University of Ljubljana
Faculty of Computer and
Information Science

FRI Natural language processing course 2024
Project report

LLM Prompt Strategies for Commonsense-Reasoning Tasks

Matic Pristavnik Vrešnjak, Mitja Kocjančič, and Songeun Hong

Abstract

The surge in the popularity of Large Language Models (LLMs) such as chatGPT, PaLM, and Gemini has led to their widespread adoption in both personal and commercial domains. Many of these cutting-edge models rely on the transformer architecture. With the increasing use of LLMs, there is a growing need to devise prompts that facilitate the generation of relevant and informative responses, particularly for tasks necessitating commonsense reasoning. Such tasks draw upon everyday knowledge for resolution. Consequently, various prompt strategies have emerged to enhance model performance on such tasks. In response to the expanding array of prompt strategies, this paper offers a comprehensive comparison of each approach, aiming to shed light on their effectiveness and applicability in enhancing LLM performance.

Keywords
Large Language Models (LLMs), Prompt Strategies, Comparative Analysis

Advisors: Slavko Žitnik

Introduction

Recent years have seen a rise in the popularity of Large language models (LLMs). LLMs such as chatGPT, PaLM, Gemini, and others have already been adopted for personal and commercial use. Many of the current state-of-the-art models are based on the transformer architecture[1]. Because of the wide adoption of LLMs, it has become more important to provide prompts that encourage the model to generate relevant and informative responses to tasks that require commonsense reasoning. Tasks that require commonsense reasoning are those that require everyday knowledge to solve. This has led to the development of several strategies for providing prompts that allow models to perform better on such tasks. Due to the growing number of prompt strategies, we present a comprehensive comparison of each strategy.

Related work

One strategy for providing prompts is Chain of thought (CoT) [2]. The core idea of CoT is that we give the model a demonstration of how to come to the solution or ask it for a step-by-step response. CoT has been proven to improve the performance of LLMs on commonsense reasoning and mathematical datasets. One limitation mentioned in the original CoT paper was that CoT only benefits large models like chatGPT and not smaller models. Frameworks like DSPy[3] have also integrated functionality that enables the implementation of custom Chains of thought. Strategies with similar approaches are In-context Learning[4] and Plan-and-Solve Techniques[5].

Another example of prompt strategies is Promptbreeder[6]. In contrast to methods like CoT, which are based on hand-crafted prompts, Promptbreeder can generate optimized prompts. Promptbreeder relies on LLM and genetic algorithms to mutate a set of prompts, which are then given to the LLM. Strategies that use generated prompts have been shown to perform better than the prompt strategies that are based on hand-crafted prompts.

The aforementioned strategies are more complex and require additional text to be added to the instruction. Simpler strategies are more oriented towards creating prompts that are clear and precise[7]. This includes forming unambiguous and specific prompts, allowing the model to give better and more detailed answers. Research has also shown that models can provide better answers when given emotional prompts[8].

Previously mentioned strategies have been tested on various LLMs. LLMs that are most commonly used for testing are chatGPT[9], PaLM[10], Gemini[11] and other models. Most researchers prefer to use models with a large number of parameters because larger LLMs benefit more from prompt engineering. This has led to very little research being done on smaller models.

Because of the growing number of prompting strategies, many datasets have been used to measure their performance. Examples of such datasets are the Winograd Schema Challenge[12], Textbook Question Answering[13], SocialIQA[14], and many other datasets. Many of the mentioned datasets were made to challenge LLM on commonsense sense reasoning tasks.

Methodology

Various prompting strategies were tested on models obtained from the Hugging Face library. The largest models for text generation on Hugging Face have around 150 billion parameters. Unfortunately, we couldn't use these models because of the limited data storage on the HPC cluster.

Mistral, developed by Mistral AI, despite its relatively small size (only 7B parameters),it has demonstrated comparable performance to models like LLama3 and GPT-3.5. One notable drawback of this model is its tendency to suffer from hallucinations, which can cause it to veer off-topic.

