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, I evaluate the trade-offs between costs and benefits of prompting using novel metrics. Different prompting methods are assessed using standardized tasks and both state-of-the-art and older models.

Introduction

Prompt engineering, the practice of developing specialized prompts and queries to improve the accuracy of Large Language Models, 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 in 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 without error (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 involved prompting, such as the human time and effort induced by increased complexity and limitations on creativity and randomness (Wu et al., 2022), and others suggest the automation of prompting to avoid some of these costs (Diao et al., 2023). It is known that token costs, degradation of quality with increased prompt context lengths, and the uncertain nature of accuracy gains are all important practical considerations (Gao, 2023). However, the extent of these issues for various techniques, so as to enable standardized comparison between them and with a control baseline (no special prompting), has not been (to my knowledge) quantified.

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 length and complexity through the introduction of extra reasoning steps for both input prompts and output responses improves performance when using chain-of-thought prompting techniques (Fu et al., 2023). 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. Uniform improvements from complex chain-of-thought prompting are not fully accepted in the literature, however - other work has noted a tendency for the method to lead to worse performance on simple questions (Shum et al., 2023).

Complexity has also been studied within the context of prompting for summarization and story generation tasks. The Chain of Density prompting technique seeks to optimize the named entity density of LLM-generated summaries through choice from a series of repeated, increasingly dense iterations (Adams et al., 2023). 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. At the same time, other work has shown is difficult to control language model output complexity, meaning that the choice of specific techniques is important. Research demonstrates current models are not yet able to achieve compliance with desired readabil-

ity 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 (Pu and Demberg, 2023; Imperial and Madabushi, 2023).

This paper uses several metrics to evaluate the benefits and drawbacks 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 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. Next, I offer a new look at these prompting methods in a period long after their introduction. The current environment is one in which far greater capabilities are native to underlying foundation models. Finally, I introduce and adapt some useful measures of accuracy, quality, length, and complexity, such as inter-paragraph cosine similarity and the ratio of interaction length with prompting to the length of an accepted human-generated answer, to the challenge of LLM evaluation.

Data

To evaluate performance, I attempt to use tasks that are general-purpose, close to real-world applications, and standardized in the literature. To this end, I selected the GSM8K dataset, a collection of elementary-level math word problems (Cobbe et al., 2021), and a creative writing task involving the generation of a coherent two-paragraph passage with random, predetermined ending sentences for each paragraph. The original creative writing task in Yao et al., 2023 uses four paragraphs and sentences, but this is too difficult for older models and for the manual production of good answer demonstrations - I simply take the first two sentences for each original question.

These tasks carry several key benefits. They cover both the mathematical and linguistic domains - two types of tasks that form the foundation of the standardized testing of humans. GSM8K is studied in the majority of the papers introducing techniques used in this paper and is among the most common datasets used in the far larger list of papers I initially surveyed. Text and story generation is a widespread foundational task in NLP. Importantly, these sets of tasks are known to be free of data contamination. The GSM8K test set has been intentionally withheld from the training of OpenAI's models (OpenAI, 2023). The creative writing task was released only in 2023, and the code and data provided with the associated paper includes only questions and not LLM responses. [1]

I perform the analysis on one cutting-edge model and one older model from closer to the time that these techniques were introduced. This provides 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 (June 13th, 2023 version: 'gpt-4-0613') and text-davinci-003. All conversations were conducted programatically via the OpenAI API.

Prompting Methods Assessed

The following methods were selected based on their popularity (see Appendix Section A) and ease of implementation. They are listed in order of initial paper release/discovery date below (according to Semantic Scholar - citation publication dates below may be for revisions). A full list of all prompts for each task can be found in Appendix Section E.

- Few-Shot Prompting: The prompter provides a few examples of successfully answered questions or tasks before the main question/task. Notably featured in Brown et al., 2020.

- Few-Shot (Manual) Chain-of-Thought Prompting: The model is provided worked examples of answers in which the reasoning steps are written out. (Wei et al.) Note, however, that such steps are not explicitly

[1] It is also relatively simple to find random sentence generators online (such as https://www.thewordfinder.com/random-sentence-generator/), which would work for this task - at the cost of comparability with the results of (Yao et al., 2023). In any case, the 2-sentence task represents a modification of the original, and the test overall makes use of subjective human evaluations of the coherence of generated stories, limiting comparisons anyway - so perhaps testing the task on more truly novel pairings is a promising direction for future work.

planned out or mentioned, as is the case in least-to-most prompting.

- Least-to-Most Prompting: The model is given few-shot examples that demonstrate how to first break down the task into smaller and simpler subproblems, then solve them sequentially. (Zhou et al., 2023)
- Zero-Shot Chain of Thought Prompting (Original): Initial advances in chain of thought prompting to improve reasoning were achieved by simply including the following before the question/task: "Let's think step by step." (Kojima et al., 2023) For the creative writing task, the prompt following the question/task is adapted to: "Plan step-by-step before writing the passage." [2]
- Zero-Shot Chain of Thought Prompting (Automatic Prompt Engineer): Automated testing has indicated that an optimal zero-shot Chain of Thought prompt is "Let's work this out in a step by step way to be sure we have the right answer." (Zhou et al., 2022) For the creative writing task, the prompt following the question/task is adapted to: "Plan step-by-step before writing the passage to be sure we have a correct and coherent answer." [3]
- Self-Refine Prompting: The model produces an initial response, then is prompted for feedback which it uses for refinement. (Madaan et al., 2023)
- Tree of Thought Prompting: The language model traverses a tree of decisions - choosing among multiple steps or ideas it has generated to arrive at a conclusion. Backtracking is possible. (Yao et al., 2023)
- Zero-Shot Control Baseline/Direct Prompting: This method consists of just providing the question/task directly.

Analyses

I provide accuracy scores as well as summary statistics (mean, standard deviation) of the metrics discussed below for each prompting method by model by question/task type. Statistical inference is uncommon in the prompting literature - most papers simply report accuracy. [4] However, in this work, I do check for significant differences in metrics between prompting methods, reporting at the 95% level. For GSM8K accuracy scores I perform McNemar's tests comparing each method to the direct prompting baseline. [5] For sample means of creative writing scores and other statistics, I perform paired t-tests. [6]

Quality and Accuracy

Results concerning the accuracy and/or quality of each method can be found in Table 1.

For GSM8K problems, I report correct/incorrect accuracy at the point a technique has been fully implemented (the end of the tree of thought or after all Self-Refinement, etc.). Chain-of-thought methods and the related least-to-most method provide substantial gains for this task when implemented both manually and with a zero-shot approach. [7] Few-shot prompting - implemented in this paper as the provision of questions and answers without reasoning - is ineffective and even harmful. The iterative Self-Refine and Tree-of-Thought methods are generally-ill suited to this task, with the exception of Self-Refine checking when implemented with GPT-4, which can bring results up to the level of chain-of-thought methods.

My initial analysis of creating writing coherence consisted of the author's assessment of generated passage coherence (on a scale of 1 to 10, 1 being incoherent and 10 being very coherent) based on several guidelines - unconnected ideas or abrupt changes in characters or settings received low scores, and opposite cases received high scores.

Grader coherence scores are inherently noisy, as noted in Yao et al., 2023, but I take several measures to ameliorate this problem. First, I limit

[2] In initial experiments, this modification was found necessary to elicit any sort of step-by-step behavior from the model.

[3] Again, in initial experiments, this modification was found necessary to elicit any sort of step-by-step behavior from the model.

[4] Effect sizes and sample sizes are usually sufficiently large to merit confidence in the statistical significance of results. All paired t-tests comparing prompting methods and human responses were significant in the one paper I did find with inference, Pu and Demberg, 2023.

[5] The same questions are administered for each prompting method, so there is dependence that this paired test accounts for.

[6] Some other metrics remain, such as the token cost of accuracy or change in accuracy divided by change in tokens for each method versus direct prompting. Bootstrapping confidence intervals for this sort of value seems possible, but the dependent nature of the data poses challenges.

