

Are Language Models Better at Generating Answers or Validating Solutions?

Anonymous ACL submission

Abstract

Recently, large language models (LLMs) have demonstrated remarkable reasoning abilities, augmented by recent advances in prompting techniques and reasoning frameworks. Many popular frameworks (Du et al., 2023; Yao et al., 2023; Shinn et al., 2023) rely on the assumption that models are able to give effective feedback on their own generations. This feedback is partly predicated on being able to correctly validate, or classify, the generated prediction as either correctly or incorrectly solving the given problem. While in traditional computer science settings validation has been shown to be as difficult as correct generation, we find empirically that language models may be better discriminators than generators. Our work studies whether leading language models are better at solving problems or validating solutions, and we attempt to gain a better understanding of why this happens. We quantify this by measuring the understanding gap — the difference between generative and discriminative accuracy. First, we further corroborate recent work (West et al., 2024) showing surprisingly that models are better generators than discriminators on some datasets. Second, we discover that understanding gaps can be closed or significantly narrowed through prompting and provide an estimate of the upper bound ϵ on the understanding gap across datasets. Third, we apply our findings to predict the settings where self-correction is most effective. This continues the conversation started by (Huang et al., 2023), where we instead show that LLMs can self-correct reasoning, and establish a link between a feature of the dataset and the language model’s ability to self-correct.

ference between generative and discriminative accuracy. First, we further corroborate recent work (West et al., 2024) showing surprisingly that models are better generators than discriminators on some datasets. Second, we discover that understanding gaps can be closed or significantly narrowed through prompting and provide an estimate of the upper bound ϵ on the understanding gap across datasets. Third, we apply our findings to predict the settings where self-correction is most effective. This continues the conversation started by (Huang et al., 2023), where we instead show that LLMs can self-correct reasoning, and establish a link between a feature of the dataset and the language model’s ability to self-correct.

1 Introduction

Language models have recently begun to demonstrate human-like reasoning capabilities, driven in large part by zero shot prompting-based self-correction algorithms. This brings to bear the ques-

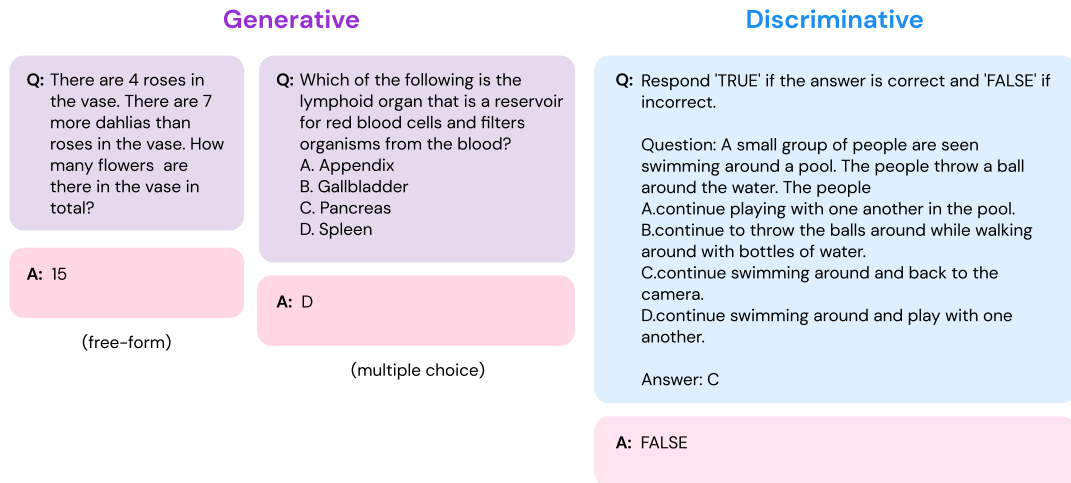


Figure 1: How do we assess "generative" and "discriminative" abilities of models? For the generative modality, we provide the model with free-form and multiple choice problems and assess based on solution accuracy. For discriminative, we provide the model with a question-answer pair and ask the model to assign a binary "TRUE" or "FALSE" label depending on whether it believes the predicted answer p to be correct given the question q .

tion, why does the research community expect — and indeed observe — that language models can improve upon their initial generation?

1.1 Can models learn new things from self-correction?

Some view self-correction as a form of post-hoc prompting (Huang et al., 2023), where generation quality improves through self-conversation quite simply because, within the confines of its context, the model is generating a better "prompt" for itself. This correlation between better prompting methods and higher generation accuracy has been demonstrated in the literature (Wei et al., 2022c,a), further supporting the strengths of this explanation.

In this view, self-correction would be viewed as several, iterative turns of a model generating prompts for itself. If this is indeed true, the recent success of self-correction methods would mean that our current prompting methods still fall short of optimal prompts. This would highlight the relevance of pursuing additional research into prompting and interpretability of language model self-corrections. While self-correction methods work well in-practice, we may not fully understand why the model chooses to issue the corrections it does, and how it chooses which to accept.

A question of efficiency also arises. Many self-correction algorithms are more computationally expensive than generating the correct answer directly from a human-specified prompt. Can we simply find the optimal prompt and forgo this expensive self-correction entirely? Ostensibly, by construction the use of this so-called optimal prompt would mean that no level of self-correction would improve upon the quality of the generation; whether measured as accuracy, determinism, or safety.

It is possible that there is a ceiling on the degree to which self-correction can improve accuracy. Otherwise, if the optimal prompt could achieve 100% accuracy across all datasets, it would seem prudent to direct significant focus towards prompt design. Until we have both better understood prompting, and we have designed or identified the models that can achieve this level of capability, however, we will likely have to rely on continued self-prompting. These ideas seem to suggest that models are not learning new things from self-correction but simply positioning themselves in a way to better retrieve existing knowledge.

