Midterm Report: The Practicality of Prompt Engineering

Isaac Liu
University of California, Berkeley
ijyliu@berkeley.edu

Abstract

This paper examines the practicality of prompt engineering in improving the performance of Large Language Models (LLMs). Through empirical analysis, we evaluate the trade-offs between costs and benefits of prompting using novel metrics. Different prompting methods are assessed using standardized tasks and both modern and older models.

Introduction

Prompt engineering, the practice of developing specialized prompts and queries to improve the accuracy of Large Language Models after training, is a prominent topic of interest in the NLP community, and among the general public. The practice is believed to allow for improvements in LLM performance a variety of domains without investment in underlying training (Martineau, 2021). It is not, however, without its critics. Some commentators believe that the practice will become irrelevant as models grow larger and more powerful, becoming more capable of directly interpreting a user's intent. (Ethan Mollick [@emollick], 2023). Others question the need for specialized professionals or training to attain minimal improvements which are often not repeatable across contexts (Shackell, 2023; Acar, 2023).

Despite such controversy, it is difficult to find empirical analyses of the tradeoff between costs and accuracy benefits associated with advanced prompting. Papers introducing new prompting techniques often only include performance benchmarks concerning the techniques' efficacy, typically within a limited domain. Some authors briefly mention problems associated with human-tailored prompts, such as the increased complexity induced by prompt-chaining and limitations on creativity and randomness (Wu et al., 2022), and others suggest the automation of prompting to avoid these costs (Diao et al., 2023), but the extent of these issues has not been (to my knowledge) quantified. Online marketplaces such as Promptbase provide input token costs for prompt texts sold on the platform, but do not provide any other information.

Briefly some practical considerations for prompt engineering On the cost side there are token costs, the filling of context window lengths leading to lower performance quality Dubious benefits due to a lack of research for some anecdotal techniques and decreased effectiveness with larger models but does not quantify these measures for specific techniques `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4504303`

The quality, length, and complexity of LLM responses have been analyzed within several individual task and technique domains.

Some research with GPT-3 series models on math and non-math reasoning tasks suggests the addition of complexity through the introduction of extra reasoning steps for both input prompts and output responses improves performance when using chain-of-thought prompting techniques https://browse.arxiv.org/pdf/2210.00720.pdf. Effects are on the magnitude of several points of accuracy per added step, with generally low costs as long as prompt examples are selected carefully. Improvements from complex chain-of-thought prompting are not fully accepted in the literature, however - some research has noted a tendency for the method to lead to degraded performance on simple questions https://browse.arxiv.org/pdf/2302.12822.pdf.

Costs and complexity are important!

The Chain of Density prompting technique seeks to optimize the named entity density of generated summaries through choice from a series of repeated, increasingly dense iterations https://browse.arxiv.org/pdf/2309.04269.pdf. Human preferences tend to align with 0.1 - 0.15 named entities per token, a point near the middle of the usual sequence of generations, demonstrating the existence of a tradeoff between informativeness and clarity.

Response to why can't you just ask for a certain level of length/complexity: It is difficult to control language model output length/complexity on tasks: Research also demonstrates current models are not yet able to achieve compliance with desired readability and complexity instructions for the tasks of story generation, simplification, and summarization, though a small amount of improvement is achievable through careful prompt word choice and the use of few-shot examples https://aclanthology.org/2023.acl-srw.1.pdf https://browse.arxiv.org/pdf/2309.05454.pdf.

This paper uses several metrics to evaluate the costs of prompt engineering methods systematically, and analyzes the tradeoffs inherent in their application to standardized data. Such an assessment is valuable on several dimensions. Beyond quantifiably testing the practicality of prompt engineering as a whole, it can be used to compare the performance of different approaches, useful in a world where so many competing techniques are available. I also offer a new look at these prompting methods in a period long after ideas were introduced and in an enviroment with greater capabilities in underlying models. Finally, I introduce and adapt some useful measures of costs and complexity, such as the ratio of interaction length with prompting to the length of an accepted human-generated answer, to the challenge of LLM evaluation.