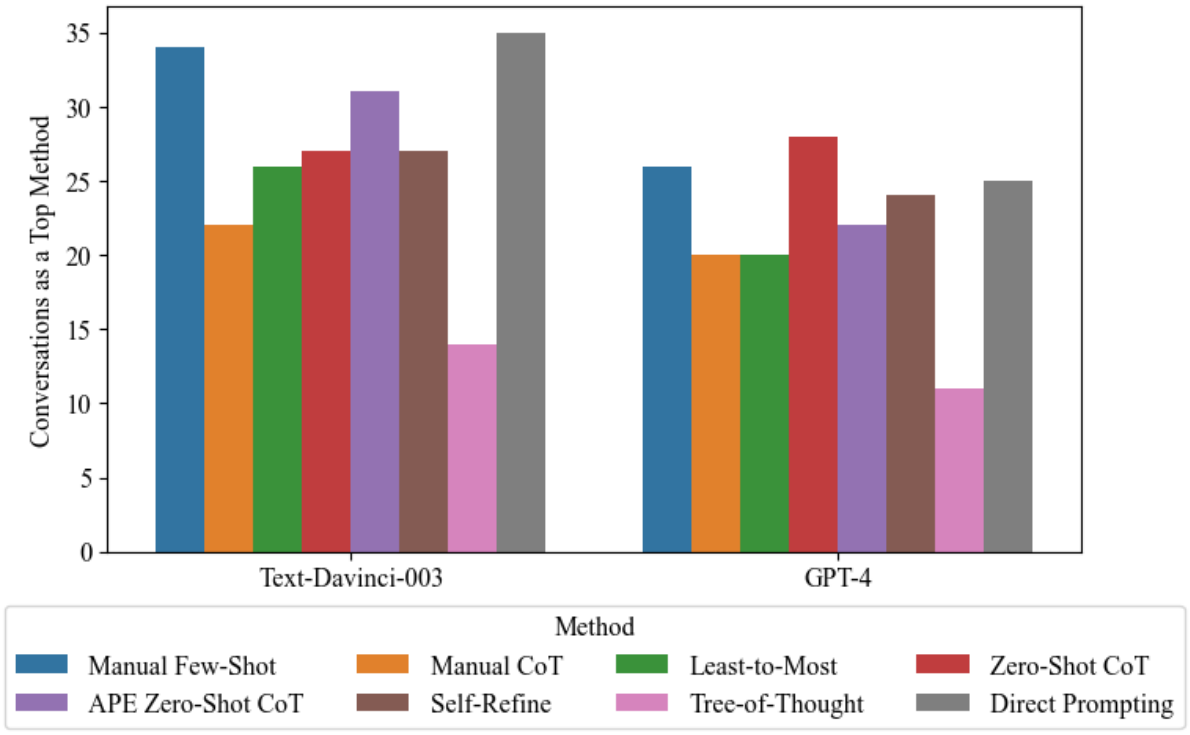
[7] Surprisingly, APE-Improved Zero-Shot CoT is not more effective that the simple "Let's Think Step by Step approach, as was the case in Zhou et al., 2022.

Table 1: Mean and Standard Deviation of Accuracy/Quality Scores

Task	Metric	Model	Manual Few-Shot (May 2020)	Manual CoT (Jan 2022)	Least-to-Most (May 2022)	Zero-Shot CoT (May 2022)	APE Zero-Shot CoT (Nov 2022)	Self-Refine (Mar 2023)	Tree-of-Thought (May 2023)	Direct Prompting
GSM8K	Accuracy	Text-Davinci-003	0.18	0.6*	0.58*	0.62*	0.49*	0.2	0.23	0.23
		GPT-4	0.49*	0.93*	0.95*	0.95*	0.93*	0.89*	0.4*	0.73
Creative Writing	Average Inter-Sentence Cosine Similarity	Text-Davinci-003	0.364 (0.063)	0.345 (0.061)	0.357 (0.059)	0.366 (0.077)	0.382* (0.077)	0.369 (0.064)	0.357 (0.096)	0.363 (0.064)
		GPT-4	0.346 (0.062)	0.35* (0.066)	0.341 (0.061)	0.347* (0.057)	0.35* (0.066)	0.356* (0.06)	0.351* (0.061)	0.333 (0.049)
	Average Inter-Paragraph Cosine Similarity	Text-Davinci-003	0.476* (0.163)	0.479* (0.161)	0.48* (0.162)	0.41* (0.192)	0.42* (0.192)	0.366 (0.179)	0.43 (0.222)	0.359 (0.178)
		GPT-4	0.386* (0.146)	0.422 (0.147)	0.422 (0.151)	0.465* (0.151)	0.464* (0.148)	0.412 (0.161)	0.447 (0.145)	0.42 (0.158)
	Average Inter-Sentence Cosine Similarity (Compliance Adjusted)	Text-Davinci-003	0.363 (0.064)	0.351 (0.063)	0.369 (0.059)	0.386 (0.076)	0.391* (0.072)	0.393 (0.054)	0.397 (0.104)	0.368 (0.072)
		GPT-4	0.349 (0.058)	0.355* (0.069)	0.35* (0.06)	0.344 (0.059)	0.356 (0.065)	0.351 (0.059)	0.348 (0.061)	0.334 (0.046)
	Average Inter-Paragraph Cosine Similarity (Compliance Adjusted)	Text-Davinci-003	0.433* (0.177)	0.401 (0.191)	0.414 (0.168)	0.357 (0.182)	0.389 (0.183)	0.34 (0.204)	0.224 (0.143)	0.371 (0.178)
		GPT-4	0.366* (0.154)	0.423 (0.15)	0.404 (0.159)	0.463* (0.155)	0.449 (0.159)	0.398 (0.158)	0.455 (0.143)	0.42 (0.157)
	Compliance	Text-Davinci-003	0.43	0.19*	0.25*	0.43	0.44	0.32*	0.04*	0.5
		GPT-4	0.63	0.51	0.52	0.57	0.56	0.48	0.26*	0.56

Stars indicate a significant difference from the direct prompting baseline from McNemar's tests at the 95% level for GSM8K accuracy scores and creative writing task compliance scores and paired t-tests at the 95% level for all other metrics.

Figure 1: Human-Preferred Method by Model - Creative Writing Coherence - Adjust for Non-Compliance



reported results for these grades to information concerning which method was preferred for each model and task question comprising a set of ending sentences. This limits demands on the data to only the ordinal ranking of methods. Second, I check the consistency of my methodology and results by soliciting a secondary opinion. I fine-tuned GPT-3.5 to learn my methodology for grading passage coherence - a difficult task, given the slippery nature of coherence as a concept and the wide variety of LLM responses and associated formats resulting from different models and methods. [8] I created two fine-tuned models, each trained and validated on a randomly selected 50% fold of responses (stratified by method) and human coherence scores, and deployed each model on the other 50% fold. The results demonstrate the consistency and learnability of human preferences. For 76.5% of model-sentence pairings, at least human and automated scores agree on at least one of the methods as among the top-preferred. Simulations using empirical probabilities of scores under independence indicate that this would occur in only 21.84% of cases by chance. [9]

In this and other analyses, I provide adjustments for scores accounting for whether or not the task constraints were followed - whether responses did, indeed, contain two paragraphs with specified ending sentences. [10] For human-preferred scores, adjustments for non-compliance give the outcomes in the case in which non-compliant responses are reduced to the lowest possible score.

Figure 1 shows the number of conversations where each method was one of the most preferred. It appears to be difficult to attain any sort of gains from prompt engineering on this task, at least as far as human preferences for coherence are concerned. The provision of few-shot examples indicating preferences seems to be the most promising approach, while other, more complex methods may be detrimental.

For my main results on creative writing, however, I introduce more objective measures of passage coherence that give a good sense of the concept on various levels - a novel approach for this creative writing task. To assess local coherence between sentences, I compute the average cosine similarity between consecutive sentence-level BERT embeddings (all-distilroberta-v1) (Landauer et al., 1998; noa).

For passage with n sentences indexed by i, each with embedding s_i, average inter-sentence cosine similarity is given by

$$CS_{IS} = \frac{1}{n-1}\sum_{i=1}^{n-1} cos(s_i, s_{i+1})$$

For a more global measure, I compute embeddings for each paragraph of the LLM response by averaging the sentence embeddings, and then the average cosine similarity between these paragraph embeddings for consecutive paragraphs. [11] I believe this custom method best captures success on this task, as outlined in the initial prompt and question, and it produced logical results (see Appendix Section B for examples to aid in interpretation) that also had the highest correlation with human scores of any metric. It is my preferred metric for the rest of this paper.

For passage with p paragraphs indexed by j each with number of sentences l_j, comprised of sentences indexed by i each with embedding $s_{j,i}$, average inter-paragraph cosine similarity is given by

$$CS_{IP} = \frac{1}{p-1}\sum_{j=1}^{p-1} cos(\frac{1}{l_j}\sum_{i=1}^{l_j} s_{j,i}, \frac{1}{l_{j+1}}\sum_{i=1}^{l_{j+1}} s_{j+1,i})$$

For both cosine similarity measures, I adjust for task compliance by removing non-compliant conversations.

These measures are again reported in Table 1. It is difficult to attain much improvement in inter-sentence cosine similarity, particularly when accounting for task compliance, but chain-of-thought methods and the APE Zero-Shot CoT method (adapted to included an extra request for correctness and coherence for this task) perform well.

[8] GPT-4 has been prompted to produce scalar scores for this task (Yao et al., 2023), but I generally found its grading to be inconsisent, even with few-shot prompts providing a few examples included.

[9] Contact the author for simulation details.

[10] I additionally tried to use GPT-4 to assess adherence to the original instructions - to check if the exact sentences specified in the prompt were used. In my initial experiments - in contrast to Yao et al., 2023 - I found that GPT-4 was not able to do this, repeatedly missing deviations of one or a few words, even when told to carefully perform the check in a step-by-step manner! I instead implemented simple logic using regular expressions, paired with a small amount of manual cleaning, and recommend this approach for future work.

[11] Other structural methods for assessing global coherence were considered, but not implemented due to their limited additional value and overall feasibility for short, 2-paragraph passages.

Self-Refinement also appears to be somewhat helpful. Gains from prompting are larger for inter-paragraph cosine similarity, and chain-of-thought and least-to-most methods tend to fairly consistently outperform, with some improvement coming from simple few-shot prompting on older models. Task compliance rates are low for both new and old models, suggesting careful adherence to complex instructions is an area with room for significant improvement in future models and methods. No prompt engineering method demonstates consistent improvement in task compliance, and prompting often leads to actively worse performance - potentially distracting models or adding additional complication.