1.2 Two types of discrimination

We identify two broader modalities of generative model output: generation and discrimination (see Figure 1). Generation involves the structure of task used by most datasets: the language model is provided with a free-form or multiple choice question and is asked to generate the correct solution. The model is then evaluated on the accuracy of its solutions. In the discrimination setting, the model is provided the question and a prediction, then is asked to answer TRUE or FALSE depending on whether the answer correctly follows from the question.

We hypothesize the existence of two types of bases of discrimination: logical reasoning and retrieval. The first, logical-reasoning-based discrimination, is largely non-reliant on existing knowledge and validates a solution purely based on whether the solution is valid. For humans, this is equivalent to validating a Sudoku solution by checking whether each row, column, and sub-grid is duplicate-free. The second, retrieval-based discrimination, is knowledge-dependent (hence the "retrieval" moniker) and involves solving the problem, then comparing the solution to the provided answer. For humans, this is equivalent to validating the Sudoku solution by solving the puzzle then comparing the two solutions (assuming only a single solution exists for the puzzle).

The astute reader will likely be contemplating some key characteristics that we would expect to see if this hypothesis were valid. We would expect models relying on retrieval-based discrimination to fail when presented with problems with multiple correct solutions (e.g. constraint satisfaction tasks like COLLIE (Yao et al., 2024), and some Sudoku and 24 Game puzzles). A model using retrieval-based discrimination would be expected to perform approximately equivalently on generation and discrimination, since discrimination is simply generation plus a comparison. However, a model using logical-reasoning-based discrimination would be expected typically to more significantly under- or over-perform on discrimination as opposed to generation, since its discrimination ability is decoupled from its generation ability. We believe that when researchers discuss discrimination as a proxy for understanding, they are referring specifically to reasoning-based discrimination as opposed to the more derivative retrieval-based one.

1.3 Discrimination basis is dataset dependent

We further hypothesize that discrimination is dataset dependent, and that models switch — for reasons currently beyond the scope of this paper — between these two modes when presented solely with a problem (without additional prompting such as Chain of Thought (Wei et al., 2022c), etc.). We expect that when a model attempts to use reasoning-based discrimination on a dataset by default and this form of discrimination significantly underperforms generation, we can improve discrimination accuracy up to the level of generation performance through prompting. Indeed, we find that through careful prompting (see Appendix B), we are even able to significantly improve performance even while producing only a single output token. It is possible that this prompt "switches" the model from reasoning-based to retrieval-based validation.

This paper seeks to better grasp how and what language models understand. Given that self-verification, reflection, and most multiagent debate algorithms rely on the belief that generative models are able to uncover new knowledge through repeated generations with themselves, we hope that our study on relative generator-discriminator performance will help to better understand why and when feedback is effective. We refer to this generator-discriminator gap the understanding gap and seek to find an ϵ that can upper bound the understanding gap across all datasets. We will determine a reasonable estimate for this bound by assessing results across 10 popular datasets covering a diverse range of skills spanning constraint satisfaction, arithmetic, multihop reasoning, common sense, and reading comprehension. While it is expected that manipulating prompts can improve performance, the question is whether we can completely close the understanding gap through prompting. We find, remarkably, that even when constraining the model to a single output token during discrimination, prompting can close even the large understanding gaps (some originally greater than 50%). We further attempt to apply our findings to predict the settings where self-correction is most effective, further engaging the hypotheses presented by (Huang et al., 2023) and (West et al., 2024). We specifically test the performance of multiagent debate (Du et al., 2023), which relies on self-correction, and use our learnings to model accurate the types of datasets that the algorithm performs most effectively on.

2 Related works

Prior work shows that language model performance can depend substantively on the structure of the prompt used. This begs the question of what it means for language models to understand, and whether we can conflate memory with understanding. It is possible that models are simply so good at memorizing and interpolating between these memorized answers that they appear to give semblances of understanding. Surely we wouldn't say that a grade school student has fully grasped algebra if they simply score well on problems similar to ones covered in class. That would just be memorization. We expect them instead to successfully answering never-before-seen questions probing understanding would we confidently say that this student understands the topic of algebra. Lastly, we look at influential prior work whose effectiveness may depend upon the assumption that language models are better validators than generators.

Impact of prompting on model performance

Prompting can remarkably allow models to adapt to new scenarios even with no task specific data (Wei et al., 2022a). Extending this further, the prompts themselves can be generated by models, which has shown to improve generative performance as compared to human-generated prompts (Gao et al., 2021; Guo et al., 2022; Ben-David et al., 2022). The use of previous generations as a prompt for future generations presents a recurring motif underlying much of the self-reflection and agentic debate space. However, current prompting methods are largely based on classification and generation, highlighting the need for more research on prompting for information extraction, text analysis or other interrogative understanding based tasks like discrimination (Liu et al., 2023).

Do language models understand deeply? Prior work has found that some language models have begun to show reasoning capabilities resembling a human-like general intelligence (Bubeck et al., 2023). While some studies have shown some of these emergent behaviors to be artifacts of dataset quirks (Wei et al., 2022b), other studies have more carefully investigating how human and model understanding may differ despite comparable generative capabilities (West et al., 2024). Other work has focused on whether language models are able to leverage this understanding to self-correct their generations to improve accuracy or morality (Huang

et al., 2023; Ganguli et al., 2023).