I provide a newly constructed dataset summarizing the wide variety of existing techniques and data on their popularity as measured by Semantic Scholar citations, which may be useful for future surveys of the field

Prompting Methods Assessed

This list is subject to change, pending further assessment of popularity, implementation difficulty, and potential accuracy gains for each method. Higher accuracy methods are likely to be more interesting.

The below items are listed in order of increasing complexity/human intervention:

- Zero-Shot Control Baseline/Direct Prompting: Providing the question/task directly.

- Zero-Shot Chain of Thought Prompting: Existing literature mentions several examples of this; a simple one to append to every initial question/task, found to be optimal through automated testing is "Let's work this out in a step by step way to be sure we have the right answer." (Hebenstreit et al., 2023; Zhou et al., 2022)

- Tree of Thought Prompting: This prepends the following to the task/question: "Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realises they're wrong at any point then they leave. The question is..." (Hulbert, 2023)

- Generated Knowledge Prompting: The LLM is first prompted to generate some related facts and knowledge about the task/question. It is then prompted to answer the task/question. (Liu et al., 2022) The knowledge generating and task/question models could be different, but for my purposes I use the same model.

- Chain of Verification Prompting: The LLM is prompted a series of times to produce an initial baseline response, write its own verification questions, answer those verification questions, and write a final, verified response. (Dhuliawala et al., 2023)

- Few-Shot Prompting: The prompter provides a few examples of successfully/answered questions or tasks before the main question/task. Despite the work involved in implementing this method, I believe it has potential to be effective for two reasons. First, prior evidence indicates that larger and more modern language models benefit more from few-shot learning, potentially making this a consistently useful technique. (Brown et al., 2020) Second, earlier research has found that the formatting and input/label space distribution is more important than example correctness, meaning this method is somewhat robust to human error. (Min et al., 2022)

Metrics

Accuracy

Improved accuracy can be a benefit of prompt engineering. I report:

- Correct/Incorrect accuracy at the point a technique has been fully implemented (the end of the chain of thought, etc., modelling the real world)

Length

The length of responses and interactions can effect the practicality of prompting. It could indicate that a model is carefully and correctly solving through the steps of a problem (though it may actually be a confounder of other factors in such cases https://browse.arxiv.org/pdf/2210.00720.pdf). It can also impose time and financial costs to users, or become an indicator of degraded performance as models sometimes tend to go off on tangents or repeat themselves (to the extent some platforms have imposed length limitations) (Mann).

Prompt length is helpful up to 20 tokens, detrimental past 100 tokens. p.5 of https://arxiv.org/pdf/2104.08691.pdf

I report:

- Length of the entire interaction in tokens
- Financial cost of the entire interaction in tokens
- Length of the entire interaction in tokens relative to the length of the task/question + a human/solved out/generally accepted as correct answer (or relative to direct prompting). How much is prompt engineering stretching the interaction out? This ratio can be informative.
- The change in accuracy (in percentage points, 0 to 100) divided by the change in tokens (difference in token counts), between the prompt engineering technique and direct prompting. Is any stretching of output adding value/improving accuracy?

$$\frac{Accuracy_{PE} - Accuracy_B}{Tokens_{PE} - Tokens_B}$$

- Length of the entire interaction in time (seconds). This can include time writing a response, waiting for a response, or reviewing a response. More granular data on these each of the component steps may be hard to collect, but it might be possible to look at human assessments of time spent on these activities. Attempts will be made to have queries to models made at a consistent time during off-peak hours to minimize confounding due to server load, connectivity issues, etc.

Complexity

Similarly to length, complexity could be an indicator of high accuracy. In the case of summarization tasks, increased output complexity achieved by requests for summaries at an "expert" level can improve precision and recall in the face of concerns about named entity hallucination and detection https://aclanthology.org/2023.acl-srw.1.pdf. However, complexity has substantial costs in potentially making review of LLM output more difficult, and on simple questions it may even lead to degraded accuracy https://browse.arxiv.org/pdf/2302.12822.pdf.