Do larger/more modern models benefit more from prompt engineering, or are the techniques becoming obsolete? Gains from prompting on GSM8K questions have fallen as default performance on the task has improved dramatically. The creative writing task, however, is very much not a solved problem, and yet the gains from prompting still mostly appear to be larger for older models - though task compliance also seems to suffer more under almost all techniques. Earlier evidence demonstrated that gains from few-shot learning increased with model scale - but this paper does not seem to support evidence that trend has held. (Brown et al., 2020) Few-shot benefits have, if anything, decreased or flatlined with model improvement. For the moment at least, the overall relevance of prompting is a task-specific question, which depends on the metrics used to measure model performance.

One might expect more recent prompt engineering techniques (as measured by paper release/publication date) to be more powerful and useful, but the evidence shows this is not the case. Chain-of-thought prompting, popularized throughout mid-late 2022 (and including least-to-most prompting) seems to have been the most significant innnovation not only for math-based reasoning tasks as might be expected, but also for the language-based creative writing challenge. The culmination of chain-of-thought improvements in zero-shotting is able to attain comparable and sometimes superior results relative the provision of manual/few-shot chain-of-thought examples. Few-shotting does still have its uses, however, particularly for the creative writing task and when working with older models, where the construction of a coherent passage (a potentially subjective target) can be demonstrated. Complex iterative techniques such as Self-Refinement and traversal of the Tree-of-Thoughts, both introduced in 2023, largely do not lead to improvement (or attain less improvement than other methods), at least for these tasks. They may be appropriate only for some more specialized applications, and can only be successfully applied for the most recent models. [12]

We have seen that performance for a method does seem fairly repeatable across tasks, but what about the variability of performance within a task? Very high accuracy/compliance scores indicate consistent performance, and are present for only a few models and methods - perhaps only GPT-4 with one of the chain of thought methods can be relied upon, and only for the GSM8K task. Though inter-sentence results are consistent, variability in inter-paragraph cosine similarity is moderately large, and it is not entirely clear any method is sufficiently reliable.

Length

Table 2 contains results on the length and cost of conversations. I begin with an analysis of the length of the entire interaction in tokens (using OpenAI's tiktoken tokenizer for the appropriate model). Prompt engineering generally requires anywhere from 1-10 times as many tokens as direct prompting, which ends up having downstream effects for the amount of human effort and direct financial costs of tokens. Zero-Shot methods show moderate increases in length - factors of only 1-2 that of direct prompting. Self-Refine and Tree-of-Thought methods exhibit massive amounts of variability across questions and models that can lead to very long conversations - likely a result of differences in LLM decisions and backtracking. Few-shot methods generally have long lengths as a result of the space occupied by examples - oftentimes many times that of the actual question and results. This is clear from the results on input length, which are the vast majority of overall length for these methods - unlike the results for others.

For the GSM8K task, we can compare the length of conversations for each method with the length of the provided question + answer in Figure 2, providing another perspective as to the extent to which prompt engineering can "stretch out" interactions.

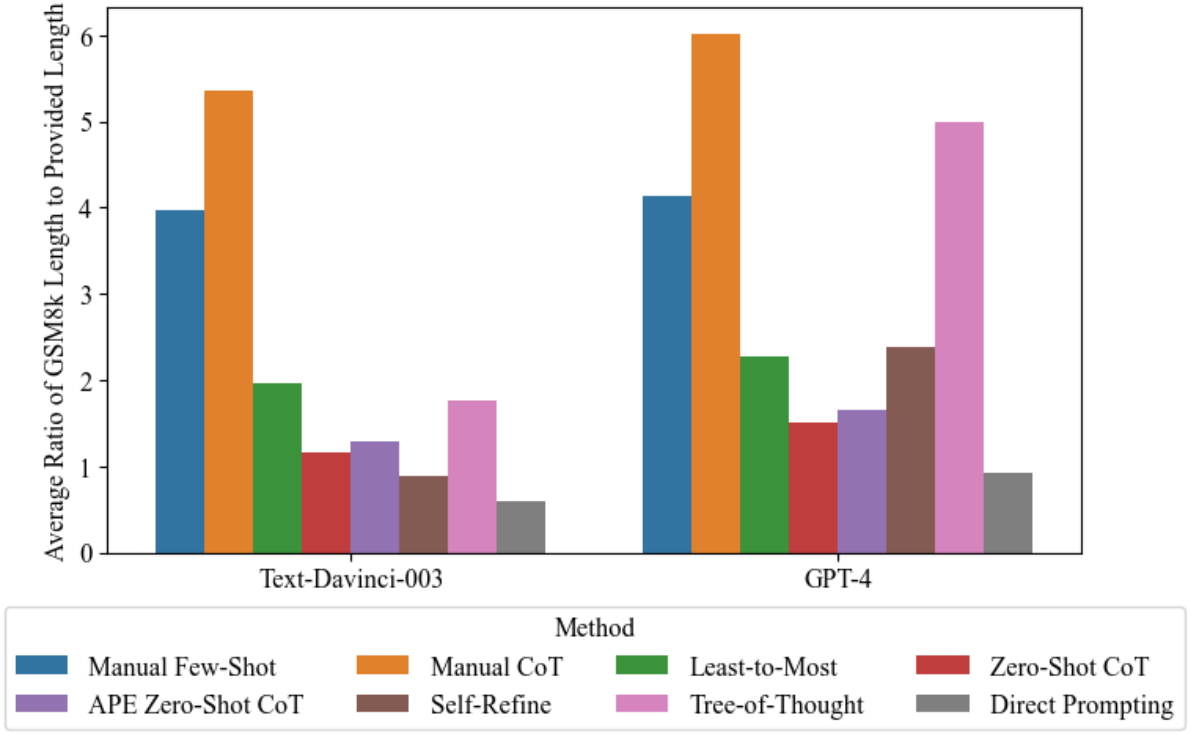
[12] Inspection of results found significant non-compliance with prompt instructions for these methods, especially with older models.

Table 2: Mean and Standard Deviation of Length Metrics

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting
GSM8K	Conversation Length	Text-Davinci-003	533.94* (21.162)	722.7* (35.469)	272.27* (49.824)	166.02* (47.204)	181.76* (54.995)	125.99* (36.175)	248.8* (116.245)	87.06 (42.023)
		GPT-4	579.69* (21.036)	850.04* (50.96)	332.36* (58.196)	223.88* (59.148)	243.93* (58.879)	344.86* (116.243)	764.72* (312.361)	146.69 (79.543)
	Input Length	Text-Davinci-003	531.48* (21.043)	655.48* (21.043)	158.48* (21.043)	67.48* (21.043)	80.48* (21.043)	99.48* (21.043)	137.68* (77.523)	59.48 (21.043)
		GPT-4	567.5* (21.012)	741.5* (21.012)	181.5* (21.012)	80.5* (21.012)	93.5* (21.012)	167.7* (38.215)	407.42* (154.353)	72.5 (21.012)
	Conversation Cost	Text-Davinci-003	0.011* (0.0)	0.014* (0.001)	0.005* (0.001)	0.003* (0.001)	0.004* (0.001)	0.003* (0.001)	0.005* (0.002)	0.002 (0.001)
		GPT-4	0.018* (0.001)	0.029* (0.003)	0.014* (0.003)	0.011* (0.003)	0.012* (0.003)	0.016* (0.006)	0.034* (0.014)	0.007 (0.004)
Creative Writing	Conversation Length	Text-Davinci-003	695.41* (31.517)	960.51* (36.447)	1025.02* (34.559)	256.53* (46.982)	274.04* (58.593)	382.19* (141.596)	916.81* (157.042)	200.41 (30.565)
		GPT-4	751.04* (35.272)	968.75* (55.611)	1091.41* (46.794)	459.98* (49.628)	470.72* (53.057)	520.06* (210.22)	1325.17* (158.559)	337.33 (43.738)
	Input Length	Text-Davinci-003	504.49* (7.697)	683.49* (7.697)	736.49* (7.697)	63.49* (7.697)	73.49* (7.697)	140.48* (38.088)	209.49* (7.697)	52.49 (7.697)
		GPT-4	522.21* (7.492)	701.89* (7.453)	754.89* (7.453)	75.21* (7.492)	84.89* (7.453)	160.37* (43.864)	264.89* (7.453)	65.89 (7.453)
	Conversation Cost	Text-Davinci-003	0.014* (0.001)	0.019* (0.001)	0.021* (0.001)	0.005* (0.001)	0.005* (0.001)	0.008* (0.003)	0.018* (0.003)	0.004 (0.001)
		GPT-4	0.029* (0.002)	0.037* (0.003)	0.043* (0.003)	0.025* (0.003)	0.026* (0.003)	0.026* (0.011)	0.072* (0.009)	0.018 (0.003)

Stars indicate a significant difference from the direct prompting baseline from paired t-tests at the 95% level.

Figure 2: Average Length vs. Provided Length - GSM8K



For text-davinci-003, direct prompting generally produces LLM conversations and responses somewhat shorter than provided answers, but zero-shot chain-of-thought prompting brings lengths up into the expected range. Other methods typically add large multiples. For GPT-4, direct responses are already near the expected length, and all forms of prompting add more tokens.