ROUGE, a set of metrics commonly used in natural language processing, evaluates performance in text summarization and translation. The metrics offered by ROUGE include:

- Precision, measures the accuracy of positive predictions made by a model, calculated as the ratio of true positive predictions to the total number of positive predictions. A perfect score is 1, and the worst is 0.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

- **Recall**, also known as sensitivity or true positive rate, evaluates a model's effectiveness in identifying all positive samples. It is computed as the ratio of true positive predictions to the total number of actual positive samples, with an ideal score being 1 and the worst being 0.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2}$$

- **The F1 score**, a metric that combines precision and recall, offers a balanced assessment of model performance. It reaches its optimal value at 1 and its lowest at 0. It integrates both precision and recall, making it suitable for scenarios where a balance between these metrics is desired as shown in the equation below:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

BLEU is a scoring algorithm commonly used to assess the quality of machine-translated text. Text with scores close to 0 is typically considered poor, while scores closer to 1 are deemed good.

The datasets employed to assess various prompt strategies were obtained from Hugging Face. The datasets used to test various prompt strategies were obtained from Hugging Face. They were selected based on the task domains they covered, such as mathematics, question answering (Q&A) and more. Noteworthy datasets utilized thus far include:

- **The Winograd Schema Challenge** is a dataset frequently used to evaluate the performance of Large Language Models (LLMs). It comprises 285 sentences, each containing two phrases and an ambiguous pronoun. The model is tasked with determining which phrase better fits the sentence context. An example of such a prompt is illustrated in 1.

```
Classify the text into A or B based on pronoun
    .
    Text: Joan made sure to thank Susan for
        all the help she had recieved.
    Options:
    A) Joan
    B) Susan
    Answer:
    A
```

Figure 1. Example of prompt and answer for winogrand schema challenge

- **The multiArith** dataset is often utilized to evaluate a model's capability to perform multiple arithmetic operations for each question. An example of a prompt and its corresponding answer from the dataset is depicted in Figure 2. This dataset encompasses approximately 400 questions of similar nature.

```
Q: A pet store had 64 puppies. In one day they
     sold 28 of them and put the rest into
    cages with 4 in each cage. How many cages
    did they use?
A:9
```

Figure 2. Example of prompt and answer for multiArith dataset

There are numerous popular prompting strategies. Among those we've evaluated are:

- **Zero-shot** This strategy relies on the model being adequately trained and exposed to extensive amounts of data. Typically, minimal changes are made to the prompt, usually involving only formatting. One improvement to this strategy involves appending "Let's think about it" to the end of the prompt to encourage the model to solve the problem step by step. An example of a zero-shot prompt is shown in Figure 3.

- **Few-shot** This approach enhances the prompt by incorporating examples of previously solved problems. It's important to note that this doesn't include steps on how to solve the problem, but rather presents the question and its corresponding answer. The number of solved problems added to the prompt can vary but is typically

```
Classify the text into neutral, negative or
    positive.
Text: I think the vacation is okay.
Sentiment:
```

Figure 3. Example of zero-shot prompt strategy with improvement

```
This is awesome! // Negative
This is bad! // Positive
Wow that movie was rad! // Positive
What a horrible show! //
```

Figure 4. Example of few-shot strategy

around 4 or 5. An example of this strategy is depicted in Figure 4.

- **Chain-of-thought** This strategy is often combined with few-shot. It relies on adding steps to the examples provided to the model. A simple example of this strategy can be seen in Figure 5.

```
The odd numbers in this group add up to an
    even number: 4, 8, 9, 15, 12, 2, 1.
A: Adding all the odd numbers (9, 15, 1) gives
     25. The answer is False.

The odd numbers in this group add up to an
    even number: 15, 32, 5, 13, 82, 7, 1.
A:
```

Figure 5. Example of chain-of-thought strategy prompt

Results

Quantitative analysis

Dataset	Zero-shot			Improved zero-shot			Few-shot			Chain-of-thought		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Winogrand	0.34	0.64	0.41	0.26	0.67	0.36	0.20	0.80	0.32	0.24	0.78	0.37
multiArith	0.16	0.65	0.24	0.23	0.65	0.34	0.086	0.38	0.13	0.17	0.62	0.26

Table 1. Performance Metrics for Different Strategies

In Table 1, it's evident that the improved zero-shot prompt performs best on the following datasets. This is indicates that the model generates more accurate text when this prompting strategy is used. Compared to other strategies, the prompt's length remains relatively small, which is advantageous for smaller models since they have fewer attention heads. Applying the chain-of-thought strategy typically yields more relevant text, as indicated by the higher recall score. However, few-shot with four added examples performed the worst. This is because when this strategy was applied, the model would often return irrelevant information or, in some cases, attempt to explain already solved problems. This cloud be the result of improper model parameter settings on our part or because small models aren't able to fully utilize all examples because of there small size.

