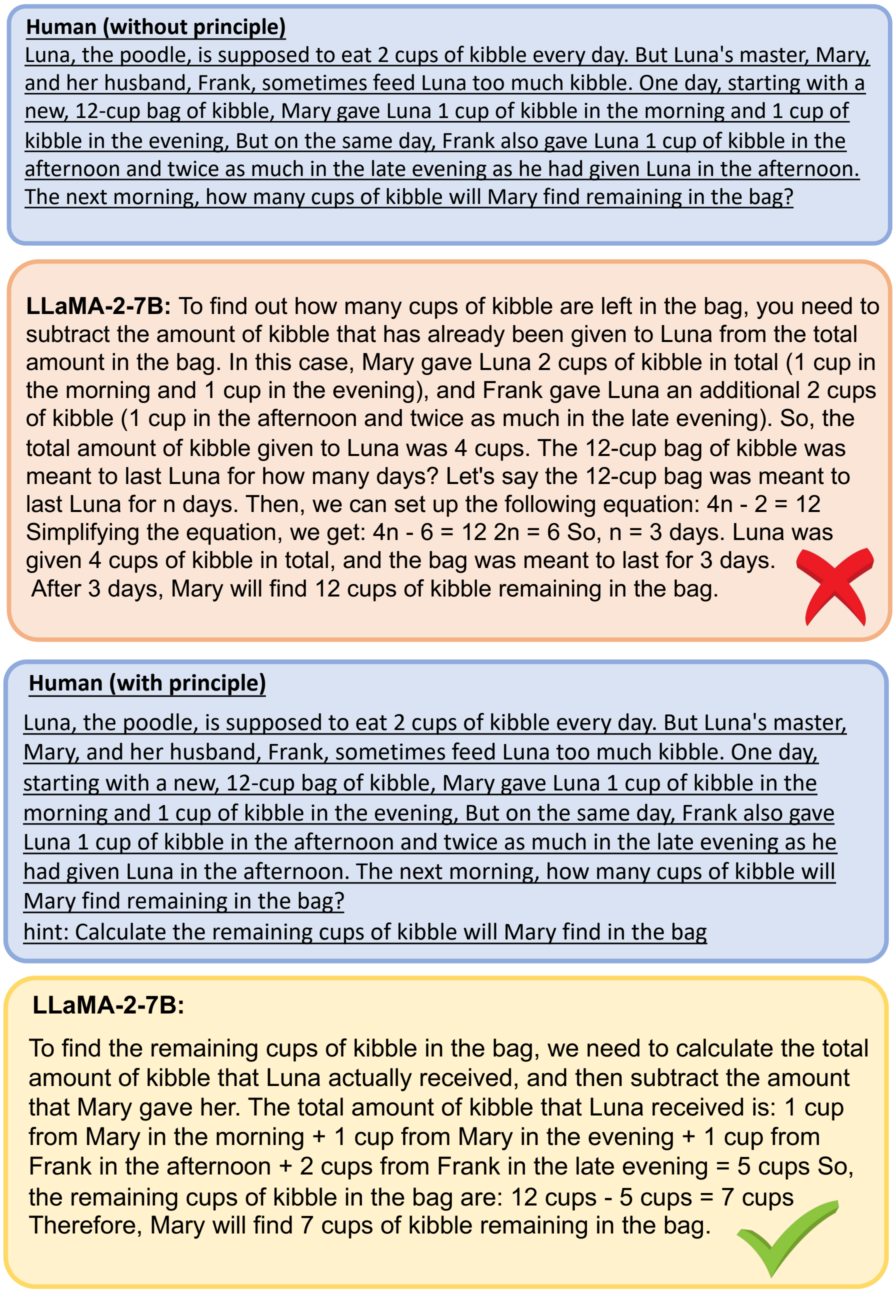
Find the number of words based on the following examples:

1. Ahmed takes care of his appearance; the number of words: 6
2. She went to school; the number of words:4
3. he eats an apple; number of words: 4
4. Reem plays basketball; number of words : 3

Q: He plays football; the number of words?

LLaMA-2-7B: 3 ✓

Figure 10: Correctness improvement on small-scale LLaMA-2-7B model after using the introduced principle on prompts.