In Table 2, the cost of conversations was computed using rates as of November 11, 2023 - 2 cents per 1000 tokens for text-davinci-003, and 3 cents per 1000 tokens (input), 6 cents per 1000 tokens (output) for GPT-4. Prompting is the difference between spending a fraction of a penny on every conservation to several cents or more. Zero-shot methods do not increase costs much at all, and GPT-4's relative discount on input tokens makes manual/few-shot methods closer in expense to zero-shot ones, though there are still some differences. Similar to the results on length, the iterative Self-Refine and Tree-of-Thought methods have highly variable conversation cost.

Figure 3 considers the average change in accuracy or quality divided by the change in tokens, between the prompt engineering technique and the direct prompting baseline. Is any stretching of output adding value/improving accuracy/quality?

$$\frac{AQ_{PE} - AQ_B}{Tokens_{PE} - Tokens_B}$$

For GSM8K, the results are in percentage points per additional token. Zero-Shot Chain-of-Thought prompting (particularly with the original think step-by-step prompt) is cheap and extremely effective, and it outperforms techniques introduced earlier and later. The extra details and length of Least-to-Most and Manual Chain-of-Thought prompting do not provide the same benefit. Improvements in creative writing cosine similarity are harder to come by - note that changes are per 1,000 additional tokens, longer than most conversations ever last. However, we can still see that Zero-Shot Chain-of-Thought prompting is the most effective, with some benefit for the provision of examples using few-shot methods with the older text-davinci-003 model.

Figure 4 repeats the same calculation, but per additional cent of financial cost and with different y-axis scales.

$$\frac{AQ_{PE} - AQ_B}{Cost_{PE} - Cost_B}$$

The unique price structures for GPT-4 make this plot different, and the new y-axis scale provides additional insight - though one should be careful with extrapolation given the small cost (fraction of to a few cents) of most conversations (these plots should be interpreted in the context of earlier tables for any given practical decision). The intricacies of pricing structure reduce the relative gains on GPT-4 and level out comparisons between methods. On older models, prompting, and zero-shot-prompting in particular is very likely cost-effective, but newer models have changed this calculation.

Complexity

Other metrics beyond just length may be of interest - for example, in understanding how methods work, or when considering the important task of human review/checking of LLM responses. In Table 3, I report information on the complexity of responses. For both tasks, I examine the number of reasoning steps - linebreaks, sentences (NLTK sentence tokenizer), and strings "step i" (Fu et al., 2023) and "1. ", "2. ", "3. ", etc. in the response. [13] Due to the intricacies of results for various methods, none of these measures provides a full picture of complexity on its own, but their combination is more insightful. Here I report meaningful results based on knowledge of the underlying data.

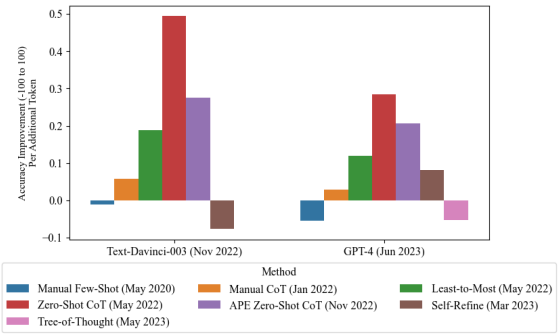
For the GSM8K task, which lacks standard formatting, drawing conclusions from the number of linebreaks is difficult, but by looking at the number of sentences we get a clearer picture. All prompting methods, except for manual few-shot prompting (which encourages the model to just directly state a number) add more sentences. The long structures of methods such as Tree-of-Thought prompting add the most. [14] For the creative writing task, data on the number of linebreaks shows that prompting does add a significant amount of complexity through additional steps, with Chain-of-Thought and Tree-of-Thought methods adding the most. Considering the number of sentences places some doubt on this conclusion, but for Creative

[13] Sentences seem to be a better measure of complexity than just periods as were used in prior work (decimals, abbreviations, etc. present challenges though the problems can be mitigated somewhat with regex). Semicolons were also considered, but in experiments these did not appear unless models were specifically prompted towards including them. "step i" comes from Fu et al., 2023, but "1. ", "2. " etc. are novel metrics, to the best of my knowledge.

[14] Care should be taken in interpreting the results on Least-to-Most prompting as detection of periods in explicitly formatted "1.", "2." etc. may have thrown off the tokenizer.

Figure 3: Gains Per Token v. Direct Prompting

(a) GSM8K



(b) Creative Writing

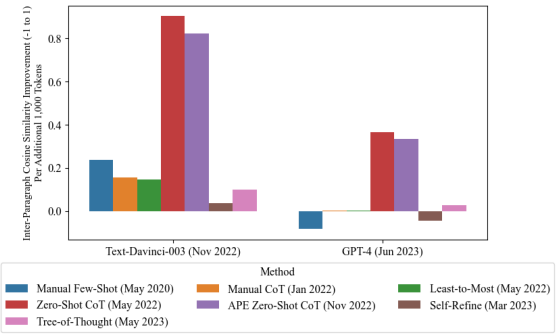
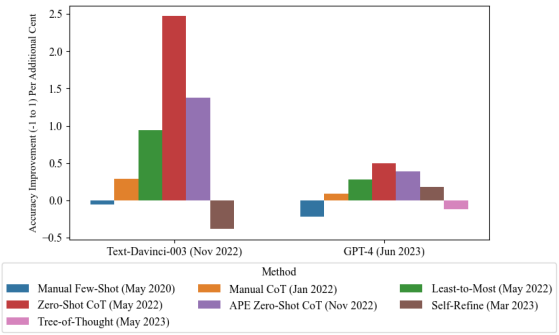
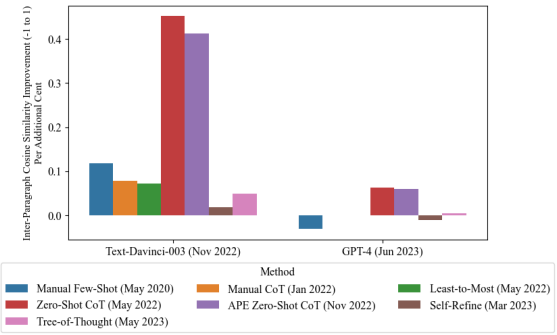


Figure 4: Gains Per Cent v. Direct Prompting

(a) GSM8K



(b) Creative Writing



Writing, sentences might mostly correspond to passage, rather than planning length - an introduction of additional noise that makes them less useful as a metric. Despite the usage of the appearance of language such as "step i" as a metric in prior literature, it appears sparsely in my data, even for zero-shot methods without formatting. My new metric of the number of "1. ", "2. ", etc. is more useful. The written planning of explicit steps elucidated by Least-to-Most and Chain-of-Thought prompting on GSM8K is clear, and there is also some planning for many methods in creative writing.

For the creative writing task, I also examine the sentence length (using NLTK word and sentence tokenizers) in the response and the Flesch Reading Ease (implemented via the textstat Python package) of the response (Flesch, 2016; Aggarwal). Most prompting methods produce slightly shorter sentences, except for the Tree-of-Thoughts, which produces longer ones - potentially through inclinations of the model to choose more complex paths and drafts of passages. Higher FRE scores indicate that material is easier to read. When few-shot examples are provided for the manual prompting methods and for Least-to-Most prompting, scores and readability increase somewhat relative to direct prompting (a modest effect, far less than a grade level), but for the other methods they remain similar or fall. Controlling readability may be difficult, but in line with the literature it is clear few-shot examples can provide a starting point. (Imperial and Madabushi, 2023)

Table 4 compares a selected set of the metrics above in responses to provided answers and prompts. GSM8K responses are compared to the provided answer. Manual few-shot prompting leads to short and simple responses by design. Other methods generally introduce more reasoning steps and sentences than the provided answers - Least-to-Most prompting, in particular, shows a dramatic increase. The iterative, choice-based setup of Tree-of-Thought prompting is again notable in the addition of steps. For the creative writing task, sentence length is generally somewhat longer in the responses relative to the prompts for few-shot methods (data on the other methods, which do not contain examples, is not as interpretable). For text-davinci-003, the few-shot responses are somewhat more readable than the prompts, but for GPT-4 they are less so - a concerning trend for the prospect of