COmplicated prompts may hurt robustness to irrelevant context: https://proceedings.mlr.press/v202/shi23a/shi23a.pdf

Chaining prompts creates complexity but also enables fluency and transparency - qualitative statements https://arxiv.org/pdf/2110.01691.pdf. In a study of prompt chaining, participants qualitatively reported...

I report:

- Vocabulary - share of words on the Academic Vocabulary List (AVL) for natural language and non-code components of responses. (Gardner and Davies, 2014) Share of novel n-grams in the response (words not in the prompt), presence of contrasting words 'while', 'but', 'though', 'although', 'other', 'others', 'however'. https://aclanthology.org/2023.findings-acl.591.pdf
- Number of named entities https://browse.arxiv.org/pdf/2309.04269.pdf
- Number of reasoning steps - line-breaks, periods, "step i" strings, and semicolons serve as separators. https://browse.arxiv.org/pdf/2210.00720.pdf
- Sentence length and Flesch reading ease (implemented via the textstat Python package) for natural language and non-code components of responses. (Flesch, 2016; Aggarwal)
- Cyclomatic complexity for code responses (implemented via the radon Python package). (Lacchia)

- Ratio or difference of these scores in prompts vs. responses, responses vs. accepted/outside correct answer

- Human assessment of need for specialized knowledge/difficulty of implementation of the technique. This could be task specific (done for some novel real world example questions/prompting scenarios), or it could be done overall based on a pre-existing description of the technique. Perhaps a balance of both is best. The actual metric will be a numeric score and a qualitative description.

- Human assessment of output complexity (ease of evaluating results). This could be task specific (done for some novel real world example questions/prompting scenarios), or it could be done overall based on pre-existing examples of the technique. Perhaps a balance of both is best. The actual metric will be a numeric score and a qualitative description.

- Human assessment of amount of irrelevant text generated

Data

To evaluate performance, I attempt to use tasks that are both general-purpose and close to practical, real-world applications. In this spirit, I use GRE General Test questions (from a purchased prep book or practice exams recently published online to minimize contamination), as well as HumanEval coding problems. For multiple choice choice questions, I will perform a string search, or use another (very capable) LLM, or perform manual interpretation to determine the letter answer the model selected based on its response (experimentation will be necessary before picking a method to use here). HumanEval uses the pass@k metric to automatically assess the probability a code solution is correct given a set of unit tests. (Chen et al., 2021)

I perform the analysis on one cutting-edge model and one older model, closer to the time that these techniques were introduced. This will provide a picture of the changing costs and benefits of advanced prompting, a trend that may even be extrapolated into the future if current LLM scaling laws continue to hold. As the most widely used models and the ones behind much original work in the field, I select two models from the OpenAI series: GPT-4 and text-davinci-003 on OpenAI playground. Should text-davinci-003 (a legacy model) become unavailable during the course of the project, or should I encounter other difficulties, I will use the simplest/smallest model available, likely GPT-3.5 (something to note is that models older than GPT-3.5 have, in the past, scored 0% for accuracy on coding problems - I will need to test all of my evaluations quickly to get a sense of feasible model choices before scaling up). All of these models are available both via the OpenAI API and in web interfaces - where possible I will use web interfaces to limit resources required.

To the extent possible, I will also report accuracy scores on the domain dataset as they are in the original paper introducing each technique. It may also be possible to use any prompts, LLM responses, and correct responses provided along with original papers to calculate other simple metrics such as response lengths and complexity. However, some metrics (time taken, human assessments of complexity) will require my own evaluations.

Analyses

I provide summary statistics of the metrics for each prompting method by model by question/task type. In cases where human/textual assessment and comments have been provided, it might be interesting to use NLP methods to evaluate responses (ex: for sentiment).

Limitations

It was difficult to select prompt engineering methods to try for this paper, and there is potential for my choice of methods to be somewhat biased. I mostly picked methods based my perception of their popularity and ease of implementation. If anything, this may lead to an underestimation of costs.

Just as my evaluation comes at a time with significantly more capable LLMs relative to those available when much work began on prompting, I expect the underlying calculus concerning prompting to continue to change in the future. However, I again only expect relative costs of complex engineering to increase as models get better.

