Investigating Prompt Engineering in Large Language Models

Tom Gallagher

Introduction

The aim of this experiment is to compare the effectiveness of different prompt engineering strategies in the domain of algebraic word problems. Prompt engineering is a valuable tool in a world where Large Language Models (LLMs) are being rapidly integrated into many industries and our everyday lives. Prompt engineering can be particularly valuable when LLM resources are limited, with the potential to bring the performance of small models up to the standard of larger ones using carefully structured prompts. In this study, the domain of algebraic word problems provides reasoning-heavy tasks, and different prompting techniques provide different levels of information to the tested LLM, or encourage it to follow a certain process in its solution. The metrics gathered in this study allow analysis of the effectiveness of each prompting technique in a reasoning-heavy domain.

Hypotheses

I hypothesise that, because of the reasoning-heavy nature of the problem domain, reasoning-based prompting techniques will increase the accuracy of the LLM in solving algebraic word problems, and will outperform example-based techniques.
I also expect that this effect will be more pronounced as problem difficulty increases, and reasoning-based prompting techniques will be more robust to problem difficulty than example-based techniques.

Research Summaries

A systematic survey of prompt engineering in large language models: Techniques and applications.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024)

This paper provides a structured survey of prompt engineering techniques for large language models. Prompt engineering is presented as the process of carefully designing inputs to guide model behavior, with the goal of improving accuracy, controllability, and efficiency across a variety of tasks. The paper categorizes techniques into major families, including zero-shot, one-shot, and few-shot prompting, reasoning-oriented methods such as chain-of-thought, and more advanced strategies like generated knowledge and instruction-based prompting.

The authors examine how these methods perform across different application domains, including question answering, reasoning, text generation, and summarization. They also highlight common challenges, such as prompt sensitivity (where small wording changes lead to large performance differences), lack of robustness, and the computational cost of experimenting with many prompt variations. Case studies demonstrate that prompt engineering can bring smaller models up to the standard of larger ones.

Overall, the survey concludes that prompt engineering is critical for effectively using LLMs, especially in research and applied settings where fine-tuning may not be feasible. It also emphasizes that reasoning-focused prompting methods, such as chain-of-thought and generated knowledge, play a particularly important role for complex problem-solving. The paper suggests that future work should focus on developing systematic frameworks and automated tools to allow prompts to be reproduced using the best prompting strategies.

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022).

This paper investigates whether or not LLMs can use multi-step reasoning if their prompts include a chain of thought. This prompting approach uses intermediate steps, attempting to encourage the model to follow a more human-like solving process.

Chain-of-thought prompts included worked-out reasoning steps for examples, rather than just Q&A pairs like standard shot-style prompting. For this experiment, chain-of-thought prompting was evaluated using few-shot prompting as a control. The effects of chain-of-thought prompting were also evaluated across a range of models at different scales, using problems of varying complexity. Problems for this experiment were sourced from the GSM8K dataset as well as SVAMP, ASDiv, AQuA and MAWPS.

The results of this experiment showed that chain-of-thought prompting allowed PaLM 540B to outperform the state-of-the-art model at the time (ChatGPT with a verifier). The differences noticed with chain-of-thought prompting were more significant as problem size increased, indicating that chain-of-thought prompting may help keep the model on track as it moves through multiple numerical steps in a word problem. Analysis also determined that chain-of-thought prompting was an emergent property of model scale, as very small models produced illogical or nonsensical chains of reasoning. Overall, this study suggests that chain-of-thought prompting is an effective prompting technique, particularly in the domain of mathematical reasoning. This highlights the value of reasoning-focused prompt engineering.

Description of Process

For this investigation, I applied a range of prompting techniques with a locally run Ollama model (llama3.1:8b) to solve maths word problems from the GSM8K dataset. This dataset contains maths word problems of varying difficulty, presented in natural language. As stated in the dataset description, “solutions primarily involve performing a sequence of elementary calculations using basic arithmetic operations (+ − ×÷) to reach the final answer.”

The set of prompting techniques used includes both example-based and reasoning-based techniques, as well as zero-shot prompting, which was used as a baseline/control.

The example-based techniques will include:

- One-Shot Prompting
- Few-Shot Prompting (3 shots)

The reasoning-focused techniques will include:

- Chain-Of-Thought Prompting
- Generated Knowledge Prompting

Llama3.1:8b was chosen as it is free and easy to run locally, and is not very powerful. For the purposes of this study, it is best to have a small model, as the effects of prompt engineering will be more pronounced when the model has a low baseline accuracy with zero-shot prompting.