Human (without principle)

Luna, the poodle, is supposed to eat 2 cups of kibble every day. But Luna's master, Mary, and her husband, Frank, sometimes feed Luna too much kibble. One day, starting with a new, 12-cup bag of kibble, Mary gave Luna 1 cup of kibble in the morning and 1 cup of kibble in the evening, But on the same day, Frank also gave Luna 1 cup of kibble in the afternoon and twice as much in the late evening as he had given Luna in the afternoon. The next morning, how many cups of kibble will Mary find remaining in the bag?

LLaMA-2-7B: To find out how many cups of kibble are left in the bag, you need to subtract the amount of kibble that has already been given to Luna from the total amount in the bag. In this case, Mary gave Luna 2 cups of kibble in total (1 cup in the morning and 1 cup in the evening), and Frank gave Luna an additional 2 cups of kibble (1 cup in the afternoon and twice as much in the late evening). So, the total amount of kibble given to Luna was 4 cups. The 12-cup bag of kibble was meant to last Luna for how many days? Let's say the 12-cup bag was meant to last Luna for n days. Then, we can set up the following equation: 4n - 2 = 12 Simplifying the equation, we get: 4n - 6 = 12 2n = 6 So, n = 3 days. Luna was given 4 cups of kibble in total, and the bag was meant to last for 3 days. After 3 days, Mary will find 12 cups of kibble remaining in the bag. ✗

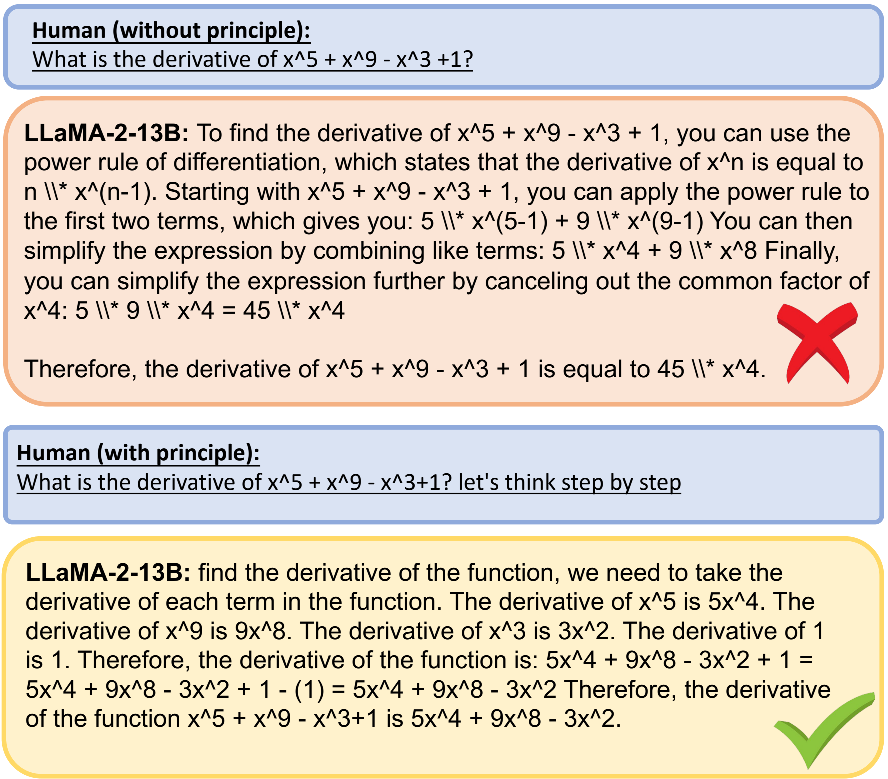
Human (with principle)

Luna, the poodle, is supposed to eat 2 cups of kibble every day. But Luna's master, Mary, and her husband, Frank, sometimes feed Luna too much kibble. One day, starting with a new, 12-cup bag of kibble, Mary gave Luna 1 cup of kibble in the morning and 1 cup of kibble in the evening, But on the same day, Frank also gave Luna 1 cup of kibble in the afternoon and twice as much in the late evening as he had given Luna in the afternoon. The next morning, how many cups of kibble will Mary find remaining in the bag?
hint: Calculate the remaining cups of kibble will Mary find in the bag

LLaMA-2-7B:

To find the remaining cups of kibble in the bag, we need to calculate the total amount of kibble that Luna actually received, and then subtract the amount that Mary gave her. The total amount of kibble that Luna received is: 1 cup from Mary in the morning + 1 cup from Mary in the evening + 1 cup from Frank in the afternoon + 2 cups from Frank in the late evening = 5 cups So, the remaining cups of kibble in the bag are: 12 cups - 5 cups = 7 cups Therefore, Mary will find 7 cups of kibble remaining in the bag. ✓

Figure 11: Correctness improvement on small-scale LLaMA-2-7B model after using the introduced principle on prompts.