Another potential problem is the extent that prompting techniques have been absorbed into default LLM behavior, likely through reinforcement learning. GPT-4 in particular does seem to automatically implement chain-of-thought methods when presented with a sufficiently complex problem. In this environment, this paper become less of an evaluation of prompting techniques themselves, but more of an evaluation of their intentional and manual implementation.

Finally, though I have taken steps to limit it, data contamination remains a real concern. The questions/tasks I use are unlikely to have been used in pretraining, but they may have been introduced to LLMs through reinforcement learning and other evaluations. On the other hand, this seems unlikely to bias the results for any one particular prompting method relative to the others or versus the control/direct prompting - comparisons internal to this paper are still likely to be useful.

Acknowledgements

The template for this document was adapted by Jordan Boyd-Graber, Naoaki Okazaki, and Anna Rogers.

References

Semantic Scholar | AI-Powered Research Tool.

Oguz A. Acar. 2023. AI Prompt Engineering Isn't the Future. *Harvard Business Review*. Section: Technology and analytics.

Shivam Bansal Aggarwal, Chaitanya. textstat: Calculate statistical features from text.

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. 2020. Language Models are Few-Shot Learners. ArXiv:2005.14165 [cs].

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. ArXiv:2107.03374 [cs] version: 2.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-Verification Reduces Hallucination in Large Language Models. ArXiv:2309.11495 [cs].

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active Prompting with Chain-of-Thought for Large Language Models. ArXiv:2302.12246 [cs].

Ethan Mollick [@emollick]. 2023. I have a strong suspicion that "prompt engineering" is not going to be a big deal in the long-term & prompt engineer is not the job of the future AI gets easier. You can already see in Midjourney how basic prompts went from complex in v3 to easy in v4. Same with ChatGPT to Bing. https://t.co/BTtSN4oVF4.

Rudolf Flesch. 2016. How to Write Plain English.

Dee Gardner and Mark Davies. 2014. A New Academic Vocabulary List. *Applied Linguistics*, 35(3):305–327.

Konstantin Hebenstreit, Robert Praas, Louis P. Kiesewetter, and Matthias Samwald. 2023. An automatically discovered chain-of-thought prompt generalizes to novel models and datasets. ArXiv:2305.02897 [cs].

Dave Hulbert. 2023. Using Tree-of-Thought Prompting to boost ChatGPT's reasoning. Original-date: 2023-05-22T19:03:27Z.

Michele Lacchia. radon: Code Metrics in Python.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated Knowledge Prompting for Commonsense Reasoning. ArXiv:2110.08387 [cs].

Jyoti Mann. Microsoft limits Bing chat exchanges and conversation lengths after 'creepy' interactions with some users.

Kim Martineau. 2021. What is prompt tuning?

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? ArXiv:2202.12837 [cs].

Cameron Shackell. 2023. Prompt engineering: is being an AI 'whisperer' the job of the future or a short-lived fad?

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *CHI Conference on Human Factors in Computing Systems*, pages 1–22, New Orleans LA USA. ACM.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large Language Models are Human-Level Prompt Engineers.

A The Popularity of Some Prompting Methods

I used Semantic Scholar to compile the following statistics concerning the popularity of X NUMBER general-purpose prompt engineering methods in the collection at `https://www.promptingguide.ai/papers#approaches` on October X, 2023 (noa). Here are the top 50 papers in terms of citations per day and their associated methods.

(See Internet Archive for list of paper collection stablelink)

Another good resource for prompt engineering methods and evaluations is the paperswithcode website https://paperswithcode.com/task/prompt-engineering. I did not make use of this page, however, as it seemed to be missing many prominent approaches, contains text-to-vision methods, and focuses on GitHub implementations (which are often low integer numbers difficult to compare) and currently trending social media items.

B Prompts Used

Below I have listed question and prompt examples for each method.

B.1 GSM8K

The sample problem is the first one in the GSM8K test dataset. All prompt examples for non-zero-shot methods are drawn from the training dataset.

<Question> "Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?"

Zero-Shot Control Baseline/Direct Prompting: <Question>

Zero-Shot Chain-of-Thought: <Question> Let's think step by step.