Table 3: Mean and Standard Deviation of Complexity Metrics

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting
GSM8K	Number of Linebreaks	Text-Davinci-003	0.0* (0.0)	0.0* (0.0)	4.16* (1.85)	3.25* (1.74)	4.44* (2.37)	1.16* (0.66)	1.39* (0.93)	0.16 (0.72)
		GPT-4	0.0* (0.0)	1.17 (1.69)	5.64* (1.99)	3.83* (3.28)	4.64* (3.16)	5.37* (3.55)	12.26* (5.37)	1.34 (1.73)
	Number of Sentences	Text-Davinci-003	1.0* (0.0)	4.96* (1.22)	12.31* (3.06)	4.66* (2.31)	4.38* (3.01)	2.25* (1.05)	8.41* (3.38)	1.49 (0.97)
		GPT-4	1.0* (0.0)	3.5* (1.54)	8.59* (2.57)	2.84* (1.61)	3.15* (1.86)	5.12* (2.31)	8.03* (4.27)	1.51 (0.58)
	Number of Step 1, Step 2, etc.	Text-Davinci-003	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.32* (1.02)	0.56* (1.25)	0.0 (0.0)	2.0* (0.0)	0.0 (0.0)
		GPT-4	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.39* (1.14)	0.0 (0.0)	0.22 (1.32)	0.0 (0.0)
	Number of 1., 2., etc.	Text-Davinci-003	0.04* (0.2)	2.54* (2.63)	6.96* (3.06)	1.84* (2.62)	2.33* (3.24)	0.5 (0.99)	1.0 (0.0)	0.72 (1.77)
		GPT-4	0.04* (0.2)	2.8* (3.7)	6.95* (3.36)	1.99 (3.63)	1.95* (3.21)	2.36* (4.39)	3.5* (4.23)	1.28 (2.98)
Creative Writing	Number of Linebreaks	Text-Davinci-003	1.07* (0.26)	6.01* (0.5)	7.03* (0.3)	4.67* (2.47)	4.37* (2.97)	3.1* (1.54)	11.29* (2.75)	0.98 (0.25)
		GPT-4	2.01* (0.17)	3.69* (2.5)	7.5* (1.55)	10.93* (2.38)	10.77* (2.97)	4.37* (2.36)	18.54* (5.19)	2.08 (0.27)
	Number of Sentences	Text-Davinci-003	10.08* (1.86)	15.8* (1.72)	17.82* (1.88)	10.03* (2.38)	10.12* (2.48)	13.54* (5.37)	31.37* (6.9)	7.6 (1.33)
		GPT-4	10.25* (1.89)	11.93 (2.7)	14.76* (2.45)	15.92* (3.26)	15.43* (3.9)	15.97* (7.73)	39.42* (6.18)	11.32 (2.27)
	Number of Step 1, Step 2, etc.	Text-Davinci-003	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	1.71* (1.71)	1.62* (1.79)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
		GPT-4	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.66* (1.72)	0.9* (1.99)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
	Number of 1., 2., etc.	Text-Davinci-003	0.01 (0.1)	2.9* (0.46)	2.99* (0.1)	0.85* (1.18)	0.51* (0.89)	0.63* (0.79)	2.84* (0.92)	0.0 (0.0)
		GPT-4	0.0 (0.0)	0.94* (1.35)	2.49* (0.7)	3.58* (2.36)	2.81* (2.68)	0.8* (1.02)	3.93* (1.48)	0.0 (0.0)
	Sentence Length	Text-Davinci-003	16.24 (2.31)	14.4* (1.54)	13.14* (1.36)	15.38* (3.24)	16.08 (3.6)	14.63* (2.08)	18.94* (2.97)	16.41 (2.46)
		GPT-4	17.85* (2.45)	17.8* (2.71)	17.61* (2.42)	18.64 (3.69)	19.76 (4.85)	17.38* (2.0)	20.78* (2.67)	19.11 (2.6)
	Flesch Reading Ease Score	Text-Davinci-003	76.66* (7.58)	74.1 (6.22)	75.95 (5.7)	72.39* (8.54)	71.93* (8.89)	73.69 (7.86)	66.68* (8.17)	74.73 (8.24)
		GPT-4	67.84* (7.43)	67.76* (7.14)	67.37* (5.97)	59.95* (7.08)	60.74* (6.69)	62.37 (7.16)	57.57* (6.62)	63.78 (7.16)

Stars indicate a significant difference from the direct prompting baseline from paired t-tests at the 95% level.

Table 4: Differences of Complexity Metrics

Task	Metric	Model	Manual Few-Shot	Manual CoT	Least-to-Most	Zero-Shot CoT	APE Zero-Shot CoT	Self-Refine	Tree-of-Thought	Direct Prompting
GSM8K	Difference in Number of Sentences (Responses - Provided Answer)	Text-Davinci-003	-1.68	2.28	9.63	1.98	1.7	-0.43	5.73	-1.19
		GPT-4	-1.68	0.82	5.91	0.16	0.47	2.44	5.35	-1.17
	Difference in Number of Step 1, Step 2, etc. (Responses - Provided Answer)	Text-Davinci-003	0.0	0.0	0.0	0.32	0.56	0.0	2.0	0.0
		GPT-4	0.0	0.0	0.0	0.0	0.39	0.0	0.22	0.0
	Difference in Number of 1., 2., etc. (Responses - Provided Answer)	Text-Davinci-003	-1.24	1.26	5.68	0.56	1.05	-0.78	-0.28	-0.56
		GPT-4	-1.24	1.52	5.67	0.71	0.67	1.08	2.22	0.0
Creative Writing	Difference in Sentence Length (Responses - Prompts)	Text-Davinci-003	4.54	2.82	1.64					
		GPT-4	3.37	2.62	2.14					
	Difference in Flesch Reading Ease Score (Responses - Prompts)	Text-Davinci-003	1.66	-0.14	2.07					
		GPT-4	-7.29	-5.56	-6.87					

controlling readability in future.

The final components of my analysis of complexity are human-provided ratings of the ease of review and difficulty of implementation for each method. For ease of review, each method was rated in Appendix Section G. These results concur with the finding that Least-to-Most and Tree-of-Thought prompting add steps and complexity, while few-shot methods generally decrease it. Though Chain-of-Thought reasoning may add steps, these evaluations note the fact that they are often not too difficult to follow. On the other hand, steps from Least-to-Most and Tree-of-Thought prompting are often difficult to piece back together. An additional note is that the provision of examples in few-shot prompting can, aside from readability, help with the formatting of responses. Appendix Section C offers some observations on the difficulty of implementing each method. Zero-Shot methods are the easiest to implement. Few-shot methods just require a few examples - though providing chains of reasoning can be tricky. Iterative Self-Refine and Tree-of-Thought methods are complicated and can require specialized skills and time investment.

Length or Complexity?

As is the case in the study of chain-of-thought setting of Fu et al., 2023, are any gains in performance coming from reasoning steps as opposed to length in tokens? A generalized answer to this question across methods is central to our understanding of if and how prompt engineering works. Relationships may be complex. For language understanding tasks, prompt length has been found to improve model performance, with optimality achieved somewhere in the range of 20-100 tokens (Lester et al., 2021). Evidence shows that in longer conversations models may get distracted, go off on tangents, or get stuck in a loop repeating themselves (to the extent some platforms have imposed length limitations) (Shi et al., 2023; Mann). Accounting for non-linearity and relationships with reasoning steps to understand these phenomena may be helpful.

Table 5 implements regressions (with quadratic terms to model non-linearity) controlling for length and complexity to begin to address this question. To limit collinearity with my preferred length variable of thousands of tokens, I selected the number of steps/ideas as my desired complexity metric. I summed the "step i" and "1.", "2. ", etc. measures into this single measure to improve data quality.

For GSM8K, increases in linear conversation length do tend to improve performance, but the far larger squared term indicates that beyond a certain point further increases are detrimental. The coefficients are farily small, being per thousand tokens, and less significant than those for the number of reasoning steps/ideas. Reasoning steps are

more clearly linearly beneficial for performance, and their quadratic term is small.

Does increased length or complexity limit creativity and randomness? Table 5 also includes regression results on the creative writing task, which may be helpful - coherence requires creativity. Length and complexity (more steps/ideas and lower FRE scores) actually increase coherence, and the quadratic terms are weak. This comes at a cost of decreased task compliance, which follows an opposite relationship.

Conclusion

This broad standardized and quantitative evaluation of the trade-offs behind prompt engineering has revealed a fair amount of worthwhile benefit for a selected set of techniques. Chain-of-thought prompting outperforms other techniques and direct prompting on both math and language-based tasks, and it can be implemented quite cheaply and easily in a zero-shot manner.

Why does chain-of-thought prompting do so well? One plausible explanation is that LLMs have many examples loosely following the method in their training data. It is easy to find examples of reasoning problems worked out step-by-step in humanity's collective corpus of text, but ostensibly more difficult to think of examples where a tree of possible decisions is considered or where there is text laying out a conversation of responses, feedback, and refinement.

In a similar vein, LLMs may struggle with the creative writing task - even with prompt engineering - and with compliance with task instructions (to an alarming degree) as it is a unique task difficult to find examples for. Though the skills behind it and the task of maximizing passage coherence are generalizable and well-studied, the highly random nature of sentences used and the need to follow instructions exactly makes for a challenge. Gains on this task from prompting are statistically significant, but small and hard-fought.

Aside from being a somewhat task-specific question, the efficacy of prompting has indeed shifted over time, with benefits falling along with improvements in base models. Prompting is also a length and cost multiplier, especially for few-shot and interactive/complex methods. Putting these facts together, and further accounting for the differing cost structures for newer models, the gains in terms of accuracy and quality per additional token and cent spent on prompting have further fallen.

In addition to increasing the length of responses, prompting does introduce additional complexity - reasoning steps and structure, in addition to worsening readability (though sentence length may fall somewhat). Steps and complexity are not always a serious problem for human review, especially, if it done in a structured, ordered manner (Chain-of-Thought methods) - though this issue and complexity of implementation are real issues for the Tree-of-Thought (and to a lesser extent Self-Refine and Least-to-Most) method. Few-shot prompting is indeed a promising way to slightly reduce readability problems, in addition to improving formatting and the alignment of responses to human preferences. Though gains relative to direct prompting are somewhat larger, responsiveness to the readability of few-shot example passages has fallen for newer models.