Methods relying on strong discriminator capabilities A number of effective methods reliant on strong model discrimination have emerged in the literature. The first of two types involves improving factuality of model generations by using self-generated verification questions or through multi-agent debate which rely on the ability of models to probe their own or each other’s understanding through generations with varying priors (Dhuliawala et al., 2023; Du et al., 2023). Inspired in part by the increase in model reasoning ability when using its own generations as scaffolding for future generation (Wei et al., 2022c), the second of these types involves improving reasoning through prompts that allow the model to self reflect (Shinn et al., 2023; Yao et al., 2023; Madaan et al., 2023).

3 Are language models better generators or discriminators?

We attempt to formally investigate a phenomena that has often been assumed to be true — that discrimination can help improve generation quality. This belief may arise from our expectations that the training data used by these models more closely resembles the generation task paradigm than the discriminator one. Notably, we specifically seek to gain a generalized understanding beyond single datasets, realizing that while models may achieve very strong discriminative performance relative to generative on some datasets it is much more helpful to show that we can get comparable discriminator-generator performance across most datasets through carefully crafted prompts.

Suppose we have a question q and a prospective answer a . For instance, consider a question q from the GSM8K dataset, a prediction p , and the reference answer a (see Table 1). Note that the prediction a may be different from the ground truth, as it is in this case. The goal of the discriminator is to identify whether the prediction p is correct, given the question q . The ground truth discriminator response is TRUE if $p = q$ and FALSE otherwise.

Intuitively, we expect models to have discriminative abilities that surpass its generative ones. This is because given q and p , we can always ask the model to predict the correct answer to q , then compare this with p . For a simple TRUE/FALSE discriminator, this would give us a discrimination accuracy identical to the generation accuracy. Surprisingly, for a generator with accuracy below 50%, this style

| | |
|----------|---|
| q | Jame will turn 27 in 5 years. In 8 years his cousin will be 5 years younger than twice his age. How many years separate the age of the two now? |
| p | 33 |
| a | 25 |

Table 1: Example of a question q and prospective answer a pair. Note that a is actually an incorrect answer in this case.

of prompting would give us an accuracy that is worse than random guessing between the two TRUE, FALSE options (50%).

However, if we want to limit models to roughly as many tokens when generating vs. discriminating, we would no longer be able to use this approach, since discrimination would necessarily take more tokens than generation. This might cause the model to generate more hastily, likely reducing generation quality, since it still needs token space to perform the comparison between its generated answer and the prediction p . This is why we might expect that, in the worst case, that validator performance might trail generator performance slightly.

3.1 Reguritation with interpolation

We take the understanding gap to be the difference between generation and understanding abilities of a language model. Bridging the understanding gap shows that language models understand their generations, rather than simply performing what we call regurgitating with interpolation. That is to say, we expect models to produce correct answers to questions it has never seen before, and while direct regurgitation of training data would produce lackluster performance, models may interpolate between regurgitations in adjacent areas of knowledge to produce reasonable or even correct generations. This phenomenon could be increasingly likely given the large scope and magnitude of training data. In these cases of regurgitation with interpolation, we would expect a large gap between generation and understanding abilities. This gap would mean poor performance when probing the language model on why it chose its generation or asking it to validate a prediction that it’s not told is its own. The first step to bridging the self-reflection chasm, the point where models can teach

themselves new information by simply reflecting on their responses, is to close the understanding gap.

Where the Generative AI paradox (West et al., 2024) provides a hypothesis based on relative discriminator performance between humans and models, we provide a hypothesis that investigates the discriminative ability of models relative to their own generation ability. The hypothesis guiding this paper is as follows:

Hypothesis 1: Models are, approximately, as good at validating solutions as they are at generating answers to questions. However, this performance requires the careful choice of an appropriate validator prompt. We state the hypothesis formally as,

$$\forall t \in T, \exists p_t \in P \text{ s.t. } g(t) - u(t, p_t) < \epsilon$$

where t is some task in the set of all possible tasks T , p_t is some task-specific prompt in the set of all possible prompts P , $g(t)$ is the generation accuracy for a task t , $u(t, p)$ is the understanding (proxied by validation accuracy) on a task t with prompt p , and ϵ is some small non-negative value; the upper bound on the generator-validator performance spread across all tasks. We consider $g(t) - u(t, p_t)$ to be a formal definition of the understanding gap.

To make a strong case supporting this hypothesis, we desire to show that across a diverse range of datasets there exists an upper bound on generator-validator accuracy spread. To increase the robustness of our study, we also investigate two subhypotheses whose validity we expect to be consistent with Hypothesis 1. If these two sub-hypotheses are found to be concurrently valid with those of Hypothesis 1, we believe this presents a strong case for the latter. These two subhypothesis investigate the ability of prompting to improve validation performance and whether validation accuracy captures underlying model understanding well.

3.2 Prompt dependency of validation

While it may seem trivial that prompting has an effect on generation quality, the extent and nature of the effect is less clear. Many popular prompting methods are designed for generation rather than discriminative settings, and it is unclear whether prompting improves generations simply because more computation is performed or whether it is unlocking an otherwise elusive understanding of the

problem. Further, while a better prompt may improve accuracy on a dataset, it is uncertain whether, in the eyes of a model, a generation with a better prompt will allow a model to become a teacher to itself, producing a single model student-teacher model configuration.

We take a positive stance on the ability of prompting to tighten the upper bound. Specifically, since $g(t)$ and $v(t, p)$ are both measures of accuracy between 0 and 1, the trivial upper bound ϵ is 1. We also realize that some tasks may naturally have negative understanding gaps because correct answers may be especially difficult to generate for both humans and machines, despite being fairly easy to validate. These tasks might, for instance, involve satisfaction of a simply stated mathematical or textual constraint that has a large state space of potential answers with non-intuitive mapping of the answer space to constraint satisfaction.