The prompts were generated and performed in python, which allowed for an increased dataset size as compared to manual prompting. The methods used to generate each type of prompt are detailed in the prompt generation specification (Appendix A). A total of 70 problems were provided to the model, using all 5 prompting methods. For each trial, the correctness of the response was saved, providing a total accuracy level for each prompting method across the full dataset. Each problem was also saved in plain text, to have its difficulty determined by ChatGPT 5 based on the number of algebraic steps. The automatic determination of the number of steps was supervised and sanity checked.

This range of prompting techniques and metrics allows enough data for:

- Comparing the value of example-based and reasoning-focused prompting techniques.
- Identification of the most effective prompting techniques for math word problems.
- Analysis of how each prompting technique performs as problem difficulty increases, using the number of algebraic steps determined by ChatGPT 5 as a measure of difficulty.

Results

Overall Accuracy by Prompting Technique

The accuracy across all of the problems was evaluated first for each technique without considering difficulty.

- Reasoning-based prompting techniques (generated knowledge and chain-of-thought) received much higher accuracy scores than example-based prompting techniques
- Example-based techniques outperformed the control (Zero-shot)\

The table and figure below display the overall accuracy levels as detailed above.

Technique	Accuracy (%)
Generated Knowledge	**82.9**
Chain-of-Thought (CoT)	81.4

One-shot	47.1
Few-shot	40.0
Zero-shot	28.6

Table 1 - Accuracy by Prompting Technique

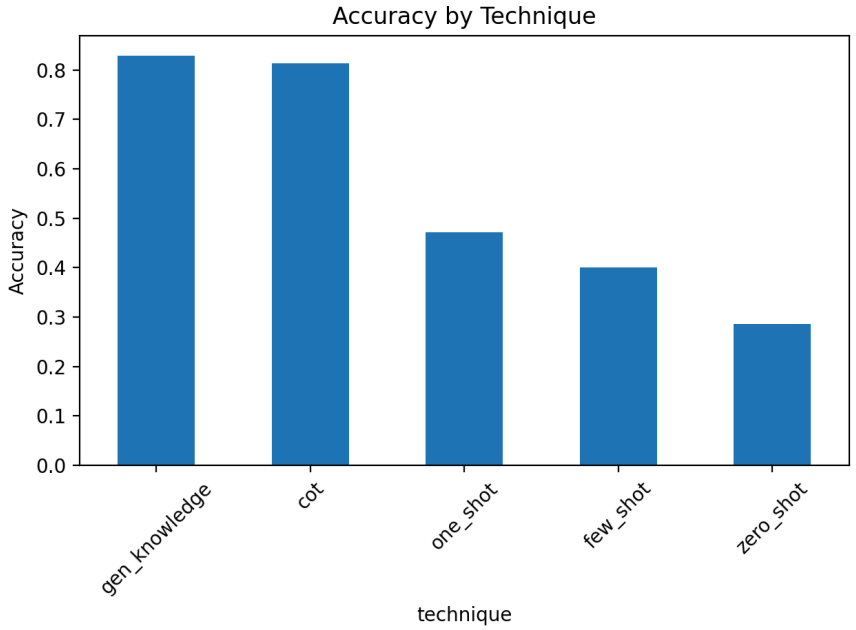

Figure 1 - Accuracy by Prompting Technique

Accuracy Across Difficulty Levels by Prompting Technique

Secondly, the problems were split by the number of algebraic steps required, categorising them as easy or hard. Problems were classified as “hard” if they required at least 4 algebraic steps to solve. The percentage decrease from the accuracy level for easy problems to the accuracy level for hard problems was also determined, to determine the robustness of each technique.

- Reasoning-based prompting techniques, generated knowledge in particular, showed much smaller percentage decreases than example-based prompting techniques.
- Example-based techniques became significantly less accurate when presented with harder problems
- Zero-shot prompting showed an increase in accuracy for harder problems, but was still the least accurate method for problems of any difficulty level.

The table and figures below show the accuracy levels for each prompting technique across problems of each difficulty class, and the percentage drop from easy to hard problems for each prompting technique, as detailed above.

