FRI Natural language processing course 2024
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

LLM Prompt Strategies for Commonsense-Reasoning Tasks

Žan Počkar, Amer Mujagić, and Ivan Nikolov

Abstract

With the rise in LLM usage, diverse optimization techniques have emerged. In order to evaluate and compare the efficiency of different prompt strategies, we first needed to describe what commonsense-reasoning tasks are. After describing the tasks we made a selection of different prompt strategies which required us reviewing the different strategies and choosing the ones that seemed to be both in popular use and different enough so that these evaluations could be extended to different sub-types of mentioned strategies. After choosing the strategies, we also choose models and datasets to test the strategies on, and reviewed possible evaluation methods.

Keywords
Prompt Strategies, Prompt Evaluation

Advisors: Aleš Žagar

Introduction

Artificial language models especially large language models (LLMs) have in recent time vastly improved at tackling many tasks when provided with substantial training material. Yet enhancing their commonsense remains difficult. This paper explores approaches to nurture more natural reasoning in these models.

Commonsense reasoning, the ability to understand and make judgments based on everyday knowledge and experiences, is a crucial aspect of human intelligence. While humans effortlessly apply this knowledge in their daily lives, imparting this ability to LLMs is a complex task. This paper investigates methods such as Chain of Thought (CoT), in-context learning, and plan-and-solve techniques to enhance model performance in tasks requiring common sense knowledge. To better analyse the results we will start our research with a Unicorn LLM[1].

In this paper we will firstly designing experiments to evaluate the effectiveness of each strategy. Later we will analyzing Unicorns's reasoning processes, and document how different prompting techniques influence the outcomes. A key aspect of this work is the selection of an appropriate commonsense reasoning dataset. After considering several options, we choose the Rainbow [2] benchmark, as this benchmark is a universal commonsense reasoning benchmark that brings together six existing commonsense reasoning tasks.

Problem

Commonsense reasoning in Large Language Models (LLMs) is a complex problem due to the implicit and context-dependent nature of commonsense knowledge. For instance, consider the sentence “The ice cream was too hot to eat.” Humans intuitively understand this sentence is likely incorrect because ice cream is typically cold. However, an LLM might not flag this as an anomaly unless it has been explicitly trained on similar examples or has learned to associate certain objects with their typical properties.

Another challenge is the representation and utilization of commonsense knowledge in LLMs. LLMs learn from the patterns in the data they are trained on, so the size and quality of the training dataset significantly influence their performance. However, encoding commonsense knowledge into a machine-readable format that an LLM can learn from is not straightforward. For example, understanding that “people usually sleep at night” requires the model to learn not just the concepts of “people,” “sleep,” and “night,” but also the typical association between these concepts.

Lastly, the evaluation of commonsense reasoning in LLMs is a significant challenge. Traditional evaluation metrics may not adequately capture the nuances of commonsense reasoning. For instance, if an LLM generates the sentence “The man opened his umbrella because it was raining,” it demonstrates an understanding of the causal relationship between rain and using an umbrella. However, quantifying this understanding and comparing it across different models or tasks is not trivial, necessitating the development of more sophisticated evaluation metrics and benchmarks.

Prompting Techniques

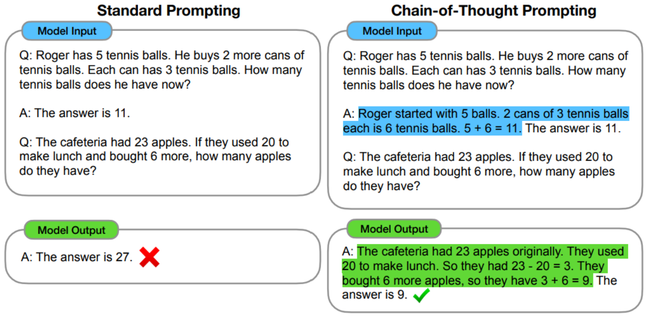
Prompts techniques are the proposed input structures that guide LLMs to produce outputs with the desired form and content quality. That is, to simplify, we can say that we use different prompting techniques to ”program” LLM inputs by following specific patterns in order to obtain an output. [3]

Few-shot prompting

Few-shot prompting [4] is a prompting technique that consists of adding examples of how a response to a problem would look a number of times before stating the problem we want solved. Depending on the number of examples we give we can more specifically define our technique as n-shot learning, where n is the number of examples we gave before asking the question. Techniques similar in concept to few-shot prompting are zero-shot prompting (just asking the question directly) or one-shot prompting (adding a single example to a prompt). Examples that show the differences between zero-shot, one-shot and few-shot prompts can be seen in Figure 3.