Sub-hypothesis 1: We can tighten the upper bound on model error through prompt selection.

3.3 Useful validation as a form of understanding

We next investigate whether correct validation constitutes understanding, and resultingly whether such understanding allows self-reflection to improve model performance. If it does, validation correctness presents a much more simple and objective metric to assess self-reflection quality, while also potentially being more accurate since assessments of intermediate reflection in practice would likely be human- rather than model-preference-centric. If correct validation constitutes understanding, we would expect the following sub-hypothesis to be true.

Sub-hypothesis 2: Self-correction-based algorithms work best on datasets with lower understanding gap.

4 Methodology

Benchmarks. We run experiments across datasets spanning various natural language skills. We investigate the ability of models to perform constraint satisfaction through **COLLIE** (Yao et al., 2024), arithmetic through **24 Game** (Yao et al., 2023) and **GSM8K** (Cobbe et al., 2021), and multihop reasoning via **HotpotQA** (Yang et al., 2018), **MultispanQA** (Li et al., 2022) and **MMLU** (Hendrycks et al., 2021). Equally important are the abilities to operate in settings of commonsense

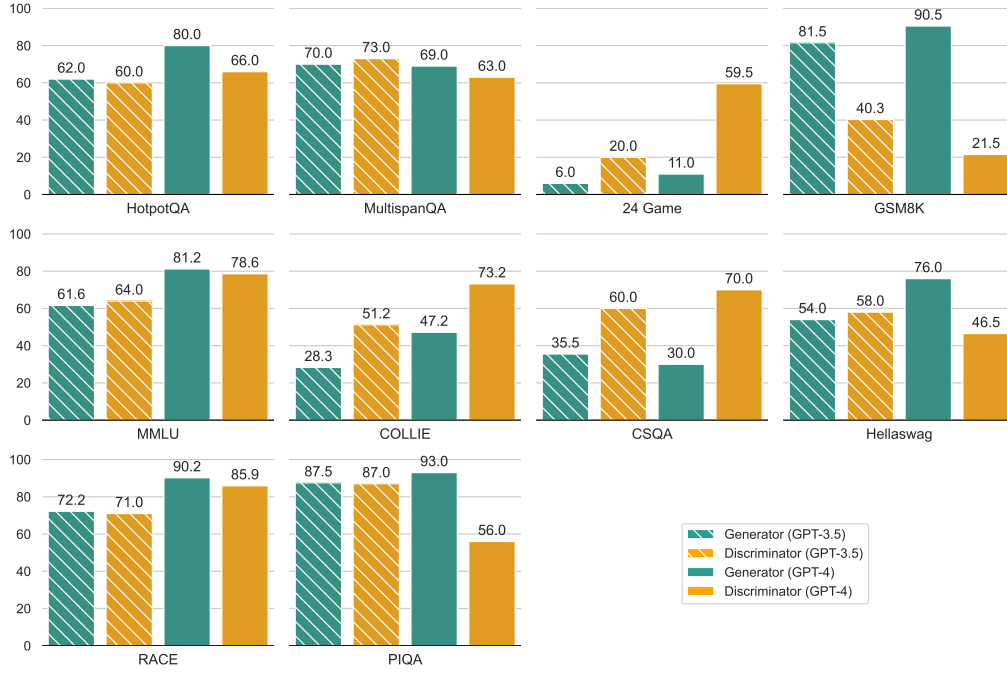


Figure 2: Generation and discrimination of GPT 3.5 across various skills, in both the natural dataset settings and the log probs setting

through **CSQA** (Talmor et al., 2019), **Hellaswag**, (Zellers et al., 2019) and **PIQA** (Bisk et al., 2020), and reading comprehension via **RACE** (Lai et al., 2017).

Models. Since we are testing the limits of current state-of-the-art language model generation and understanding capabilities, we choose to evaluate on the most popular (among both researchers and industry users) and capable language models. To that end we evaluate our hypothesis primarily on **GPT-3.5** and **GPT-4**, specifically gpt-3.5-turbo-1106 and gpt-4-1106-preview, respectively (John Schulman et al., 2022; OpenAI et al., 2023).

Evaluation. We consider each of the benchmarks both in their originally intended configurations and a setting we call log probs, in which we convert each task into a multiple choice problem and normalize the log probability of each option over all valid choice generations, taking inspiration from (Holtzman et al., 2021). For the regular generation setting, we assess based on the dataset’s accuracy measure. For regular discrimination, we present the model with the original question and its own generation and ask it to respond with TRUE if it believes the answers to be correct and FALSE otherwise.

For our baselines (see Figure 2), we use a simple

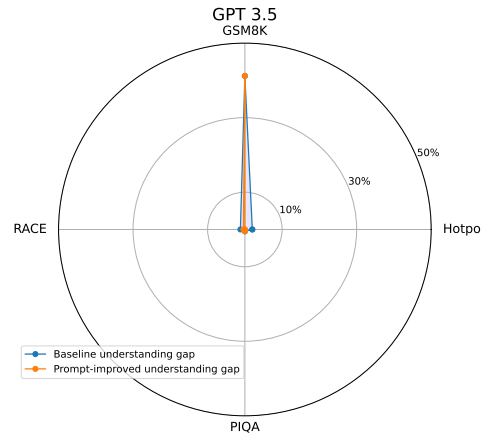


Figure 3: Lowest achieved understanding gap by adjusting validator prompt on GPT 3.5 across various skills, compared to baseline understanding gap

system prompt with a minimal amount of information necessary to instruct the model to complete each task. To evaluate the effect of prompt, we try various system prompts following at times architectures shown to be effective in past works. In evaluating the effect of understanding gap on reflection algorithms, we implement two popular techniques for self-correcting reasoning: Reflexion (Shinn et al., 2023) and Multiagent Debate (Du et al., 2023).