Technique	Easy Accuracy (%)	Hard Accuracy (%)	Percentage Decrease
Chain-of-Thought (CoT)	**91.3**	76.6	16.5
Generated Knowledge	87.0	80.9	7.0
One-shot	56.5	42.6	24.7
Few-shot	47.8	36.2	24.2
Zero-shot	26.1	29.8	**–14.0** (increase)

Table B - Accuracy for Difficulty Levels by Prompting Technique Plot

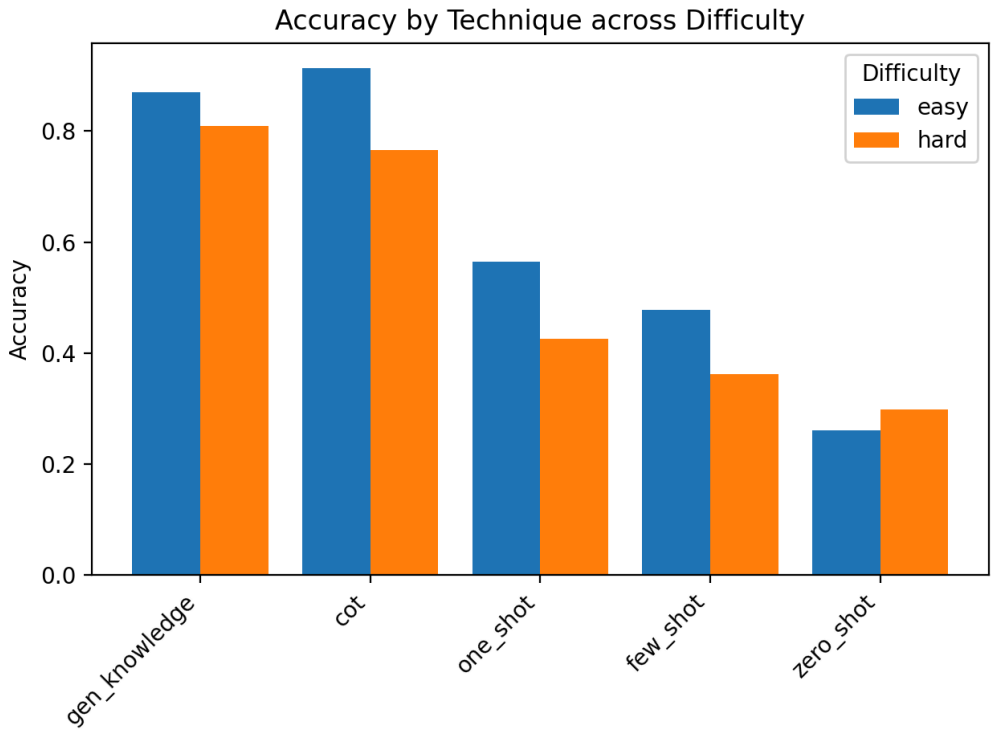

Figure 2 - Accuracy for Difficulty Levels by Prompting Technique

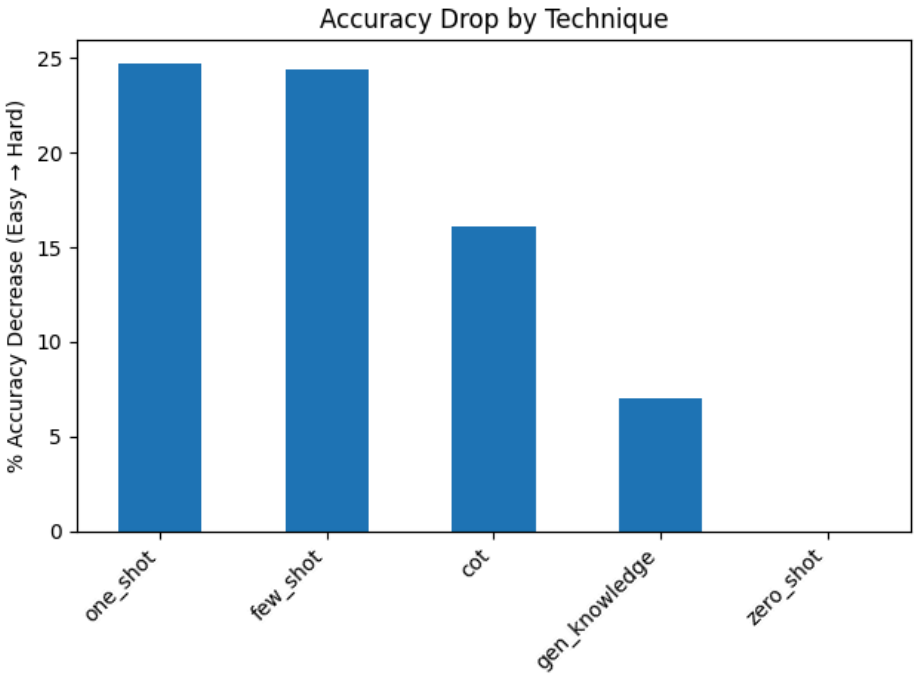

Figure 3 - Percentage Decrease in Accuracy for Difficulty Levels by Prompting Technique