Chain of thought

Chain of Thought (CoT) [5] prompting involves adding a series of intermediate reasoning steps to the intended task. It has been shown to improve performance in more complex prompts. These intermediate reasoning steps can be thought of as giving an answer to a similar problem but with detailed steps on how to get to that answer as shown in Figure 1. This is similar to few-shot prompting, which also provides examples of answers to similar questions, however few-shot prompts include less detail that is they usually provide only questions and answers, no intermediate reasoning steps in between. [5]



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

A: The answer is 11.

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

A: The answer is 27. ✗

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

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

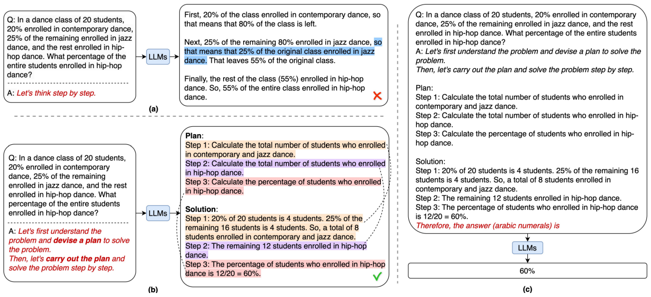
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

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

Figure 1. Example of a Chain of Thought prompt versus a regular prompt

Plan and solve

The plan and solve [6] prompting technique breaks up our prompt into multiple prompts. It starts with a prompt that constructs a plan to solve the stated problem, after which it uses the generated plan to guide the model into providing the final solution. An example of this approach is shown in Figure 2



Q: In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance?

A: *Let's think step by step.*

First, 20% of the class enrolled in contemporary dance, so that means that 80% of the class is left.

Next, 25% of the remaining 80% enrolled in jazz dance, so that means that 25% of the original class enrolled in jazz dance. That leaves 55% of the original class.

Finally, the rest of the class (55%) enrolled in hip-hop dance. So, 55% of the entire class enrolled in hip-hop dance. ✗

Q: In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance?

A: *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.*

Plan:
Step 1: Calculate the total number of students who enrolled in contemporary and jazz dance.
Step 2: Calculate the total number of students who enrolled in hip-hop dance.
Step 3: Calculate the percentage of students who enrolled in hip-hop dance.

Solution:
Step 1: 20% of 20 students is 4 students. 25% of the remaining 16 students is 4 students. So, a total of 8 students enrolled in contemporary and jazz dance.
Step 2: The remaining 12 students enrolled in hip-hop dance.
Step 3: The percentage of students who enrolled in hip-hop dance is 12/20 = 60%. ✓

Q: In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance?
A: *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.

Plan:
Step 1: Calculate the total number of students who enrolled in contemporary and jazz dance.
Step 2: Calculate the total number of students who enrolled in hip-hop dance.
Step 3: Calculate the percentage of students who enrolled in hip-hop dance.

Solution:
Step 1: 20% of 20 students is 4 students. 25% of the remaining 16 students is 4 students. So, a total of 8 students enrolled in contemporary and jazz dance.
Step 2: The remaining 12 students enrolled in hip-hop dance.
Step 3: The percentage of students who enrolled in hip-hop dance is 12/20 = 60%.
Therefore, the answer (arabic numerals) is

60%

Figure 2. Plan and solve prompt series example

Three of thoughts

Tree of Thoughts (ToT) [7] is a technique that is based one exploring multiple reasoning paths over thoughts. ToT frames any problem as a search over a tree, where each node is a state representing a partial solution with respect to the input and the sequence of previous thoughts. A specific construction of ToT can be divided into solving four sub-problems:

1. Decomposition of the intermediate process into thought steps
2. Potential thought generation based on each individual state
3. State evaluation
4. Choice of search algorithm

The differences between similar strategies are shown in Figure 3.