Finally, this paper tested generalized models concerning the relationship between length and complexity and accuracy/quality - offering further insight into how methods work. On GSM8K, reasoning steps are indeed generally more important than length - which is also affected by a strong, negative quadratic term. For creative writing coherence, length and complexity improve cosine similarity but decrease task compliance. A chain of concise, organized, and well-chosen steps (or in a few cases, examples) can still bring out moderate gains from prompting, though some challenges and drawbacks remain.

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 net costs if easy-to-implement and high quality methods are likely to be popular.

Though I believe that the evaluation tasks I selected represent a wide range of areas in which LLMs may be useful - logical reasoning/problem solving, and creativity - they are not comprehensive, and different adaptations of prompting methods to different tasks may lead to different results.

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

Table 5: Regression Results

Model	Conversation Length (Thousands of Tokens)	Conversation Length (Thousands of Tokens) Squared	Number of Steps/Ideas	Number of Steps/Ideas Squared	Flesch Reading Ease	Flesch Reading Ease Squared
GSM8K Correct, Logit	0.277 (0.143)	-0.498* (0.127)	0.029* (0.007)	-0.002* (0.0)		
GSM8K Correct, Linear	0.282* (0.114)	-0.418* (0.091)	0.043* (0.006)	-0.002* (0.0)		
Creative Writing Cosine Similarity	0.212* (0.045)	-0.119* (0.03)	0.01* (0.004)	-0.0 (0.001)	-0.011* (0.004)	0.0* (0.0)
Creative Writing Compliance, Logit	-0.557* (0.161)	0.23* (0.11)	-0.025 (0.018)	0.005 (0.003)	0.041* (0.017)	-0.0* (0.0)
Creative Writing Compliance, Linear	-0.531* (0.158)	0.226* (0.107)	-0.033* (0.016)	0.006* (0.003)	0.045* (0.016)	-0.0* (0.0)

For logit regressions, the average of marginal effects are reported. Standard errors in parentheses. For linear models, standard errors are clustered by task question by method. Stars indicate significance at the 95% level.

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.

Due to the inherently unstable nature of LLM performance even in a fixed environment, and their broad and critical implications for society, there is great value even in testing models with exactly the same questions and prompts repeatedly. The statistical robustness of results from this paper and others should be tested repeatedly in future research.

Acknowledgements

Feedback from David Bamman, Kent Chang, Morris Chang, Kenan Carames, Mayank Bhushan, Ankita Shanbhag, and Wenjing Lin was much appreciated. Mingyu Yuan's advice encouraging the implementation of an objective measure of passage coherence was particularly helpful.

This document was created from a template made by Jordan Boyd-Graber, Naoaki Okazaki, and Anna Rogers.

References

sentence-transformers/all-distilroberta-v1 · Hugging Face.

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

Griffin Adams, Alexander Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting. ArXiv:2309.04269 [cs].

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].

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. ArXiv:2110.14168 [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.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-Based Prompting for Multi-Step Reasoning. ArXiv:2210.00720 [cs].

Andrew Gao. 2023. Prompt Engineering for Large Language Models.

Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. Flesch or Fumble? Evaluating Readability Standard Alignment of Instruction-Tuned Language Models. ArXiv:2309.05454 [cs].

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. ArXiv:2205.11916 [cs].

Thomas K Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. ArXiv:2104.08691 [cs].

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. ArXiv:2303.17651 [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].

OpenAI. 2023. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

Dongqi Pu and Vera Demberg. 2023. ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1–18, Toronto, Canada. Association for Computational Linguistics.

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

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large Language Models Can Be Easily Distracted by Irrelevant Context. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31210–31227. PMLR. ISSN: 2640-3498.

KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. ArXiv:2302.12822 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. ArXiv:2305.10601 [cs].

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. ArXiv:2205.10625 [cs].

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

Table 6 displays the popularity (in terms of Semantic Scholar citations) of some of the most popular generalizable prompt engineering methods based on the lists at https://www.promptingguide.ai/papers#approaches and https://en.wikipedia.org/wiki/Prompt_engineering#Text-to-text as of October 22, 2023. Please contact the author for a full list of the 162 papers considered.

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.

Table 6: Popularity of Selected Prompt Engineering Methods

Paper Title	Prompt Engineering Method	Citations Per Day Since Release
Language Models are Few-Shot Learners	Few-Shot Learning	13.23
Chain of Thought Prompting Elicits Reasoning in Large Language Models	Chain-of-Thought Prompting	3.33
Large Language Models are Zero-Shot Reasoners	Zero-Shot Chain-of-Thought	1.71
Tree of Thoughts: Deliberate Problem Solving with Large Language Models	Tree-of-Thought	1.43
Self-Refine: Iterative Refinement with Self-Feedback	Self-Refine	0.97
ReAct: Synergizing Reasoning and Acting in Language Models	ReAct	0.87
Least-to-most prompting enables complex reasoning in large language models	Least-to-Most Prompting	0.71
PAL: Program-aided Language Models	Program Aided Language Models	0.58
Large Language Models Are Human-Level Prompt Engineers	Automatic Prompt Engineer	0.55
How Can We Know What Language Models Know?	Prompt Mining, Prompt Paraphrasing	0.54
Automatic Chain of Thought Prompting in Large Language Models	Automatic Chain of Thought Prompting	0.53
Show Your Work: Scratchpads for Intermediate Computation with Language Models	Scratchpads	0.46
Multimodal Chain-of-Thought Reasoning in Language Models	Multimodal CoT	0.35
Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm	Metaprompt	0.32
CAMEL: Communicative Agents for "Mind" Exploration of Large Scale Language Model Society	Role-Playing	0.31
Chain-of-Verification Reduces Hallucination in Large Language Models	Chain-of-Verification	0.31
Complexity-Based Prompting for Multi-Step Reasoning	Complexity-Based Prompting	0.3
Decomposed Prompting: A Modular Approach for Solving Complex Tasks	Decomposed Prompting	0.27
Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models	Plan-and-Solve Prompting	0.27
Large Language Models Can Be Easily Distracted by Irrelevant Context	Instruction to Ignore Irrelevant Information	0.26
Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers	EvoPrompt	0.21
AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts	Chaining	0.2
Prompting GPT-3 To Be Reliable	Prompting for Reliability	0.19
Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP	Demonstrate-Search-Predict	0.19
ART: Automatic multi-step reasoning and tool-use for large language models	Automatic Reasoning and Tool-Use	0.18
Promptagator: Few-Shot Dense Retrieval From 8 Examples	Few-Shot Dense Retrieval	0.17
Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations	Maieutic Prompting	0.17
Reframing Instructional Prompts to GPTk's Language	Reframing	0.16
Generated Knowledge Prompting for Commonsense Reasoning	Generated Knowledge Prompting	0.15
Teaching Algorithmic Reasoning via In-context Learning	Algorithmic Prompting	0.15
Hard Prompts Made Easy: Gradient-Based Discrete Optimization for Prompt Tuning and Discovery	Gradient-Based Prompt Optimization	0.15

B Average Inter-Paragraph Cosine Similarity Examples

Inter-Paragraph Cosine Simlarity: 0.01

Learning to do a handstand is a fun activity for people of all ages. It takes practice to master, but once you get the hang of it, you'll learn to balance your body in an upside-down position. It isn't difficult to do a handstand if you just stand on your hands.

John, a space explorer, felt excited when he first stepped on the moon. It was a unique experience that he thought he would never forget. To his surprise, a smell of seared steak followed him up the mountain of dust. It caught him off guard that space smelled of seared steak.

Inter-Paragraph Cosine Simlarity: 0.25

The explorers were fascinated in equal parts fear and curiosity as they surveyed the unfamiliar terrain of this new planet. A unique alien species of flora, turning the water around it a shocking ochre, was the planet's highlight, causing a stark contrast to the otherwise jade green surroundings. Trembling hands reached out towards the phenomenon, a capsule took a sample, and rigorous testing was conducted. Initial concerns turned to relieved laughter when the tests ran positive. Just because the water is red doesn't mean you can't drink it.

The day ended, and the crew retreated to the layered confines of their habitat module. The walls were littered with mission updates and data charts, each element portraying the organized chaos of their voyage. In the space assigned for leisure, a small wooden table, reminiscent of Earth, was cluttered with mugs and a single book. Earlier discussions and laughter started to die out, replaced by the hum of computers and machines as each of them turned to their individual activities. Amidst this controlled mayhem, the book lay untouched, a visual metaphor of their longing for home and comfort. The book is in front of the table.

Inter-Paragraph Cosine Simlarity: 0.5

The sun beat down heavily on the sandy beach, and the smell of the salty ocean air mixed with the humidity. Suddenly, thick rain clouds appeared overhead and the looking sky filled with a dark shade of grey. The rain came in quickly, quickly, washing the thick sand into the ocean with a powerful force. Just as the storm intensified, she peered out over the horizon and saw something unexpected–a few crocodiles were launched into the water by the mass of the flooding rain. The sudden rainstorm washed crocodiles into the ocean.

She watched in appreciation as the creatures navigated the swells of the ocean, slipping out and in of view as the waves rose and fell. There was something meditative about it–the way the rain mixed with the sky–and she was mesmerized. Her gaze eventually fell on the man a few feet from her, watching the same scene unfold. She wondered what his eyes were saying beneath his mirrored sunglasses.