Analysis & Discussion

Across all 70 problems, the two reasoning-based prompting techniques, being chain-of-thought prompting and generated knowledge prompting, received significantly stronger accuracy scores than the example-based prompting techniques and the control technique (Zero-shot). This indicates that there is value in encouraging LLMs to follow a human-like process to solve problems in this domain. With the problems requiring multiple steps and being presented in natural language, it is no surprise that reasoning-based prompting is effective. Generated knowledge prompting seems to have improved the LLM's ability to determine useful numeric information in the problem, and chain-of-thought prompting seems to encourage the LLM to identify the correct steps and perform them with the relevant numeric information.

As the model is presented with more complex problems, the value of reasoning-based prompting becomes even clearer. Both reasoning-based prompting techniques show less significant declines in performance than the example-based techniques, which suggests that reasoning-based prompting techniques may be more robust as problem difficulty increases.

These results together indicate that reasoning-based prompting is an effective technique to apply to the domain of algebraic word problems, particularly when trying to solve problems of higher complexity. It is also worth noting that the model used for this study is quite small, and it is likely that the effects of prompt engineering would be less pronounced for a more

powerful model, which should be able to achieve a significantly higher baseline accuracy score with just zero-shot prompting.

This investigation has several limitations that should be considered when interpreting the results. First, the experiments were conducted on a relatively small subset of 70 problems from the GSM8K dataset. While this sample was sufficient to highlight differences between prompting strategies, a larger and more diverse set of problems would provide stronger evidence and reduce the risk of sample-specific effects. The classification of problem difficulty relied on automated step-counting from model outputs. Although this provides a practical way to separate “easy” and “hard” problems, the method may not perfectly capture true difficulty, especially for problems that involve conceptual reasoning rather than sequential arithmetic steps (although the dataset description indicates they should contain mostly simple steps). Accuracy was measured using a strict binary metric of correct versus incorrect final answers. This does not capture cases where a model produced mostly correct reasoning but made a small arithmetic slip, nor does it account for the quality of intermediate reasoning steps. As a result, the analysis may understate the benefits of reasoning-focused prompting, which can improve interpretability even when final answers are wrong. It is also worth noting that the results of this study may not generalise to mainstream models that are much more powerful.

Conclusion

Reasoning-based prompting methods should be considered an effective prompt engineering strategy in the domain of algebraic word problems. This domain lends itself to multi-step reasoning, and correct identification of relevant numeric and operational information from natural language, which can be encouraged in LLMs using reasoning-based prompt engineering techniques, such as chain-of-thought prompting and generated knowledge prompting, which both proved to be effective in this study.

While example-based prompting does offer performance increases when compared to zero-shot prompting, it was not only less accurate than reasoning-based prompting in this domain, but also appeared to be less robust as problem complexity increased.

Ultimately, this study highlights the value of prompt engineering when LLM resources are limited, particularly in domains in which reasoning is important, such as the algebraic word problems used for this study.

For future work, it would be valuable to address the limitations of this study. Increasing the subset of data used would provide stronger evidence of the relationship between prompting technique and accuracy. Development of stronger evaluation methods could also provide more insights into the value of each technique. It would also be interesting to expand the problem domain, to identify which techniques are most effective for which type of problem.

This would also require further evaluation metrics to assess responses in domains where numerical correctness is not applicable. It could also be beneficial to perform a similar study with multiple LLMs, to see how the effects of prompt engineering translate to more capable models.

References

[1] Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *https://www.researchgate.net/publication/378183279_A_Systematic_Survey_of_Prompt_Engineering_in_Large_Language_Models_Techniques_and_Applications*

[2] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html*

Appendix A

This Github repository contains:

- A pdf document detailing the specification for how prompts are generated
- A script to generate and perform prompts locally with Ollama
- Two scripts for analysing results and plotting accuracy scores.
- The saved prompt and response data.
- Numeric and visual representations of the analysed data
- The GSM8K test dataset
- A copy of this report

Generative AI was used to create the skeletons of the scripts, which I adjusted to ensure they represented the method of my study correctly, and provided the desired metrics and graphs.