APE Improved Zero-Shot Chain-of-Thought: <Question> Let's work this out in a step by step way to be sure we have the right answer.

Tree-of-Thought: https://github.com/princeton-nlp/tree-of-thought-llm/tree/master https://github.com/kyegomez/tree-of-thoughts

One reasoning step:

Problem: <Question> Current plan of reasoning: <Current Plan> Task: Generate k different possible one-sentence thoughts to serve as step X in solving the problem. Only work on step X. Put each thought on a new line. Do not number them. Response:

Problem: <Question> Current plan of reasoning: <Current Plan> Potential next thoughts: <Potential Next Thoughts> Task: State the thought among "Potential next thoughts" that is most likely to contribute to solving the problem. If one of the thoughts fully solves the problem correctly, instead state the solution to the problem and output the word STOP on a new line. State only a thought or the solution and the word STOP. Response:

NOTE: This is very hard for older models to follow! davinci-002 can't really do any form of ToT tried

Baked in calculations:

Problem: <Question> Current plan of reasoning: <Current Plan> Task: Generate k different possible one-step calculations to serve as step X in solving the problem. Only work on step X. Put each calculation on a new line. Do not number them. Response:

Still hard to follow...

Zero-Shot Tree-of-Thought: Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realises they're wrong at any point then they leave. The question is... <Question>

Self-Refine: https://selfrefine.info/ See python code

Least-to-most Prompting (1-shot): https://arxiv.org/pdf/2205.10625.pdf Q: Elsa has 5 apples. Anna has 2 more apples than Elsa. How many apples do they have together? A: Let's break down this problem: 1. How many apples does Anna have? 2. How many apples do Elsa and Anna have together? 1. Anna has 2 more apples than Elsa. So Anna has 2 + 5 = 7 apples. 2. Elsa and Anna have 5 + 7 = 12 apples together.

Q: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market? A: Let's break down this problem:

Append to this: "The answer is:" in a second pass.

Unclear how to do Least-to-most Prompting (best):

Q: Four years ago, Kody was only half as old as Mohamed. If Mohamed is currently twice 30 years old, how old is Kody? A: To answer the question "How old is Kody?", we need to know: "How old is Mohamed?", "How old was Mohamed four years ago?", "How old was Kody four years ago?".

Q: If Pam is currently twice as young as Rena is, and in 10 years Rena will be 5 years older than her, how old is Pam now? A: To answer the question "How old is Pam now?", we need to know: "How much older is Rena than Pam currently?".

Q: As a freelancer, Baylor is paid for every finished work of a client he does on a freelance marketplace. Currently, he has $4000 on his dashboard from previous work done. He is currently working for three clients, with the first client paying him half the amount of money he currently has on his dashboard once the job is done. The second client will pay him 2/5 times more money than the first client once Baylor finishes his work. The third client will pay him twice the amount of money the first and second clients pay him together once he finishes the job. How much money will Baylor have in his dashboard after all the clients pay him for his work? A: To answer the question "How much money will Baylor have in his dashboard after all the clients pay him for his work?", we need to know: "How much will Baylor's first client pay him for his work?", "How much more will Baylor's second client pay him for his work compared to the first client?", "How much will Baylor's second client pay him for his work?", "How much will the first and second clients pay him together once he finishes the job?", "How much will Baylor's third client pay him for his work?", "How much money will all the clients pay Baylor for his work?".

Q: Cappuccinos cost $2, iced teas cost $3, cafe lattes cost $1.5 and espressos cost $1 each. Sandy orders some drinks for herself and some friends. She orders three cappuccinos, two iced teas, two cafe lattes, and two espressos. How much change does she receive back for a twenty-dollar bill? A: To answer the question "How much change does she receive back for a twenty-dollar bill?", we need to know: "How much did the cappuccinos cost in total?", "How much did the iced teas cost in total?", "How much did the cafe lattes cost in total?", "How much did the espressos cost in total?", "How much did the drinks cost in total?".