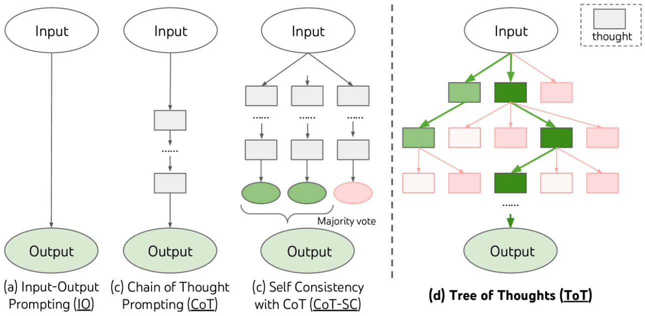
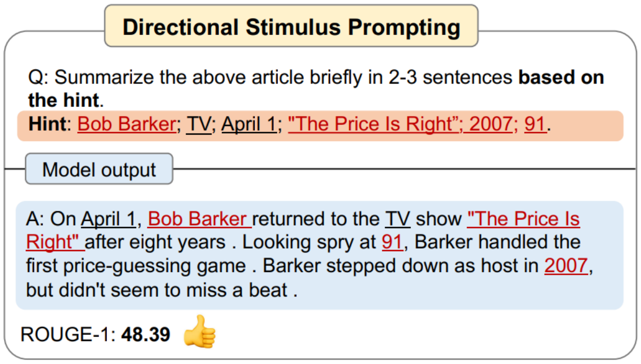

Input
Output
Input
Output
Input
Majority vote
Output
Input
thought
Output
(a) Input-Output Prompting (IO)
(c) Chain of Thought Prompting (CoT)
(c) Self Consistency with CoT (CoT-SC)
(d) Tree of Thoughts (ToT)

Figure 3. Differences between IO, CoT, CoT-SC and ToT

Directional Stimulus prompting

This prompting technique adds a component named ”directional stiumulus” to the prompt. The goal is to provide guidance for a specific prompt. This technique adds hints and clues to the input query to try and guide the model to producing a higher quality response. This method differs from other methods as it doesn’t add additional knowledge to the prompt. [8] An example of this approach is shown in Figure 4.



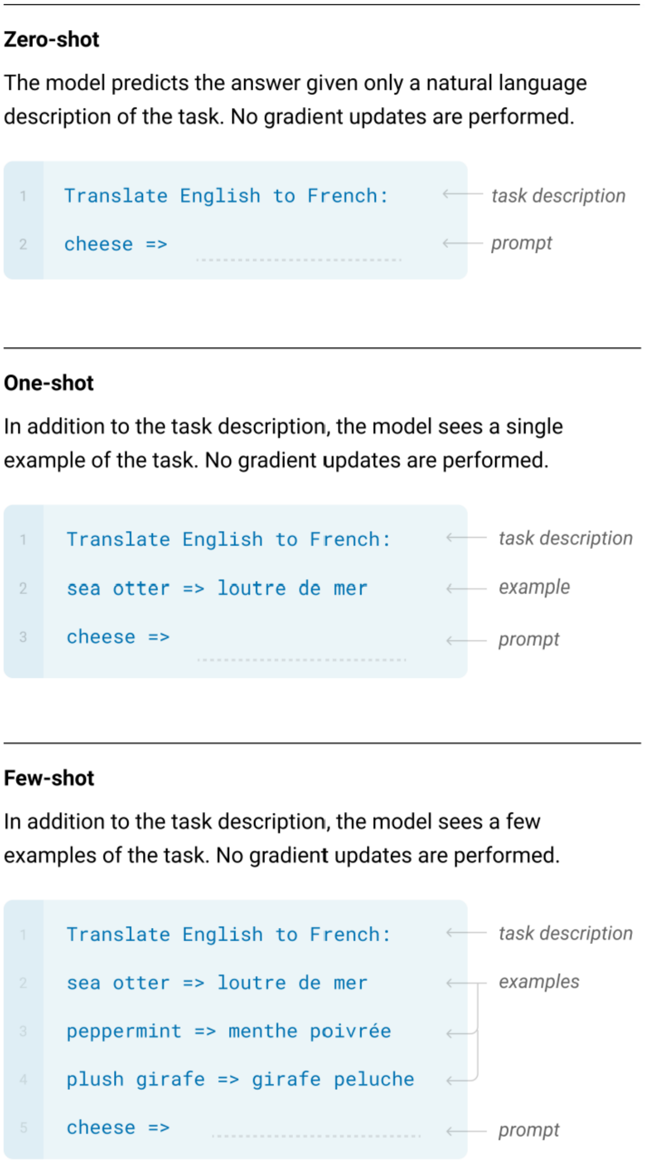
Q: Summarize the above article briefly in 2-3 sentences **based on the hint**.