Qualitative analysis

Zero-shot has proven to be quite unreliable, especially when using small models where the architecture isn't robust enough to compensate for unoptimized prompts. Another issue with this strategy is its heavy reliance on the model having seen a large amount of data, which can lead to the model going off-topic or providing incorrect answers. Examples of poor outputs produced by this model can be observed in Figures 6. These responses are not only incorrect but also somewhat unfocused. When applied to mathematical problems, this strategy often results in even more incoherent responses that are completely unrelated to the question.

```
Classify the text into A or B based on pronoun.
    Text: The city councilmen refused the
        demonstrators a permit because they
        advocated violence.
    Options:
    A) The city councilmen
    B) The demonstrators
    Answer:
    1. Identify the subject and object of the
        sentence.
        Subject: The city councilmen
        Object: a permit
    2. Determine which pronoun is used to refer to
         each noun.
        A) The city councilmen -> They
        B) The demonstrators -> They
    3. Check if the pronouns match.
        They (city councilmen) = They (
            demonstrators)
    4. Classify the text as A or B based on the
        matching pronoun.
        Since both 'They' refer to different
            groups, this question should be
            classified as B. However, since the
            correct answer is A, there seems to be
             an error in the options provided.

    Corrected answer: A
```

Figure 6. Example of a bad zero-shot response

When zero-shot is enhanced with a simple chain-of-thought strategy, it produces much better and more on-topic responses. In our case, we improved it by simply adding "Solve the problem step by step, please". The model's response can be seen in Figure 7. This demonstrates that asking the model to show its reasoning steps leads to improved performance. However, it's important to note that even with this improvement, the model can still produce strange and nonsensical outputs. One such example is when the model concludes that both options are correct or that no option is correct. This strategy shows the largest improvement on mathematical problems because we typically solve mathematical problems by thinking in steps.

When few-shot was employed to enhance prompts, the model demonstrated significantly better performance in staying on topic compared to the previous two strategies. This improvement was particularly notable in questions related to mathematical problems. However, one weakness of few-shot

```
Classify the text into A or B based on pronoun.
    Solve the problem step by step please
    Text: The city councilmen refused the
        demonstrators a permit because they
        advocated violence.
    Options:
    A) The city councilmen
    B) The demonstrators
    Answer:
Generated output
1. Identify the subject and object of the sentence
    .
        Subject: The city councilmen
        Object: They (demonstrators)
    2. Determine which group used the pronoun "
        they" in the sentence.
        Pronoun "they" refers to the demonstrators
            .
    3. Based on the pronoun, classify the text.
        Since the pronoun "they" refers to the
            demonstrators, the text should be
            classified as B.
```

Figure 7. Example of a good zero-shot response when a simple chain-of-taught

is that beyond a certain number of examples, the strategy may not provide additional benefits and could potentially lead to worse performance. Additionally, this strategy requires the examples to be somewhat similar to the question, as disparate examples might mislead the model. The example answer also have to be other wise the model gives the same answer for every question.

On the other hand, when the chain of thought was utilized as a prompting strategy, the model's results were notably more accurate. The generated text was more concise and focused, with less meandering from the model. Similar to few-shot, this method also requires the chains of thought to be somewhat relevant to the problem the model is attempting to solve.

Conclusions and future plans

In this report, we presented the impact of each prompting strategy and briefly outline their respective benefits. We conducted both quantitative and qualitative analyses for each strategy. Our future goals include:

1. Gather more datasets for testing prompt strategies: Expanding the variety of datasets will allow us to evaluate the effectiveness of prompting strategies across different domains.

2. Improve quantitative analysis: Our current analyses are somewhat unreliable, so we aim to enhance them by incorporating additional metrics such as accuracy, relevancy, and others.

3. Addressing issues with small models: We observed that small models are prone to producing strange outputs when the input prompts are lengthy. Therefore, we plan to repeat the experiments using larger models to mitigate this issue.

4. Implementing more complex prompting strategies: We intend to explore and implement more advanced prompting strategies such as self-consistency and automatic chain-of-thought, which could potentially further improve model performance.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[2] 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.

[3] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

[4] Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey, 2024.

[5] 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.

[6] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution, 2023.

[7] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2023.

[8] Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli, 2023.

[9] OpenAI et al. Gpt-4 technical report, 2024.

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran,

Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[11] Gemini Team et al. Gemini: A family of highly capable multimodal models, 2023.

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

[13] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384, 2017.

[14] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019.