Q: Betty & Paige are raising money for their kids' little league team by hosting a bake sale. Betty has baked 4 dozen chocolate chip cookies, 6 dozen oatmeal raisin cookies and 2 dozen regular brownies. Paige baked 6 dozen sugar cookies, 3 dozen blondies and 5 dozen cream cheese swirled brownies. If they sell the cookies for $1.00 apiece and the blondies/brownies at $2.00 apiece, how much money will they raise? A: To answer the question "How much money will they raise?", we need to know: "How many dozen cookies did they bake (not including blondies/brownies)?", "How many cookies did they bake (not including blondies/brownies)?", "How many dozen blondies/brownies did they bake (not including cookies)?", "How many blondies/brownies did they bake (not including cookies)?", "How much money will they raise from the cookies (not including blondies/brownies)?", "How much money will they raise from the blondies/brownies (not including cookies)?".

Q: On a moonless night, three fireflies danced in the evening breeze. They were joined by four less than a dozen more fireflies, before two of the fireflies flew away. How many fireflies remained? A: To answer the question "How many fireflies remained?", we need to know: "How many fireflies joined?".

Q: Sam, Sid, and Steve brought popsicle sticks for their group activity in their Art class. Sam has thrice as many as Sid, and Sid has twice as many as Steve. If Steve has 12 popsicle sticks, how many popsicle sticks can they use for their Art class activity? A: To answer the question "How many popsicle sticks can they use for their Art class activity?", we need to know: "How many popsicle sticks does Sid have?", "How many popsicle sticks does Sam have?".

10.4.5 LEAST-TO-MOST (BEST) II - PROBLEM SOLVING: 4 EXAMPLES

Four years ago, Kody was only half as old as Mohamed. If Mohamed is currently twice 30 years old, how old is Kody? Q: How old was Mohamed four years ago? A: We were told that Mohamed is currently twice 30 years old, so he is currently

30 * 2 = 60 years old. That means that four years ago he must have been 60 - 4 = 56 years old. The answer is 56. Q: How old is Kody? A: Four years ago, Kody was half as old as Mohamed, so Kody must have been 56 / 2 = 28 years old then. Since Kody was 28 years old four years ago, she must now be 28 + 4 = 32 years old. The answer is 32.

Carla bought 2 bags of mini peanut butter cups on clearance. Each bag was $6.00 but was 75% off. How much did she spend on 2 bags of candy? Q: How much did she spend on 2 bags of candy? A: Each bag was $6.00 but was 75% off. So each bag cost $6.00 * (1 - 0.75) = $6.00 * 0.25 = $1.50. Carla bought 2 bags. So she spent $1.50 * 2 = $3.00. The answer is 3.

If Pam is currently twice as young as Rena is, and in 10 years Rena will be 5 years older than her, how old is Pam now? Q: How much older is Rena than Pam currently? A: Since Rena will be 5 years older than Pam in 10 years, she must be 5 years older than Pam now as well. The answer is 5. Q: How old is Pam now? A: If Pam is currently twice as young as Rena, that means that Rena is currently twice as old as Pam is. So if P stands for Pam's age now and R stands for Rena's age now, then we know that R = 2 * P And since Rena is 5 years older than Pam now, we know that R = P + 5. By substitution, we have P + 5 = 2 * P, which means that P = 5. The answer is 5.

Cappuccinos cost $2, iced teas cost $3, cafe lattes cost $1.5 and espressos cost $1 each. Sandy orders some drinks for herself and some friends. She orders three cappuccinos, two iced teas, two cafe lattes, and two espressos. How much change does she receive back for a twenty-dollar bill? Q: How much did Sandy spend on drinks? A: Sandy ordered three cappuccinos, which cost $2 each, so she spent $2 * 3 = $6 on cappuccinos. She ordered two iced teas, which cost $3 each, so she spent $3 * 2 = $6 dollars on ice teas. She ordered two cafe lattes, which cost $1.5 each, so she spent $1.5 * 2 = $3 on cafe lattes. She ordered two espressos, which cost $1 each, so she spent $1 * 2 = $2 on espressos. So altogether, Sandy spent $6 + $6 + $3 + $2 = $17 on drinks. The answer is 17.