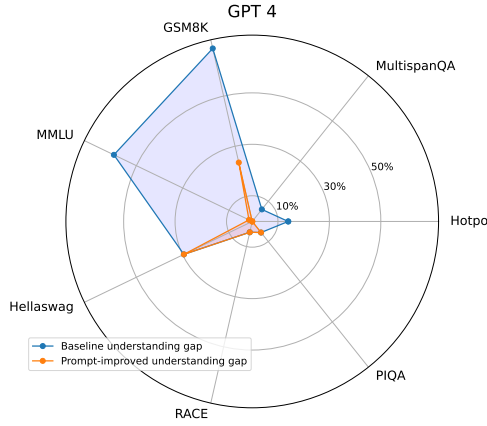


Figure 4: Lowest achieved understanding gap by adjusting validator prompt on GPT 4 across various skills, compared to baseline understanding gap

5 Discussion

5.1 Are language models better generators or discriminators?

Of the 10 tasks spanning skills in constraint satisfaction, arithmetic, multihop reasoning, common sense, and reading comprehension, 3 have negative understanding gaps where discrimination performance surpasses generation performance on both models (see Figure 2). Surprisingly, we find that while GPT-4 generally outperforms GPT-3.5 in generative capacities, it sometimes surprisingly underperforms on discrimination on those same tasks, perhaps suggesting semblances of overfitting in GPT-4 to more oft seen dataset paradigms. In fact, GPT-3.5 presents an understanding gap on only 4 of the 10 datasets, while GPT-4 presents a gap on 7 of the 10.

For the tasks with positive understanding gaps on GPT 3.5, however, the gap is often small. The exception is GSM8K where the model significantly underperforms in a discrimination setting, trailing even random guessing (50% for a TRUE/FALSE configuration) despite high generation accuracy, suggesting anticorrelation. This is promising, since a trivial (and problematic on principle) solution could be to prompt the model to answer the opposite of what it thinks. Ostensibly, this would then give us a $(100\% - 40.3\% = 59.7\%)$ accuracy, but wouldn't really be faithful to our underlying exploration of model understanding.

In the baseline setting, the model exhibits understanding gaps on multihop reasoning, arithmetic,

reading comprehension, and common sense skills. We next attempt to close these understanding gaps through prompting.

5.2 Can we make up for the understanding gap through prompting?

On GPT-3.5, we close the understanding gap on HotpotQA by using prompt P1 (see Appendix A). We find remarkably that despite the model still only outputting a single token (TRUE or FALSE), validation accuracy surpasses the generator accuracy; closing the gap and even resulting in a negative understanding gap. We are able to reduce both understanding gaps for RACE and PIQA below 0.5%, however, the model continues to discriminate poorly on GSM8K, meaning our observed understanding bound ϵ remains large at 0.412.

On GPT-4, we close the understanding gap on 2 datasets (HotpotQA and MultispanQA) and reduce the gap on 3 others to below 5.5%. However, discrimination continues to lag on GSM8K and Hellaswag datasets. In both HotpotQA and Multispan settings, we find remarkably that the model significantly outperforms the baseline discriminator as a result of prompting without outputting additional tokens.

We discover a prompting technique we refer to as Constrained Thought (see Figure 6), which involves encouraging the model to reason and reflect while simultaneously constraining the model to outputting a single token (e.g. in our case, TRUE or FALSE, but which could also be a final numerical or textual answer). We compare its performance to standard and chain of thought performance in Table 2. We note that on these datasets, Constrained Thought (CnT) outperforms Chain of Thought while simultaneously being significantly more output-token-efficient. We caveat, however, that the exact mechanism of this technique is still fairly elusive, and hence we do not give it much attention in this paper.

We show that we can make up a substantial portion of the understanding gap through prompting and are able to reach an upper bound on the understanding gap across all 10 datasets of 0.412. This performance in addition to the near-zero understanding gap produced suggests that it is possible to prompt the model to transition from reasoning-based to retrieval-based discrimination.

5.3 When self-correction algorithms work better?

Do algorithms based on models’ self-correction perform better on tasks with smaller understanding gaps? We find in the literature that the tasks with greatest improvement from prompting (41% boost over CoT in 24 game (Yao et al., 2023)). More curiously, however, some reflection-based algorithms 31% over CoT in MultispanQA (Dhuliawala et al., 2023))

6 Conclusion

Whereas prior works shed light onto the performance of language models on discrimination tasks relative to humans for tasks with generative accuracy parity, we present a hypothesis that specifically aims to better understand the understanding gap, the difference between generative and discriminative accuracies. We find most importantly, that even when we restrict the model to outputting only a single token, we can significantly improve discrimination accuracy to rival or surpass generation capabilities, even on tasks where the model is — in absence of any additional prompting — a much better generator.

Further study is required to investigate the generalizability of the estimated upperbound on the understanding gap, $\epsilon = 0.412$, established across the 10 datasets.

7 Ethical Considerations

We do not foresee any ethical considerations in direct relation to our work. While there are broader risks from general intelligence systems, and this research contributes towards our understanding of language models and ultimately our grasp of this goal, we hope that our paper provides interpretability to language model understanding. We hope to further the pursuit of gradually peeling back the black box that constitute many aspects of modern large language models.