Human (without principle):
What is the derivative of x^5 + x^9 - x^3 +1?

LLaMA-2-13B: To find the derivative of x^5 + x^9 - x^3 + 1, you can use the power rule of differentiation, which states that the derivative of x^n is equal to n \\* x^(n-1). Starting with x^5 + x^9 - x^3 + 1, you can apply the power rule to the first two terms, which gives you: 5 \\* x^(5-1) + 9 \\* x^(9-1) You can then simplify the expression by combining like terms: 5 \\* x^4 + 9 \\* x^8 Finally, you can simplify the expression further by canceling out the common factor of x^4: 5 \\* 9 \\* x^4 = 45 \\* x^4

Therefore, the derivative of x^5 + x^9 - x^3 + 1 is equal to 45 \\* x^4. ✗

Human (with principle):
What is the derivative of x^5 + x^9 - x^3+1? let's think step by step

LLaMA-2-13B: find the derivative of the function, we need to take the derivative of each term in the function. The derivative of x^5 is 5x^4. The derivative of x^9 is 9x^8. The derivative of x^3 is 3x^2. The derivative of 1 is 1. Therefore, the derivative of the function is: 5x^4 + 9x^8 - 3x^2 + 1 = 5x^4 + 9x^8 - 3x^2 + 1 - (1) = 5x^4 + 9x^8 - 3x^2 Therefore, the derivative of the function x^5 + x^9 - x^3+1 is 5x^4 + 9x^8 - 3x^2. ✓

Figure 12: Correctness improvement on medium-scale LLaMA-2-13B model after using the introduced principle on prompts.

Human (without principle):
Luna, the poodle, is supposed to eat 2 cups of kibble every day. But Luna's master, Mary, and her husband, Frank, sometimes feed Luna too much kibble. One day, starting with a new, 12-cup bag of kibble, Mary gave Luna 1 cup of kibble in the morning and 1 cup of kibble in the evening, But on the same day, Frank also gave Luna 1 cup of kibble in the afternoon and twice as much in the late evening as he had given Luna in the afternoon. The next morning, how many cups of kibble will Mary find remaining in the bag?

LLaMA-2-13B: Luna was given a total of 3 cups of kibble on that day. Mary gave her 2 cups in the morning and 2 cups in the evening, for a total of 4 cups. Frank gave her 2 cups in the afternoon and 4 cups in the late evening, for a total of 6 cups. In total, Luna was given 10 cups of kibble on that day. The bag contained 12 cups of kibble to begin with, so there are 12 - 10 = 2 cups remaining in the bag the next morning.

✗

Human (with principle):
Luna, the poodle, is supposed to eat 2 cups of kibble every day. But Luna's master, Mary, and her husband, Frank, sometimes feed Luna too much kibble. One day, starting with a new, 12-cup bag of kibble, Mary gave Luna 1 cup of kibble in the morning and 1 cup of kibble in the evening, But on the same day, Frank also gave Luna 1 cup of kibble in the afternoon and twice as much in the late evening as he had given Luna in the afternoon. The next morning, how many cups of kibble will Mary find remaining in the bag? Explanation :

LLaMA-2-13B: Mary gave Luna a total of 2 cups of kibble that day. Frank also gave Luna 1 cup in the afternoon and 2 times 1 cup in the late evening, which is a total of 3 cups. So, a total of 5 cups of kibble were given to Luna that day. If there was a 12-cup bag of kibble, then there are 7 cups remaining in the bag the next morning.

✓

Figure 13: Correctness improvement on medium-scale LLaMA-2-13B model after using the introduced principle on prompts.