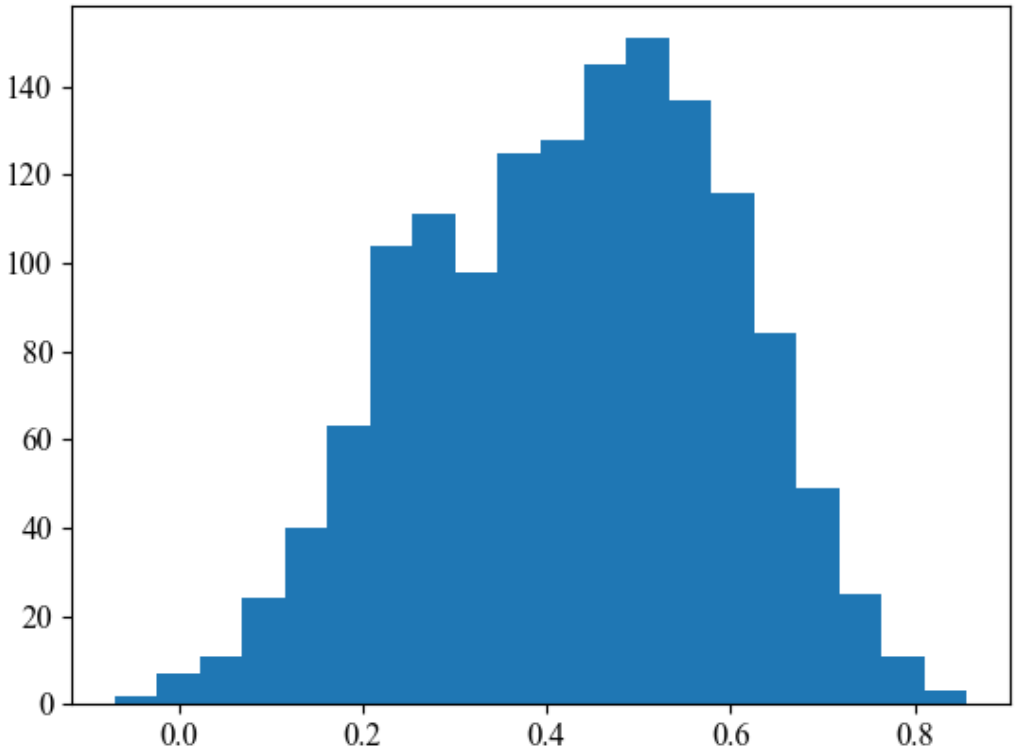
Inter-Paragraph Cosine Simlarity: 0.75

Elsa had been feeling lost since she graduated college. She hadn't found her passion in life yet and was looking for a way to feel purposeful and fulfilled. She decided to spend some of her savings on a road trip. She wanted to reclaim the feeling of being free that she had when she was a child, and the open road seemed like the way to do that. So, she grabbed her map and a duffle bag, and hit the highway. Her hair was windswept as she rode in the black convertible.

It took her to a small village in the mountains far from the city. She had days filled with new people, food, and experiences. She even took the time to learn some new skills and hobbies. She traveled because it

cost the same as therapy and was a lot more enjoyable. As the miles rolled by, she came to understand that by living intentionally and being courageous, one can find a second chance at life.

Figure 5: Histogram - Average Inter-Paragraph Cosine Similarity



C Notes on Ease of Implementation

Here are my rough evaluations of the difficulty of implementing each method selected for this paper. Ratings are on a scale from 1-6, with 1 being the easiest.

Zero-Shot Methods

Rating: 1. These methods append a common string to the prompt that is extremely similar for all tasks.

Manual Few-Shot

Rating: 2. A few examples of questions and solutions must be provided, but these are usually relatively easy to find.

Manual Chain-of-Thought

Rating: 3. Worked questions and solutions must be provided, but these are usually not very difficult to find - a good share of provided solutions also include explanations.

Few-Shot Least-to-Most

Rating: 4 A little more detail must be added to chain of thought solutions to demonstrate the structure behind splitting a problem into subproblems.

Self-Refine

Rating: 5. An iterative feedback and refinement loop, where the LLM examines and revises its previous responses, must be constructed. This is generally a straightforward process to build, however.

Tree-of-Thought

Rating: 6. A specific search tree structure and a method for traversing it must be constructed. This can vary a lot depending on the task - considerations are breadth-first versus depth-first, how many nodes/possibilities to consider at each step, etc. It is also necessary to consider how to structure a feedback loop where the LLM takes into account its previous responses. Any instructions must be simple enough for the LLM to follow.

D Notes on Ease of Review

Here are my rough evaluations of the difficulty of reviewing the output of each method selected for this paper. Ratings are on a scale from 1-6, with 1 being the easiest.

Manual Few-Shot, Manual Chain-of-Thought

Rating: 1. The prompt design and examples provided intentionally make it extremely easy to check the final answer - output is consistently formatted.

Direct Prompting

Rating: 2. The LLM responses are usually direct and the lack of extra instructions makes the output simple.

Zero-Shot Chain-of-Thought, APE Improved Zero-Shot Chain-of-Thought

Rating: 3. The steps the model takes for each individual problem and their formatting can be inconsistent. On the other hand, the steps are usually clearly spelled out, and the final answer or passage is usually fairly clear.

Self-Refine

Rating: 4. Self-Refine is an iterative process across several pieces of dialogue, though they do follow a simple structure.

Least-to-Most Prompting

Rating: 5. The planning and execution sets of steps of Least-to-Most Prompting can be hard to untangle and piece back together.

Tree-of-Thought

Rating: 6. Specific steps are split up across many pieces of dialogue and many potential tree branches, potentially with backtracking, and it is difficult to reconstruct the entire chain of logic and actions.

E Prompts Used

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

E.1 GSM8K

The sample problem below is the first one in the GSM8K test dataset. (Cobbe et al., 2021)

<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?"

All prompt examples for few-shot methods come from the training dataset.

Manual Few-Shot (randomly drawn questions and final answers from the training set - following the methodology on p.10 Brown et al., 2020. 8 examples were chosen, so as to enable comparison with the Chain-of-Thought exemplars below, and in light of evidence that this is around the optimal number of exemplars (Min et al., 2022). Explanations were removed from the answers.)

Q: For every 12 cans you recycle, you receive $0.50, and for every 5 kilograms of newspapers, you receive $1.50. If your family collected 144 cans and 20 kilograms of newspapers, how much money would you receive?

A: 12

Q: Betty picked 16 strawberries. Matthew picked 20 more strawberries than Betty and twice as many as Natalie. They used their strawberries to make jam. One jar of jam used 7 strawberries and they sold each jar at $4. How much money were they able to make from the strawberries they picked?

A: 40

Q: Jack has a stack of books that is 12 inches thick. He knows from experience that 80 pages is one inch thick. If he has 6 books, how many pages is each one on average?

A: 160

Q: James dumps his whole collection of 500 Legos on the floor and starts building a castle out of them. He uses half the pieces before finishing and is told to put the rest away. He puts all of the leftover pieces back in the box they came from, except for 5 missing pieces that he can't find. How many Legos are in the box at the end?

A: 245

Q: Ines had $20 in her purse. She bought 3 pounds of peaches, which are $2 per pound at the local farmers' market. How much did she have left?

A: 14

Q: Aaron pays his actuary membership fees each year. The membership fee increases yearly by $10. If he pays $80 in the first year, how much does his membership cost, in dollars, in the sixth year?

A: 130

Q: Joseph invested $1000 into a hedge fund. The fund promised a yearly interest rate of 10%. If he deposited an additional $100 every month into the account to add to his initial investment of $1000, how much money will he have in the fund after two years?

A: 3982

Q: The price of buying a wooden toy at the new Craftee And Best store is $20, and the cost of buying a hat is $10. If Kendra went to the shop with a $100 bill and bought two wooden toys and three hats, calculate the change she received.

A: 30

Q: <Question>

A:

Manual Chain-of-Thought (taken from p.35 of Wei et al.)

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:

Least-to-most Prompting (1-shot; taken from p.58 of Zhou et al., 2023)

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: <Question>

A: Let's break down this problem:

Zero-Shot Chain-of-Thought

Q: <Question>

A: Let's think step by step.

APE Improved Zero-Shot Chain-of-Thought

Q: <Question>

A: Let's work this out in a step by step way to be sure we have the right answer.

Self-Refine

Q: <Question>

A: <Response>

Task: Please check the answer above. If there is an error, state what the error is, but don't fix it. If there are no errors, output STOP.

Feedback: <Feedback>

Task: Redo the entire problem based on the most recent feedback.

A: <Response>

Tree-of-Thought

Q: <Question>

Task: Generate 3 different possible one-step calculations to serve as step 1 in solving the problem. Only work on step 1. Put each calculation on a new line. Do not number them.

<Response>

Task: State the calculation above that is most likely to contribute to solving the problem. If it fully solves the original problem, also output STOP and the solution to the problem. If none of the calculations are correct, output ERROR and generate three new ones.

<Response>

Task: Generate 3 different possible one-step calculations to serve as the next step in solving the problem. Only work on the next step. Put each calculation on a new line. Do not number them.