8 Limitations

A result of our choice to evaluate understanding on the most capable and popular language models is that we experiment primarily on GPT-3.5 and GPT-4. We foresee potential limitations with evaluating solely on closed source language models. We also attempt to estimate the upper bound ϵ and other behaviour through a limited number of datasets. While we attempt to choose datasets spanning a broad range of skills, choose datasets before performing any experiments, and report results on all datasets regardless of performance, we ultimately only evaluate on 10 datasets which is a fraction of the full dataset space. HotpotQA is licensed under Apache-2.0, MultispanQA: no license and publicly available by authors, 24 Game: MIT, GSM8K: MIT, MMLU: MIT, COLLIE: MIT, CSQA: no license and publicly available, HellaSwag: MIT, RACE: no license and publicly available, and PIQA: Apache-2.0. Usage of benchmarks is consistent with intended use. All benchmarks are in English, and train/test/dev splits are as originally used on each dataset. We evaluate primarily on test splits, but use validation splits where ground truth is unavailable in the test split.

Acknowledgements

Grateful for TBD.

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. [PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains](#). *Transactions of the Association for Computational Linguistics*, 10:414–433.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439. Issue: 05.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. [Sparks of Artificial General Intelligence: Early experiments with GPT-4](#). ArXiv:2303.12712 [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].

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].

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. [Improving Factuality and Reasoning in Language Models through Multiagent Debate](#). ArXiv:2305.14325 [cs].

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiušė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. 2023. [The Capacity for Moral Self-Correction in Large Language Models](#). ArXiv:2302.07459 [cs].

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. [Making Pre-trained Language Models Better Few-shot Learners](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2022. [Efficient \(Soft\) Q-Learning for Text Generation with Limited Good Data](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6969–6991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring Massive Multitask Language Understanding](#). In *International Conference on Learning Representations*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. [Surface Form Competition: Why the Highest Probability Answer Isn’t Always Right](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. [Large Language Models Cannot Self-Correct Reasoning Yet](#). ArXiv:2310.01798 [cs].

John Schulman, Barret Zoph, and Christina Kim. 2022. [Introducing ChatGPT](#).

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. [RACE: Large-scale ReAding Comprehension Dataset From Examinations](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. [MultiSpanQA: A Dataset for Multi-Span Question Answering](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. [Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing](#). *ACM Computing Surveys*, 55(9):195:1–195:35.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon,