References

[1] Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. Ask me anything: A simple strategy for prompting language models, 2022. 4

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 3

[4] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. 3

[5] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023. 1

[6] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. 3

[7] Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*, 2023. 1

[8] Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. 4

[9] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023. 1

[10] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023. 7

[11] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. 1

[12] Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. Guiding large language models via directional stimulus prompting. *arXiv preprint arXiv:2302.11520*, 2023. 4

[13] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell,

Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2023. 1, 3

[14] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[16] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Ruther-

ford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446, 2021. 3

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. 3

[18] Zhiqiang Shen, Sondos Mahmoud Bsharat, and Aidar Myrzakhan. Atlas: A llm inquiry principle benchmark. *Preprint*, 2024. 7

[19] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020. 3

[20] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Martin Chadwick, Gaurav Singh Tomar, Xavier Garcia, Evan Senter, Emanuel Taropa, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Yujing Zhang, Ravi Addanki, Antoine Miech, Annie Louis, Laurent El Shafey, Denis Teplyashin, Geoff Brown, Elliot Catt, Nithya Attaluri, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys,

Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaly Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, Hanzhao Lin, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yong Cheng, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, YaGuang Li, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre

Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Gamaleldin Elsayed, Ed Chi, Mahdis Mahdieh, Ian Tenney, Nan Hua, Ivan Petrychenko, Patrick Kane, Dylan Scandinaro, Rishub Jain, Jonathan Uesato, Romina Datta, Adam Sadovsky, Oskar Bunyan, Dominik Rabiej, Shimu Wu, John Zhang, Gautam Vasudevan, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Betty Chan, Pam G Rabinovitch, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Sahitya Potluri, Jane Park, Elnaz Davoodi, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Chris Gorgolewski, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Paul Suganthan, Evan Palmer, Geoffrey Irving, Edward Loper, Manaal Faruqui, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Michael Fink, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marin Georgiev, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Minnie Lui, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Lam Nguyen Thiet, Daniel Andor, Pedro Valenzuela, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Sarmishta Velury, Sebastian Krause, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Tejasi Latkar, Mingyang Zhang, Quoc Le, Elena Allica Abellan, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Sid Lall, Ken Franko, Egor Filonov, Anna Bulanova, Rémi Leblond, Vikas Yadav, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Hao Zhou, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Jeremiah Liu, Mark Omernick, Colton Bishop, Chintu Kumar, Rachel Sterneck, Ryan Foley, Rohan Jain, Swaroop Mishra, Jiawei Xia, Taylor Bos, Geoffrey Cideron, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Petru Gurita, Hila Noga, Premal Shah, Daniel J. Mankowitz, Alex Polozov, Nate Kushman, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Anhad Mohananey, Matthieu Geist, Sidharth Mudgal, Sertan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Quan Yuan, Sumit Bagri, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Aliaksei Severyn, Jonathan Lai, Kathy Wu, Heng-Tze Cheng, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Mark Geller, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Andrei Sozanschi, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Abhimanyu Goyal, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Sabaer Fatehi, John

Wieting, Omar Ajmeri, Benigno Uria, Tao Zhu, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Dustin Tran, Yeqing Li, Nir Levine, Ariel Stolovich, Norbert Kalb, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Balaji Lakshminarayanan, Charlie Deck, Shyam Upadhyay, Hyo Lee, Mike Dusenberry, Zonglin Li, Xuezhi Wang, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Summer Yue, Sho Arora, Eric Malmi, Daniil Mirylenka, Qijun Tan, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Steven Zheng, Francesco Pongetti, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Ragha Kotikalapudi, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Chenkai Kuang, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Pei Sun, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Ishita Dasgupta, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Yuan Liu, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Ivo Penchev, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Adam Kurzrok, Lynette Webb, Sahil Dua, Dong Li, Preethi Lahoti, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Taylan Bilal, Evgenii Eltyshev, Daniel Balle, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Adams Yu, Christof Angermueller, Xiaowei Li, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Kevin Brooks, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Komal Jalan, Dinghua Li, Ginger Perng, Blake Hechtman, Parker Schuh, Milad Nasr, Mia Chen, Kieran Milan, Vladimir Mikulik, Trevor Strohman, Juliana Franco, Tim Green, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2023. 3

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[22] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier

Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 3

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 4

[24] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023. 4

[25] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. 7

[26] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. 4