<Response>

Zero-Shot Control Baseline/Direct Prompting

Q: <Question>

A:

E.2 Creative Writing

The sample task below is the first one in the Creative Writing dataset. (Yao et al., 2023)

<Task> Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph must be: 1. It isn't difficult to do a handstand if you just stand on your hands. 2. It caught him off guard that space smelled of seared steak. [15]

No training data was provided, so I used random sentences courtesy of `https://www.thewordfinder.com/random-sentence-generator/` and my own answers in the construction of few-shot examples.

Manual Few-Shot

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. She couldn't get a job because she didn't have a permanent address. 2. He didn't have any hidden talents.

Response:

Laura sat on the park bench, watching the people walk by. She was homeless, and had been for a few months now. She couldn't get a job because she didn't have a permanent address.

She had tried to talk to career counselors about her situation, but the conversations often seemed fruitless. She didn't feel she had any marketable skills. Her situation was similar to that of her friend, Rodrigo, who openly shared a similar attitude with counselors in his meetings. He didn't have any hidden talents.

[15] Some preliminary experiments replacing the first sentence with "Write a coherent passage of 2 short paragraphs that flow together.", which seemed to improve zero-shot performance. This seems to nudge the model away from writing two unrelated paragraphs ending in the selected sentences - a flaw it often notices when asked for feedback on its own passages. However, I decided not to adopt this modification to stay in line with the original task.

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. He had always wanted to be a Youtuber but never thought it would actually happen. 2. My sweater got caught on the door hinge.

Response:

My brother, John, had been making home videos for years, but they never got much attention. He was always disappointed when he saw other people's videos getting thousands of views. Then one day, he got a call from a company that wanted to sponsor him. They offered him a lot of money to make videos for them. He was so excited that he couldn't sleep that night. He had always wanted to be a Youtuber but never thought it would actually happen.

As it turned out, John would need his own production staff to help with script writing and video editing. As I lived in the area and had prior experience in these fields, I was a natural choice for a part-time role on his channel. The company's sponsorship was very generous, and I would get a large portion of the profits. I was glad to finally be able to earn a substantial income in a more exciting and engaging role than my current position as a barista. I was smiling for most of our first business meeting, and strutted with pride out of our new studio. My sweater got caught on the door hinge.

Task: <Task>

Response:

Manual Chain-of-Thought

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. She couldn't get a job because she didn't have a permanent address. 2. He didn't have any hidden talents.

Response:

Ideas:

1. Make the passage about travelling circus performers looking for other work as their circus shuts down.

2. Make the passage about the struggles of a homeless person who is trying to get a job.

3. Make the passage about perceptions and preconceptions of people's skills and social status as factors in hiring.

Passage:

Laura sat on the park bench, watching the people walk by. She was homeless, and had been for a few months now. She couldn't get a job because she didn't have a permanent address.

She had tried to talk to career counselors about her situation, but the conversations often seemed fruitless. She didn't feel she had any marketable skills. Her situation was similar to that of her friend, Rodrigo, who openly shared a similar attitude with counselors in his meetings. He didn't have any hidden talents.

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. He had always wanted to be a Youtuber but never thought it would actually happen. 2. My sweater got caught on the door hinge.

Response:

Ideas:

1. Make the passage about a sister visiting her brother; the brother has recently become a successful Youtuber - she excitedly gets her sweater caught leaving a meeting with him.

2. Make the passage about a men's fashion reviewer who is working on a video review of a sweater.

3. Make the passage about a Youtuber preparing for a video shoot - as they hurry through things, their sweater gets caught but this becomes an amusing part of their vlog.

Passage:

My brother, John, had been making home videos for years, but they never got much attention. He was always disappointed when he saw other people's videos getting thousands of views. Then one day, he got a call from a company that wanted to sponsor him. They offered him a lot of money to make videos for them. He was so excited that he couldn't sleep that night. He had always wanted to be a Youtuber but never thought it would actually happen.

As it turned out, John would need his own production staff to help with script writing and video editing. As I lived in the area and had prior experience in these fields, I was a natural choice for a part-time role on his channel. The company's sponsorship was very generous, and I would get a large portion of the profits. I was glad to finally be able to earn a substantial income in a more exciting and engaging role than my current position as a barista. I was smiling for most of our first business meeting, and strutted with pride out of our new studio. My sweater got caught on the door hinge.

Task: <Task>

Response:

16

Least-to-most Prompting

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. She couldn't get a job because she didn't have a permanent address. 2. He didn't have any hidden talents.

Response: Let's break down this problem into steps: First, we will write a few ideas for the passage. Second, draft the passage.

Ideas:

1. Make the passage about travelling circus performers looking for other work as their circus shuts down.

2. Make the passage about the struggles of a homeless person who is trying to get a job.

3. Make the passage about perceptions and preconceptions of people's skills and social status as factors in hiring.

Passage:

Laura sat on the park bench, watching the people walk by. She was homeless, and had been for a few months now. She couldn't get a job because she didn't have a permanent address.

[16]Note Yao et al., 2023 prompts the model for a plan in what is considered the Chain-of-Thought method adaptation for the Creative Writing task.

She had tried to talk to career counselors about her situation, but the conversations often seemed fruitless. She didn't feel she had any marketable skills. Her situation was similar to that of her friend, Rodrigo, who openly shared a similar attitude with counselors in his meetings. He didn't have any hidden talents.

Task: Write a coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. He had always wanted to be a Youtuber but never thought it would actually happen. 2. My sweater got caught on the door hinge.

Response: Let's break down this problem into steps: First, we will write a few ideas for the passage. Second, draft the passage.

Ideas:

1. Make the passage about a sister visiting her brother; the brother has recently become a successful Youtuber - she excitedly gets her sweater caught leaving a meeting with him.

2. Make the passage about a men's fashion reviewer who is working on a video review of a sweater.

3. Make the passage about a Youtuber preparing for a video shoot - as they hurry through things, their sweater gets caught but this becomes an amusing part of their vlog.

Passage:

My brother, John, had been making home videos for years, but they never got much attention. He was always disappointed when he saw other people's videos getting thousands of views. Then one day, he got a call from a company that wanted to sponsor him. They offered him a lot of money to make videos for them. He was so excited that he couldn't sleep that night. He had always wanted to be a Youtuber but never thought it would actually happen.

As it turned out, John would need his own production staff to help with script writing and video editing. As I lived in the area and had prior experience in these fields, I was a natural choice for a part-time role on his channel. The company's sponsorship was very generous, and I would get a large portion of the profits. I was glad to finally be able to earn a substantial income in a more exciting and engaging role than my current position as a barista. I was smiling for most of our first business meeting, and strutted with pride out of our new studio. My sweater got caught on the door hinge.

Task: <Task>

Response:

Zero-Shot Chain-of-Thought

<Task>. Plan step-by-step before writing the passage.

APE Improved Zero-Shot Chain-of-Thought

<Task>. Plan step-by-step before writing the passage to be sure we have a correct and coherent answer.

Self-Refine

Task: <Task>

Response: <Response>

Your Task: Provide feedback on the correctness and coherence of the response and a rating on a scale of 1-10. If it is already coherent and correct to the extent you would award a 10, output 10 and the word STOP.

Feedback: <Feedback>

Your Task: Rewrite the passage based on the most recent feedback.

Response: <Response>

Tree-of-Thought

<Modified Task. Example: Goal: A coherent passage of 2 short paragraphs. The end sentence of each paragraph, respectively, must be: 1. It isn't difficult to do a handstand if you just stand on your hands. 2. It caught him off guard that space smelled of seared steak.>

Your Task: Generate 3 one-sentence plans for potential passages. Only generate one-sentence plans - do not write the passage.

<Response>

Your Task: Select the most coherent plan that follows the rules of the task. Only state the plan - do not write the passage.

<Response>

Your Task: Write 3 drafts of the 2-paragraph passage based on this plan.

<Response>

Your Task: Select the most coherent draft that follows the rules of the task and write it out.

<Response>

Your Task: If the draft is correct and coherent to the extent you would award a 10 on a scale of 1 to 10, output STOP. If it is not, write out a different one-sentence plan for a potential passage from among those considered and output PLAN.

<Response>

Zero-Shot Control Baseline/Direct Prompting

<Task>

F Evaluating Creative Writing Responses

Fine-tuned GPT-3.5 was prompted with the following to elucidate a scalar score of passage coherence:

<Passage>

Your Task: Rate the coherence of the passage above on a scale of 1 to 10, 1 being incoherent and 10 being very coherent. If the passage is two unconnected pieces give it a score of 1. If it has sentences that seem to be randomly inserted, has abrupt change in characters, or has abrupt change in setting, give it a low score. If it has a continuous setting or characters and seems plausible, give it a high score. If it is as coherent as you believe is possible, give it a score of 10.

Score: <Score>

G Evaluating Ease of Review

GPT-4 was prompted with the following to elucidate a scalar score of the ease of evaluating the reasoning behind a response:

<Conversation>

On a scale of 1-10 (1 being easy and 10 being difficult), how difficult is it to check the reasoning behind the above conversation? Write your justification then put the numeric rating on its own line as the last line of your response.

H Experimental: Automated GSM8K Grading

<Conversation>

Provided Answer:

<Answer>

Task:

Output 0 if the final answer does not match the provided answer and 1 if it matches the final answer.

[17]

[17] One way to get automatic GSM8K grades would be requiring the model to output a final answer on a new line in the course of prompt interactions. I did not implement this in order to keep conversations natural and because errors here would not be errors in reasoning per se.