| | | | |
|-----|---|--|--|
| 719 | Nouha Dziri, Shrimai Prabhunoye, Yiming Yang, | | |
| 720 | Shashank Gupta, Bodhisattwa Prasad Majumder, | | |
| 721 | Katherine Hermann, Sean Welleck, Amir Yazdan- | | |
| 722 | bakhsh, and Peter Clark. 2023. Self-Refine: Iterative | | |
| 723 | Refinement with Self-Feedback . In <i>Thirty-seventh</i> | | |
| 724 | <i>Conference on Neural Information Processing Sys-</i> | | |
| 725 | <i>tems</i> . | | |
| 726 | OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, | | |
| 727 | Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale- | | |
| 728 | man, Diogo Almeida, Janko Altmenschmidt, Sam Alt- | | |
| 729 | man, Shyamal Anadkat, Red Avila, Igor Babuschkin, | | |
| 730 | Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim- | | |
| 731 | ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, | | |
| 732 | Jake Berdine, Gabriel Bernadett-Shapiro, Christo- | | |
| 733 | pher Berner, Lenny Bogdonoff, Oleg Boiko, Made- | | |
| 734 | laine Boyd, Anna-Luisa Brakman, Greg Brockman, | | |
| 735 | Tim Brooks, Miles Brundage, Kevin Button, Trevor | | |
| 736 | Cai, Rosie Campbell, Andrew Cann, Brittany Carey, | | |
| 737 | Chelsea Carlson, Rory Carmichael, Brooke Chan, | | |
| 738 | Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, | | |
| 739 | Ruby Chen, Jason Chen, Mark Chen, Ben Chess, | | |
| 740 | Chester Cho, Casey Chu, Hyung Won Chung, Dave | | |
| 741 | Cummings, Jeremiah Currier, Yunxing Dai, Cory | | |
| 742 | Decareaux, Thomas Degry, Noah Deutsch, Damien | | |
| 743 | Deville, Arka Dhar, David Dohan, Steve Dowling, | | |
| 744 | Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna | | |
| 745 | Eloundou, David Farhi, Liam Fedus, Niko Felix, | | |
| 746 | Simón Posada Fishman, Juston Forte, Isabella Ful- | | |
| 747 | ford, Leo Gao, Elie Georges, Christian Gibson, Vik | | |
| 748 | Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo- | | |
| 749 | Lopes, Jonathan Gordon, Morgan Grafstein, Scott | | |
| 750 | Gray, Ryan Greene, Joshua Gross, Shixiang Shane | | |
| 751 | Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, | | |
| 752 | Yuchen He, Mike Heaton, Johannes Heidecke, Chris | | |
| 753 | Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, | | |
| 754 | Brandon Houghton, Kenny Hsu, Shengli Hu, Xin | | |
| 755 | Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, | | |
| 756 | Joanne Jang, Angela Jiang, Roger Jiang, Haozhun | | |
| 757 | Jin, Denny Jin, Shino Jomoto, Billie Jonn, Hee- | | |
| 758 | woo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Ka- | | |
| 759 | mali, Ingmar Kanitscheider, Nitish Shirish Keskar, | | |
| 760 | Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, | | |
| 761 | Christina Kim, Yongjik Kim, Hendrik Kirchner, | | |
| 762 | Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz | | |
| 763 | Kondraciuk, Andrew Kondrich, Aris Konstantini- | | |
| 764 | dis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, | | |
| 765 | Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, | | |
| 766 | Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, | | |
| 767 | Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa | | |
| 768 | Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, | | |
| 769 | Kim Malfacini, Sam Manning, Todor Markov, Yaniv | | |
| 770 | Markovski, Bianca Martin, Katie Mayer, Andrew | | |
| 771 | Mayne, Bob McGrew, Scott Mayer McKinney, | | |
| 772 | Christine McLeavey, Paul McMillan, Jake McNeil, | | |
| 773 | David Medina, Aalok Mehta, Jacob Menick, Luke | | |
| 774 | Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie | | |
| 775 | Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, | | |
| 776 | Mira Murati, Oleg Murk, David Mély, Ashvin Nair, | | |
| 777 | Reiichiro Nakano, Rajeev Nayak, Arvind Neelakan- | | |
| 778 | tan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, | | |
| 779 | Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe | | |
| 780 | Palermo, Ashley Pantuliano, Giambattista Paras- | | |
| 781 | candolo, Joel Parish, Emy Parparita, Alex Passos, | | |
| | Mikhail Pavlov, Andrew Peng, Adam Perelman, Fil- | | |
| | ipe de Avila Belbute Peres, Michael Petrov, Henrique | | |
| | Ponde de Oliveira Pinto, Michael, Pokorný, Michelle | | |
| | Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, | | |
| | Boris Power, Elizabeth Proehl, Raul Puri, Alec | | |
| | Radford, Jack Rae, Aditya Ramesh, Cameron Ray- | | |
| | mond, Francis Real, Kendra Rimbach, Carl Ross, | | |
| | Bob Rotsted, Henri Roussez, Nick Ryder, Mario | | |
| | Saltarelli, Ted Sanders, Shibani Santurkar, Girish | | |
| | Sastry, Heather Schmidt, David Schnurr, John Schul- | | |
| | man, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, | | |
| | Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon | | |
| | Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, | | |
| | Katarina Slama, Ian Sohl, Benjamin Sokolowsky, | | |
| | Yang Song, Natalie Staudacher, Felipe Petroski Such, | | |
| | Natalie Summers, Ilya Sutskever, Jie Tang, Niko- | | |
| | las Tezak, Madeleine Thompson, Phil Tillet, Amin | | |
| | Toootoonchian, Elizabeth Tseng, Preston Tuggle, Nick | | |
| | Turley, Jerry Tworek, Juan Felipe Cerón Uribe, An- | | |
| | drea Vallone, Arun Vijayvergiya, Chelsea Voss, Car- | | |
| | roll Wainwright, Justin Jay Wang, Alvin Wang, Ben | | |
| | Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, | | |
| | Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian | | |
| | Weng, Matt Wiethoff, Dave Willner, Clemens Win- | | |
| | ter, Samuel Wolrich, Hannah Wong, Lauren Work- | | |
| | man, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, | | |
| | Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Woj- | | |
| | ciech Zaremba, Rowan Zellers, Chong Zhang, Mar- | | |
| | vin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang | | |
| | Zhuang, William Zhuk, and Barret Zoph. 2023. GPT- | | |
| | 4 Technical Report . ArXiv:2303.08774 [cs]. | | |
| | Noah Shinn, Federico Cassano, Ashwin Gopinath, | | |
| | Karthik R. Narasimhan, and Shunyu Yao. 2023. Re- | | |
| | flexion: language agents with verbal reinforcement | | |
| | learning . In <i>Thirty-seventh Conference on Neural</i> | | |
| | <i>Information Processing Systems</i> . | | |
| | Alon Talmor, Jonathan Herzig, Nicholas Lourie, and | | |
| | Jonathan Berant. 2019. CommonsenseQA: A Ques- | | |
| | tion Answering Challenge Targeting Commonsense | | |
| | Knowledge . In <i>Proceedings of the 2019 Conference</i> | | |
| | <i>of the North American Chapter of the Association for</i> | | |
| | <i>Computational Linguistics: Human Language Tech-</i> | | |
| | <i>nologies, Volume 1 (Long and Short Papers)</i> , pages | | |
| | 4149–4158, Minneapolis, Minnesota. Association for | | |
| | Computational Linguistics. | | |
| | Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, | | |
| | Adams Wei Yu, Brian Lester, Nan Du, Andrew M. | | |
| | Dai, and Quoc V. Le. 2022a. Finetuned Language | | |
| | Models are Zero-Shot Learners . In <i>International</i> | | |
| | <i>Conference on Learning Representations</i> . | | |
| | Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, | | |
| | Barret Zoph, Sebastian Borgeaud, Dani Yogatama, | | |
| | Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. | | |
| | Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy | | |
| | Liang, Jeff Dean, and William Fedus. 2022b. Emer- | | |
| | gent Abilities of Large Language Models . <i>Transac-</i> | | |
| | <i>tions on Machine Learning Research</i> . | | |
| | Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten | | |
| | Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V. Le, | | |

and Denny Zhou. 2022c. [Chain of Thought Prompting Elicits Reasoning in Large Language Models](#). In *Advances in Neural Information Processing Systems*.

Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena D. Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, Benjamin Newman, Pang Wei Koh, Allyson Ettinger, and Yejin Choi. 2024. [The Generative AI Paradox: “What It Can Create, It May Not Understand”](#). In *The Twelfth International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Howard Chen, Austin W. Hanjie, Runzhe Yang, and Karthik R. Narasimhan. 2024. [COLLIE: Systematic Construction of Constrained Text Generation Tasks](#). In *The Twelfth International Conference on Learning Representations*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R. Narasimhan. 2023. [Tree of Thoughts: Deliberate Problem Solving with Large Language Models](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a Machine Really Finish Your Sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Prompts used in Sub-Hypothesis 1

We use a number of prompts to encourage the language model to match or surpass generation performance, when validating answers.

Prompt P1 Think step by step. First generate your own answer to the question and then compare this with the provided answer. Check, then double check your thinking. The last word of your response should be 'TRUE' if the answer is correct and 'FALSE' if the answer is incorrect, given the question.

The effect of the prompts most effective at closing these understanding gaps across our trials are shown in Figure 5.

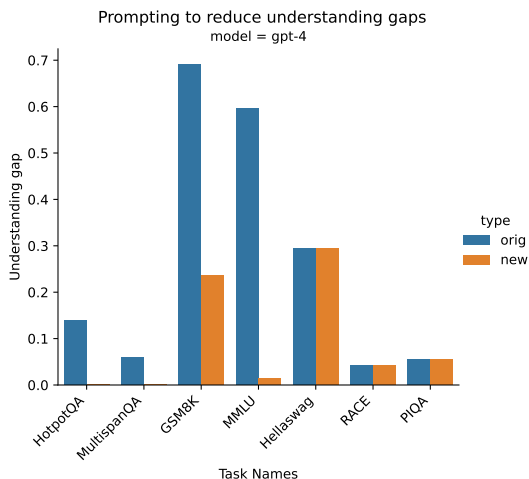


Figure 5: Reducing the understanding gap through prompting on GPT-4

B Prompting phenomena

We notice a remarkable phenomena when attempting to investigate our subhypothesis 1 on prompting. Despite prompting the model in a typical manner to generate a well-reasoned explanation and to engage in self-reflection, the model outputs a single token TRUE or FALSE answer (see Figure 6. While we are unsure of why this happens, we note that it allows us to achieve or surpass generator performance on many tasks.

The technique performs well in practice, outperforming the classical Chain of Thought (CoT) without require additional output tokens. The GPT-4 with Chain of Thought configuration surprisingly scores 0% because instead of responding with TRUE or FALSE, the most simply responds with a numerical answer to the problem itself. We try two different prompts You are given a question and a prediction. Answer 'TRUE' if the predicted answer is correct and 'FALSE' if the answer is incorrect. Think step by step. and Answer 'TRUE' if the answer is correct and 'FALSE' if the answer is incorrect. Think step by step.. In both instances, we do not receive a single generation that contains TRUE or FALSE which is very surprising and warrants future investigation. We do note that we use the latest gpt-4-1106-preview, so perhaps this is the result of recent changes to the model.

| | HotpotQA | MultispanQA | GSM8K |
|--------------------|------------|-------------|------------|
| GPT-3.5 | 60% | 73% | 40.30% |
| GPT-4 | 66% | 63% | 21.50% |
| CoT (GPT-4) | 71% | 63% | 0% |
| CnT (ours) (GPT-4) | 80% | 69% | 67% |

Table 2: Performance of Constrained Thought on GPT-4 measured against standard and Chain of Thought prompting on validation across each dataset. Note that GPT-3.5 performance actually outperforms GPT-4 on MultispanQA, despite achieving roughly similar performance in a generative mode.

Regular prompting

Model input

Sys: Answer 'TRUE' if the answer is correct and 'FALSE' if the answer is incorrect.

Q: Question: Roy has saved 40% more in money earned by chores than his brother Anthony. Anthony has saved \$10.00 more than their sister Eva. Eva has saved \$20.00. How much money does Roy have? Answer: 42

Model output

A: FALSE ❌

Chain of Thought

Model input

Sys: Answer 'TRUE' if the answer is correct and 'FALSE' if the answer is incorrect. Think step by step.

Q: Question: Roy has saved 40% more in money earned by chores than his brother Anthony. Anthony has saved \$10.00 more than their sister Eva. Eva has saved \$20.00. How much money does Roy have? Answer: 42

Model output

A: FALSE.

Let's calculate it step by step:

1. Eva has saved \$20.00.
2. Anthony has saved \$10.00 more than Eva, so Anthony has saved \$20.00 + \$10.00 = \$30.00.
3. Roy has saved 40% more in money earned by chores than Anthony. So, Roy has saved 40% of \$30.00 more than Anthony. 40% of \$30.00 is \$12.00.
4. So, Roy has saved \$30.00 + \$12.00 = \$42.00.

Given the calculation, the correct answer is that Roy has saved \$42.00, making the original statement **TRUE**. ✅

Constrained Thought (ours)

Model input

Sys: **Think step by step. First generate your own answer to the question and then compare this with the provided answer. Check, then double check your thinking. The last word of your response should be 'TRUE' if the answer is correct and 'FALSE' if the answer is incorrect, given the question.**

Q: Question: Roy has saved 40% more in money earned by chores than his brother Anthony. Anthony has saved \$10.00 more than their sister Eva. Eva has saved \$20.00. How much money does Roy have? Answer: 42

Model output

A: TRUE ✅

Figure 6: Constrained Thought, a prompting technique rivalling Chain of Thought without the additional output. As seen here, on more challenging problems, Chain of Thought may output more than one answer, which does not occur during our trials with Constrained Thought. The technique consists primarily of prompting the model to think and perform complex reasoning, but to then restrict the model to outputting a single character