Hint: Bob Barker; TV; April 1; "The Price Is Right"; 2007; 91.

A: On April 1, Bob Barker returned to the TV show "The Price Is Right" after eight years . Looking spry at 91, Barker handled the first price-guessing game . Barker stepped down as host in 2007, but didn't seem to miss a beat .

ROUGE-1: **48.39**

Figure 4. Example of directional stiumulus prompting



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Figure 5. Examples of zero-shot, one-shot and few-shot prompts

Datasets

Because commonsense reasoning is a complex topic and encompasses a wide collection of different tasks, it is difficult to generate a dataset that will evaluate the model's generalization abilities.

The Winograd schema challenge [9] was the standard for testing the commonsense reasoning, however from 2019 it is considered defeated since numerous transformer-based models achieved over 90% accuracy. [10] For that reason, we will use more recent and challenging datasets.

On the one side, we have different SOTA methods [5] evaluated on task specific datasets like GSM8K [11]. This dataset focuses on linguistically diverse grade school math problems created by human problem writers.

On the other side, there are benchmarks that combine multiple commonsense datasets. For this project we will use Rainbow which is a suite of commonsense benchmarks that contain multiple choice question-answering datasets. Rainbow contains the following datasets

- αNLI [12] - tests abductive reasoning in narratives. Models need to find the best explanation among the presented options connecting a beginning and ending;
- CosmosQA [13] — asks commonsense reading comprehensions about narratives in everyday situations
- HellaSWAG [14] — models need to find the most plausable ending to a short content.
- PIQA [15] — question answering benchmark for commonsense reasoning.
- SocialIQa [16] — commonsense reasoning about social situations and interactions
- WinoGrande [17] — large-scale collection of Winograd schema-inspired problems that test reasoning about social and physical interactions.

Other possibilities for testing commonsense reasoning are the GLUE [18] and SuperGLUE [19] datasets, which are benchmarks for general language understanding systems.

References

[1] Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. volume 15, 2021.

[2] Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Rainbow: A Commonsense Reasoning Benchmark. *arXiv preprint arXiv:2103.13009*, 2021.

[3] 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] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022.

[6] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models, 2023.

[7] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

[8] Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. Guiding large language models via directional stimulus prompting, 2023.

[9] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

[10] Hector J. Levesque. On our best behaviour. *Artificial Intelligence*, 212:27–35, 2014.

[11] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics.

[12] Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Yih, and Yejin Choi. Abductive commonsense reasoning. *ArXiv*, abs/1908.05739, 2019.

[13] Zhixing Tian, Yuanzhe Zhang, Kang Liu, Jun Zhao, Yantao Jia, and Zhicheng Sheng. Scene restoring for narrative machine reading comprehension. 2020.

[14] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? 2020.

[15] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. 2020.

[16] Gengyu Wang, Xiaochen Hou, Diyi Yang, Kathleen McKeown, and Jing Huang. Semantic categorization of social knowledge for commonsense question answering. 2021.

[17] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. 2020.

[18] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. 2018.

[19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. volume 32, 2019.