Manual Few-Shot:

Randomly draw examples from the training set, as per https://arxiv.org/pdf/2005.14165.pdf

Manual Chain-of-Thought:

Page 35 of https://arxiv.org/pdf/2201.11903.pdf

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6. Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot? A: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5. Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total? A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39. The answer is 39. Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny? A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8. The answer is 8. Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now? A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9. The answer is 9. Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room? A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29. Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday? A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls. The answer is 33. Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left? A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8. Q: <Question> A:

B.2 Last Letter

Sources: kojima for data but also original chain of thought paper

There is no training set, only a small provided set of examples.

<Question>: Take the last letters of each words in Ẅhitney Erika Tj Benitoänd concatenate them.

(sic)

Zero-Shot Control Baseline/Direct Prompting: <Question>

Zero-Shot Chain-of-Thought: <Question> Let's think step by step.

APE Improved Zero-Shot Chain-of-Thought: <Question> Let's work this out in a step by step way to be sure we have the right answer.

Tree-of-Thought: https://github.com/princeton-nlp/tree-of-thought-llm/tree/master
https://github.com/kyegomez/tree-of-thoughts

One reasoning step:

Problem: <Question> Current plan of reasoning: <Current Plan> Task: Generate k different possible one-sentence thoughts to serve as step X in solving the problem. Only work on step X. Put each thought on a new line. Do not number them. Response:

Problem: <Question> Current plan of reasoning: <Current Plan> Potential next thoughts: <Potential Next Thoughts> Task: State the thought among "Potential next thoughts" that is most likely to contribute to solving the problem. If one of the thoughts fully solves the problem correctly, instead state the solution to the problem and output the word STOP on a new line. State only a thought or the solution and the word STOP. Response:

NOTE: This is very hard for older models to follow! davinci-002 can't really do any form of ToT tried. davinci-003 struggles.

Zero-Shot Tree-of-Thought: Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realises they're wrong at any point then they leave. The question is...

Self-Refine: https://selfrefine.info/ See python code

Least-to-most Prompting (4-shot): https://arxiv.org/pdf/2205.10625.pdf

Q: "think, machine" A: The last letter of "think" is "k". The last letter of "machine" is "e". Concatenating "k", "e" leads to "ke". So, "think, machine" outputs "ke". Q: "think, machine, learning" A: "think, machine" outputs "ke". The last letter of "learning" is "g". Concatenating "ke", "g" leads to "keg". So, "think, machine, learning" outputs "keg". Q: "transformer, language" A: The last letter of "transformer" is "r". The last letter of "language" is "e". Concatenating: "r", "e" leads to "re". So, "transformer, language" outputs "re". Q: "transformer, language, vision" A: "transformer, language" outputs "re". The last letter of "vision" is "n". Concatenating: "re", "n" leads to "ren". So, "transformer, language, vision" outputs "ren". Q: <Question> A:

Manual Few-Shot:

There is no training set, only a small provided set of examples.

Q: Take the last letters of the words in "Elon Musk" and concatenate them. A: The answer is nk. Q: Take the last letters of the words in "Larry Page" and concatenate them. A: The answer is ye. Q: Take the last letters of the words in "Sergey Brin" and concatenate them. A: The answer is yn. Q: Take the last letters of the words in "Bill Gates" and concatenate them. A: The answer is ls. Q: <Question> A:

Manual Chain-of-Thought:

Page 36 of https://arxiv.org/pdf/2201.11903.pdf

Q: Take the last letters of the words in "Elon Musk" and concatenate them. A: The last letter of "Elon" is "n". The last letter of "Musk" is "k". Concatenating them is "nk". The answer is nk. Q: Take the last letters of the words in "Larry Page" and concatenate them. A: The last letter of "Larry" is "y". The last letter of "Page" is "e". Concatenating them is "ye". The answer is ye. Q: Take the last letters of the words in "Sergey Brin" and concatenate them. A: The last letter of "Sergey" is "y". The last letter of "Brin" is "n". Concatenating them is "yn". The answer is yn. Q: Take the last letters of the words in "Bill Gates" and concatenate them. A: The last letter of "Bill" is "l". The last letter of "Gates" is "s". Concatenating them is "ls". The answer is ls. Q: <Question> A: