Large Language Models' Cognitive Capabilities: A Study on OpenAI's GPT Models

Hailie Nguyen, Victor Cui, Cecilia Wu
Center for Data Science, New York Univerity
hln2020@nyu.edu, vyc8567@nyu.edu, czw206@nyu.edu

Abstract

Using tools from cognitive psychology, we study and experiment with GPT's cognitive capabilities in decision-making, information search, deliberation, and causal reasoning. More recent GPT models demonstrate remarkable improvements in their response accuracy compared to their predecessors. Prompt engineering techniques allowed us to enhance GPT's performance, even with a simple strategy. These results broaden our comprehension of present-day sophisticated language models, strengthen established findings, and lay the foundation for future work.

Keywords: GPT; cognitive psychology; prompt engineering

Introduction

Unleashing the power of OpenAI's GPT, we embark on an exploration of the remarkable cognitive capabilities that reside within its vast neural network. Our work is built upon existing work from "Using cognitive psychology to understand GPT-3" (Binz & Schulz, 2023), which examined the cognitive capabilities of GPT-3 using various canonical experiments. Two investigations were conducted: vignette-based experiments, which involved providing participants with a short and predetermined description of hypothetical scenarios, and task-based experiments, which generated scenarios programmatically on a trial-by-trial basis. During each investigation, the cognitive ability of GPT-3 was assessed in four well-known domains: decision-making, information search, deliberation, and causal reasoning. GPT-3 achieved a 50% accuracy for the vignette-based experiments and showed near human performance for task-based experiments, and even showed signatures of model-based reinforcement learning. The authors argued that studying the cognitive processes underlying GPT-3 can help us better understand the limitations and strengths of the model, as well as inform the development of future language models. The paper concluded by recommending that researchers "should not only scale up algorithms that are passively fed with data but instead let agents directly interact and engage with the world" (Binz & Schulz, 2023).

Are newer large language models better engaged with the world? With the advent of ChatGPT, GPT-4 and the concept of prompt engineering, we are interested in learning whether these models exhibit improvements in cognitive abilities.

We replicated the vignette-based and task-based experiments described in (Binz & Schulz, 2023) on GPT-3. For vignette-based experiments, we also extended them to assess the performance of GPT-3.5 and GPT-4, and came up with our own adversarial vignettes. While replication results are presented in Appendix I, in the discussion below, we focus on comparing GPT-3.5 and GPT-4 with GPT-3's performance in the original paper, and leave the validation of GPT-3's results to future work. For task-based investigations, while we wished to extend the experiments to GPT-3.5 and GPT-4, we ran into technical difficulties with GPT-3.5's API and were not able to access GPT-4's API due to its limited availability. For both sets of experiments, prompt engineering is implemented.

Background on GPT

GPT (Generative Pre-trained Transformer) is a state-of-the-art transformer-based language model developed by OpenAI. With every major releases counting from the earliest GPT-1 in June 2018 to the latest GPT-4 in March 2023, the model improved with larger model size and more training data to obtain better language understandings and zero-shot learning. More recently, the GPT-4, now accepting visual inputs, greatly outperformed GPT-3.5 and GPT-3 on exams designed for both human and machines. In addition, GPT-4 is more humane than before. Rather than the classical chatbot with fixed verbosity, tone and style, GPT-4 incorporates an additional safety measure to refuse disallowed and sensitive contents with an user prescribed tone. GPT-4 can be your personalized chemistry tutor, but it will refuse to synthesize dangerous chemicals (OpenAI, 2023).

Prompt engineering

Prompt engineering emerged as a result of the widespread adoption and popularity of large-scale pre-trained language models. Instead of the "pre-train, fine-tune" paradigm, prompt engineering refers to a "pre-train, prompt, and predict" procedure (Liu et al., 2023). Here, we applied prompt engineering by prefacing our series of questions with the following prompt: "I want you to act as a contestant to a question and answer game. I will ask you several questions related to decision-making, information search, deliberation, and causal reasoning abilities on a battery of canonical experiments from the literature. It is time to showcase your best performance."

Vignette-based Investigations

We entered canonical scenarios from cognitive psychology literature as prompts in ChatGPT, using GPT-3.5 and GPT-4, and compared their answers to GPT-3's answers recorded in the original paper. While we paid attention to the final answers' accuracy, we were also interested in whether their explanations were human-like. In addition to adversarial vignettes implemented in the original paper, we came up with new questions on our own to further assess the models' response to novel scenarios potentially not seen during training. We also applied the prompt engineering technique described above in GPT-3.5 in an attempt to achieve better quality answers, and evaluated the results.

Decision-making

GPT-3.5 and GPT-4's answers to the "Linda problem" (Linda, see Appendix I) were similar to that by GPT-3, which was a result of the conjunction fallacy. Interestingly, when prompt engineering was applied in GPT-3.5, the model chose the first option, "Linda is a bank teller", and pointed out that the second option would be an example of the conjunction fallacy.

For the "Cab problem" (Cab, see Appendix I), GPT-3 successfully avoided the base-rate fallacy and provided an approximately correct answer. GPT-3.5 with and without prompt engineering and GPT-4 all attempted to solve the problem using Bayes' theorem, thereby also successfully avoiding the base-rate fallacy. It is interesting to note that GPT-3.5 made a mistake in its calculation and provided a slightly different answer, which was not observed in answers by GPT-4 or GPT-3.5 using prompt engineering.

For the "Hospital problem" (Hospital, see Appendix I), GPT-3.5 with and without prompt engineering and GPT-4 had different approaches. While GPT-3.5 attempted to calculate the expected number of days in a year where more than 60% of babies born are boys, GPT-3.5 with prompt engineering used conditional probability, and GPT-4 cited the law of large numbers. Despite the difference in their approaches, all models arrived at the correct answer, which was an accuracy improvement from GPT-3's answer. However, GPT-3's answer, though incorrect, was considered "human-like", so GPT-3.5 and GPT-4 did not necessarily behave in a more "human-like" way than GPT-3 in this question. Nonetheless, we noted the later models' accuracy improvement.

Information search

For the "Toma v1 problem" (Toma, see Appendix I), GPT-3.5 and GPT-4 had slightly different explanations for their answer, though both gave the correct answer, as did GPT-3. While GPT-3.5 explained that option 1 would cover a broader range of possible explanations for Toma's tardiness, GPT-4 additionally pointed out that based on historical data, option 1 accounted for 3 out of the past 6 days when Toma was late. Interestingly, GPT-3.5 under prompt engineering suggested an alternative solution outside of the given options, namely asking Toma: "Why were you late today?". This demonstrates GPT-3.5's creativity in problem solving when prompt engineering is applied, though not sticking to the original question might be suboptimal in certain constrained scenarios, such as when the task is a multiple choice question rather than an open ended question.

For the "Toma v2 problem" (Toma, see Appendix I), all models again gave the right final answer, but it is important to note the quality of explanations for each model. All models noted that Toma's broken bicycle explained the majority of his tardiness in the past 8 days. However, GPT-3.5 incorrectly counted the number of days where Toma's bicycle was broken as 4, while GPT-4 correctly identified this number to be 5. GPT-3.5 also mentioned: "[Given] that Toma has only been late once due to watching TV and the majority of his lateness has been related to his bicycle or not finding his belongings, it may not be the most relevant question to ask in this case," implying that it did not consider watching TV to be a likely reason for Toma's tardiness. However, neither of the given options mentioned watching TV, so this could be considered a non-human-like explanation by GPT-3.5. GPT-3.5 under prompt engineering favored the question "Why were you late today?" similarly to its answer for the "Toma v1 problem", but this time it did pick option 2, rather than refusing to pick among the given options like in the previous question.

For the "Test problem" (Test, see Appendix I), both GPT-3.5 and GPT-4 picked the incorrect answer. GPT-3.5 fell for the congruence bias when stating "[s]ince the patient has a higher probability of having Chamber-of-Commerce disease (0.8) compared to Elk's disease (0.2), a positive result on a test for Chamber-of-Commerce disease is more likely to be correct." It also contradicted itself when saying "the tetherscopic examination has a higher probability of detecting Chamber-of-Commerce disease (90% chance of a positive result) compared to the intraocular smear for Elk's disease (90% chance of a positive result)". GPT-4, while using Bayes' theorem to answer this question, gave an incorrect answer nonetheless. GPT-3.5 with prompt engineering was able to give the correct answer, which is an accuracy improvement from GPT-3, GPT-3.5 without prompt engineering, and GPT-4.

GPT-3, GPT-3.5 with and without prompt engineering, and GPT-4 all gave the correct answer to the "Wason problem" (Wason, see Appendix I), similar to GPT-3.

Deliberation

In the CRT questions, GPT-3 gave intuitive, but incorrect answers. GPT-3.5 with prompt engineering and GPT-4 were able to overcome the tendency to give an incorrect fast response and derived the correct answers from further deliberation for all 3 questions. While GPT-3.5 was also able to give 2 out of 3 correct answers, for the "CRT2 problem" (CRT2, see Appendix I), it gave the correct answer during our initial round of experiment and gave an incorrect answer during a second round of experiment, even though it broke the problem down into a step-by-step solution during both rounds. It is interesting to note GPT-3.5's inconsistent behavior given the same problem.

Causal reasoning

For the "Blickets problem" (Blickets, see Appendix I), all models gave the correct answer that the first but not the second object was a blicket. While GPT-3.5 and GPT-4's explanations for why they thought object B is not a blicket were similar, GPT-3.5 with prompt engineering interpreted the question "Why is object B not a blicket?" as requesting potential speculations of object B's nature, causing it not to be a blicket, rather than asking about the logic behind its answer that object B is not a blicket, as understood by GPT-3.5 and GPT-4.

GPT-3, GPT-3.5 with and without prompt engineering, and GPT-4 all gave the same correct answer to the "Intervene problem" (Intervene, see Appedix I).

For the "Mature problem" (Mature, see Appendix I), it is interesting to see that GPT-3 and GPT-4 correctly gave definitive answers to all 4 questions, but GPT-3.5 without prompt engineering gave some ambiguous answers. When asked if the man would have died had he not taken pill B (question 2), and pill D (question 4), GPT-3.5 said it was unclear given he could have died from other causes. GPT-3.5 with prompt engineering contradicted itself in this question by saying no to pill D (question 4), but yes to pill B (question 2), even though the logic presented around these pills are analogous, thus exhibiting non-human-like errors.

Adversarial vignettes

In the "Black Cab problem" (Black Cab, see Appendix I), GPT-3.5 and GPT-4 successfully identified that there was no information given on black cabs to calculate the probability that the cab involved in the accident was black, which is an improvement from GPT-3's answer (0.2). GPT-3.5 with prompt engineering also pointed this out, but went a step further and assumed that the question could refer to blue cabs instead, and calculated the probability that the cab involved in the accident was blue. All models failed to infer that the probability of a black cab involving in the accident is 0, given that 85% of the cabs are green and 15% are blue (i.e. 100% of the cabs are either green or blue), which is an inference a human would be able to make.

For "Reverse Wason" (Reverse Wason, see Appendix I), GPT-3.5 with and without prompt engineering and GPT-4 gave consistent responses to those given during the "Wason problem", which is an improvement compared to GPT-3's performance, in which the answer was changed when the card order was switched.

In "Wrong CRT", GPT-4 pointed out the error in the question, and proceeded to solve the problem assuming the typo was corrected, while GPT-3.5 with and without engineering started solving the problem right away, ignoring the typo as if it had been corrected. GPT-3, as seen in the previous section on CRT, gave the same intuitive but incorrect answer.

Finally, for the set of 3 questions in "Immature Blicket" (Immature Blicket, see Appendix I), GPT-3.5 gave a response for the final question that was contradictory to its responses for the previous questions, while GPT-4 pointed out the contradiction in the information given and gave a logical explanation. GPT-3.5 with prompt engineering seemed to attempt to reconcile the information given in all parts of the question, and proposed hypotheses that would satisfy all pieces of information given. However, there were still some non-human-like contradictions in its logic, such as when it stated: "[If] the box made a sound when a green and yellow object were placed on it, and it did not make a sound when a yellow object was placed on it alone, then the yellow object is necessary to trigger the sound."

Adversarial vignettes - Extension

We attempted to come up with additional adversarial vignettes to examine all models' performance. For all three questions, GPT-3 simply repeated its answers previously given in response to the CRT prompts on which our adversarial vignettes were based. GPT-3.5 with and without prompt engineering started out with sound reasoning (Quadruple Lily, see Appendix I), but began to demonstrate non-human-like logic towards the end of its responses for all questions. GPT-4 also exhibited non-human-like inconsistencies for the "Quadruple Lily" problem, but was able to point out the potential mistake in the "Unproductive Machines" problem (Unproductive Machines, see Appendix I) and managed to give a consistent answer to the information given. For the "Half Day" problem (Half Day, see Appendix I), GPT-4 successfully gave the correct answer.

Vignette-based investigations - Summary

Assessing all models' performance on the vignette-based questions presented in Binz and Schulz (2023), we can see that GPT-3.5 and GPT-4 demonstrated remarkable improvements in their response accuracy compared to GPT-3. Surprisingly, a simple prompt engineering strategy also enhanced GPT-3.5's response quality for some questions where GPT-3.5 and GPT-4 fell for the same biases as humans and GPT-3 would. While we noticed some non-human-like behaviors from GPT-3.5 (with or without prompt engineering), they all came from its explanations rather than its final answers on which GPT-3 was assessed. If we were only to compare final answers, GPT-3.5 still outperformed GPT-3 overall. GPT-4 seems to behave the most human-like, even though it fell for some fallacies or biases that GPT-3.5 with prompt engineering managed to avoid.

It is interesting to note that when we came up with our own adversarial vignettes, GPT-3 repeated its previous answers, supporting the authors' suspicion that it might just be repeating what it had seen in the training data. GPT-3.5 with and without prompt engineering start to exhibit obvious non-human-like inconsistencies. While GPT-4 failed to reason through the "Quadruple Lily" problem, it was able to detect potential errors in the "Unproductive Machines" problem, and successfully solved the "Half Day" problem. Overall, GPT-4 seemed to act more human-like and showed stronger cognitive capabilities than its predecessors, GPT-3 and GPT-3.5, based on its vignette-based performance. Next, we turned to replicate the task-based experiments to further assess these models' cognitive capacities.

Task-based Investigations

We replicated task-based experiments presented by Binz and Schulz (2023) by implementing our own code based on the authors' GitHub repository[1] on the GPT-3 "Davinci" model, the most powerful GPT-3 model. Due to cost concerns, we decided to replicate the same procedures on a smaller subset of the data. Replicated results are shown and compared to the original paper in the following sections. In general, our results are similar to those found in the original paper and support the authors' claims. Since prompt engineering had a remarkable impact on vignette-based results, we extended prompt engineering to task-based experiments on GPT-3.

Decision-making

Binz and Schulz (2023) evaluated GPT-3's decision making capability using the "choice13k" dataset (Peterson, Bourgin, Agrawal, Reichman, & Griffiths, 2021; Bourgin, Peterson, Reichman, Russell, & Griffiths, 2019). While they tested different models in the GPT-3 family and compared them with human data, we focused on the "Davinci" model, which is the largest model in the family and the only one that had above chance performance in the original experiment. The full "choice13k" dataset contains over 13,000 problems. We executed our replication on 5,000 randomly selected problems from this dataset. Model performance was measured using a "regret" value, which is defined as the difference between the expected outcome of the optimal option and that of the chosen option. In other words, a lower regret value indicates a better decision. Figure 1 shows our model performance in comparison to the data achieved in Binz and Schulz (2023). Similar to the original paper, our results indicated that the "Davinci" model had decent above chance performance with $p < 0.001$, but did not reach human performance.

To evaluate the efficiency of prompt engineering, we conducted the experiment on 1,000 randomly selected problems by prefacing our questions with the engineered prompt described in the previous section, and compared the result with 1,000 randomly selected problems from the original results. As shown in Figure 2, prompt engineering did not have a significant effect on the model's decision-making output.

Since the performance of "Davinci" was statistically significant, Binz and Schulz (2023) further evaluated the model using 17 carefully selected problem pairs inspired by Kahneman and Tversky (1972) to evaluate whether the model exhibited human-like cognitive biases. Human data used for comparison was adapted from Ruggeri et al. (2020). During replication, we applied prompt engineering to the problems and found our results to be comparable to those from Binz and Schulz (2023) (Figure 3). Our replication results identified GPT-3 without prompt engineering (marked as triangles in Figure 3) to exhibit a certainty effect (in which a subject prefers guaranteed outcomes with slightly lower expected values to risky ones), a framing effect (in which a subject's preference depends on whether a choice is presented in terms of gains or losses), and an overweighting bias (in which a subject weights a difference between small probabilities higher than that between larger probabilities). Our results for GPT-3 with prompt engineering (marked as crosses in Figure 3) did not indicate a certainty effect or a framing effect. Prompt engineering seemed to have successfully removed two common biases from the model's responses.

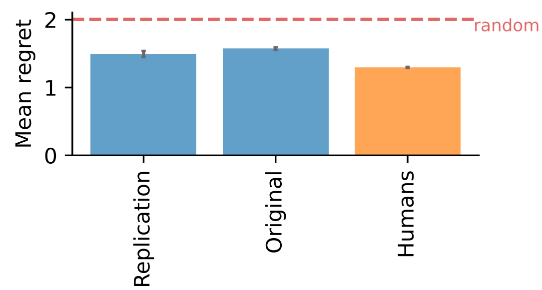

Figure 1: GPT-3 Davinci performance on decision making dataset measure with mean regret.

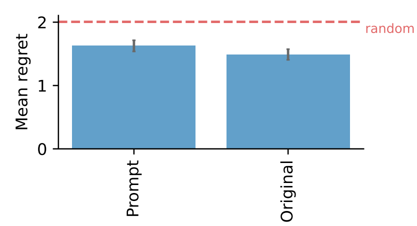

Figure 2: Comparison of GPT-3 Davinci performance using the original prompt ("original") vs with added prompt ("prompt"), on 1,000 randomly selected problems.

Information search

Wilson's horizon task was used to evaluate GPT-3's information search-related behaviors (Wilson, Geana, White, Ludvig, & Cohen, 2014). The task involves implementing a two-armed bandit paradigm, where the subject ("Davinci" in this case) is provided with four forced-choice trials with the two arms (where each round means a pull) and the outcomes. The subject is then asked to choose an arm to pull to maximize expected outcome in the next one round (Horizon 1) or in the next six rounds (Horizon 6). This multi-armed bandit test evaluates how GPT-3 handles the exploration-exploitation trade-off. Each test trial contains one of the two types of past

[1] https://github.com/marcelbinz/GPT3goesPsychology

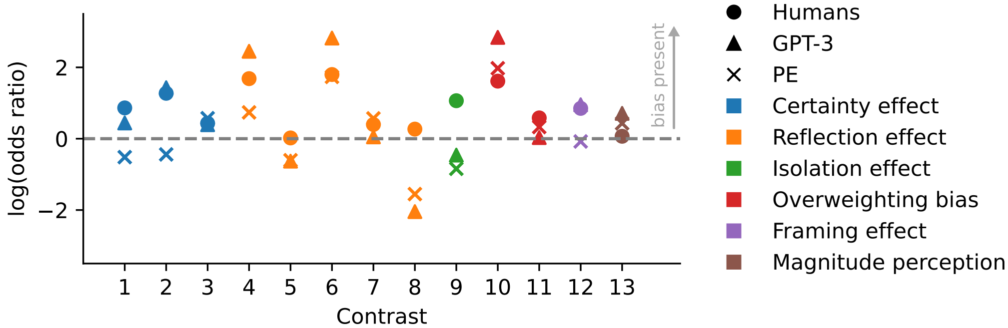

Figure 3: Log-odds ratios of different contrasts used to test for cognitive biases. ”GPT-3” refers to replication. ”PE” refers to prompt engineering. Positive values suggest presence of effect/bias.

experience: two pulls for each arm (equal), or one pull for one arm and three pulls for the other (unequal).

Binz and Schulz (2023) ran 3,200 trials on GPT-3 for both equal and unequal conditions. For replication, we ran 640 trials each for GPT-3 with and without prompt engineering. Trials were randomly generated using the Python package *gym* developed by OpenAI (Brockman et al., 2016). Figure 4 shows the mean regret of GPT-3 from our replication with and without prompt engineering, compared to that of GPT-3 from the original paper and human data (Zaller, Zorowitz, & Niv, 2021). Our results approximate the original results closely with $p > 0.2$. GPT-3's Horizon 1 performances were indistinguishable from human data with $p > 0.3$, while its Horizon 6 performances were better than human performance before the last round, but fell slightly below human performance during the last round, and showed no improvement trend along the time horizon.

Binz and Schulz (2023) further fitted a logistic regression model on equal and unequal trials to investigate GPT-3's take on the exploration-exploitation trade-off. We replicated this process on our GPT-3 results with and without prompt engineering, and found similar results to those reported in the original paper (Appendix IV).

In summary, our results suggested that GPT-3 used random exploration but without a clear strategy. There was no correlation between the estimated reward difference and the horizon. Additionally, GPT-3 did not seem to employ directed exploration, as it did not show a positive effect on the horizon under unequal condition as observed in humans. Our prompt engineering approach did not seem to affect its strategy choices. A different prompt related to exploration or exploitation may elicit more information on the true potential of GPT-3, and may be explored for future work.

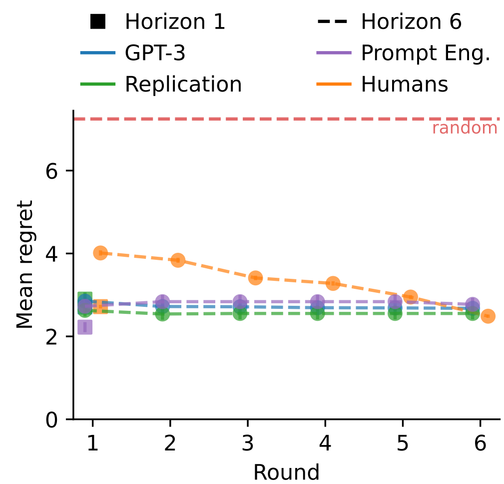

Figure 4: Mean regret for GPT-3 in original paper (”GPT-3”), replication, replication with prompt engineering, and human data. Lower regret means better performance.

Deliberation

The task-based experiment on deliberation focuses on model-free vs. model-based learning in GPT-3. For instance, if a strategy leads to a good outcome, model-free learning will exploit this strategy, while model-based learning will try to learn from this experience and update its strategy. Binz and Schulz (2023) implemented a test with a two-step paradigm, which included 100 simulations in total and 20 repetitions per simulation. From their results, Binz and Schulz (2023) concluded that GPT-3 used a model-based approach, which contradicted their findings in the vignette-based deliberation test (CRT problems). For our replication, we performed 25 simulations with 20 repetitions per simulation for GPT-3 with and without prompt engineering. Our large p-values suggested that we cannot draw statistically significant conclusions from this test (Appendix II).

Causal reasoning

In Binz and Schulz (2023), GPT-3's causal reasoning ability was tested on two conditions: the causal-chain condition and the common-cause condition. The test format was inspired by human experiments in Waldmann and Hagmayer (2005). It involved providing 20 observations of a three-variable system (denoted as A, B, and C) along with the system's causal structure and specific information on the variables. An example of a causal-chain condition is "A causes B, and B causes C", and an example of a common-cause condition is "B causes both A and C". The model was asked to predict the expected existence of C under four conditions of B: observed present/absent (observation), or added/removed (intervention). We replicated the experiments under both the causal-chain and common-cause conditions on "Davinci". Additionally, we were able to extend this experiment on GPT-3.5. GPT-3.5 had the tendency to respond with vague answers and avoid giving exact numbers, so we asked it to give a definitive answer by adding "You need to give a number" to the end of our prompts. Although it still did not give exact estimations in some cases, we were able to record its answers by averaging the answers given in its responses. We also applied prompt engineering (PE) to GPT-3.5, which resulted in even more inconclusive answers. We finally edited the prompt to ask for an approximation, and recorded responses by GPT-3.5 with and without engineering in Appendix III. All results are shown in Figure 5 (causal-chain) and Figure 6 (common-cause).

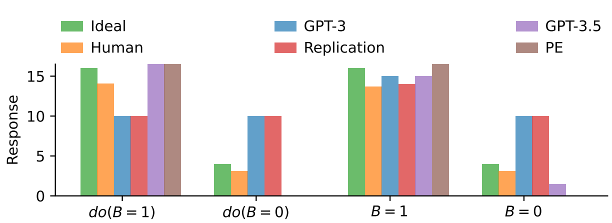

Figure 5: Causal reasoning test results for causal-chain condition. "Ideal" and "Human" are results taken from Waldmann and Hagmayer (2005). "GPT-3" represents the original GPT-3 results in Binz and Schulz (2023). "Replication" is our results on GPT-3. "GPT-3.5" is our results on GPT-3.5 without prompt engineering. "PE" refers to our results on GPT-3.5 with prompt engineering. *do()* means intervention, observation otherwise. Maximum is 20, no bar means 0.

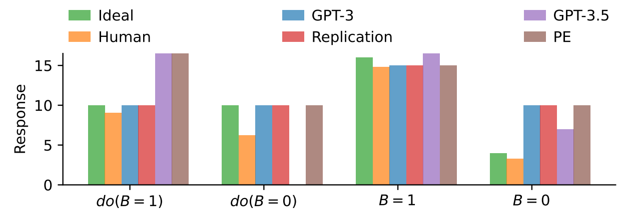

Figure 6: Causal reasoning test results for common-cause condition. "GPT-3" represents the original GPT-3 results in Binz and Schulz (2023). "Replication" is our results on GPT-3. "GPT-3.5" is our results on GPT-3.5 without prompt engineering. "PE" refers to our results on GPT-3.5 with prompt engineering. *do()* means intervention, observation otherwise. Maximum is 20, no bar means 0.

Our replication results followed that of the original paper very closely. GPT-3 showed comparable patterns with ideal and human responses under the common-cause condition except when observing B=0. Its responses were noticeably different from ideal/human responses under the causal-chain condition. We noticed that it made the same predictions as it did under the common-cause condition. This suggests that GPT-3 had trouble with causal reasoning tasks.

Interestingly, GPT-3.5 showed better performance under the causal-chain condition, but seemed to struggle more than GPT-3 under the common-cause condition. It had the tendency to give similar responses, or responses in the same directions, under all four situations. GPT-3.5 also showed a tendency to make extreme predictions (20/all or 0/none) more often. One might suggest that GPT-3.5 is very confident in its thinking directions, but considering that GPT-3.5 actively questioned the correctness of the questions and did not give definitive answers without enough information, which was in line with our observations during the vignette-based experiments, this extreme response pattern might result from being asked to make definitive predictions. Finally, prompt engineering did not seem to have an observable effect here, and sometimes even worsened the model performance.

Altogether, GPT-3.5 exhibited strong reasoning ability under the causal-chain condition, but weak performance under the common-cause condition. GPT-3 and GPT-3.5 appeared to have distinct patterns of causal reasoning.

Task-based investigations - Summary

Our replicated decision-making, information search, and causal reasoning results on GPT-3 matched that of the original paper closely, while our deliberation experiment was not statistically significant to draw a conclusion. The prompt engineering effect varied on a case-by-case basis. While no significant improvement was observed for most tasks, prompt engineering reduced the certainty and framing effect in decision making. Due to high demands of OpenAI's API, we had difficulty running large tasks on GPT-3.5 or gain access to GPT-4. When we had the chance to run causal reasoning experiments on GPT-3.5, we found that GPT-3.5 was stronger than GPT-3 at reasoning under the causal-chain condition, but it struggled more than its predecessor under the common-cause condition.

Discussion and Future Work

For vignette-based investigations, we found that GPT-3.5 and GPT-4 demonstrated remarkable accuracy improvements compared to GPT-3. A simple prompt engineering strategy was able to even further improve the accuracy of GPT-3.5, surpassing GPT-4's performance in some questions. However, when confronted with adversarial vignettes likely not seen during training, GPT-3.5 started to give illogical answers that are not human-like. GPT-4, in this regards, was able to perform better. Even though GPT-4 fell for some human-like biases less observed in GPT-3.5 using prompt engineering, we assessed GPT-4's overall performance to be better than its predecessors due to its ability to recognize and handle adversaries.

For task-based investigations, we were able to produce similar results on GPT-3 with original results by Binz and Schulz (2023). Due to resource constraints, we were only able to run causal reasoning experiments on GPT-3.5. GPT-3.5 appeared better than GPT-3 at reasoning under the causal-chain condition, but underperformed under the common-cause condition. Overall, prompt engineering brought nuances to the scene, but its enhancement depended on the specific task.

Due to funding constraints, we reduced the dataset size and the number of repetitions for our task-based replications. It will be worthwhile to test the experiment methods on the full dataset and on other LLMs in the future. Although our vignette-based results and the GPT-4 technical report (OpenAI, 2023) both indicated that GPT-4 is more human-like than ever, we were not able to conduct task-based experiments on GPT-4 and assess its performance due to its high demand.

Our experiment results suggest that newer GPT models seem to be better engaged with the world and have improved in their cognitive capabilities. This led us to question whether this conclusion can be generalized to other state-of-the-art large language models. Potential directions for future work include running the full experiments with more data and on GPT-4, as well as other LLMs, and testing out a variety of prompt engineering techniques, such as asking GPT to produce better prompts for itself. In addition, there can potentially be other psychological experiments used to study the cognitive capacities of LLMs. As OpenAI had acknowledged[2], there are many known limitations to their current models, such as hallucinations and adversarial prompts. These limitations are confirmed through out findings, and could be studied as an area of improvement for future LLMs.

[2]https://openai.com/product/gpt-4

References

Binz, M., & Schulz, E. (2023). Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, *120*(6), e2218523120. Retrieved from `https://www.pnas.org/doi/abs/10.1073/pnas.2218523120` doi: 10.1073/pnas.2218523120

Bourgin, D. D., Peterson, J. C., Reichman, D., Russell, S. J., & Griffiths, T. L. (2019, 09–15 Jun). Cognitive model priors for predicting human decisions. In *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 5133–5141). PMLR.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *Openai gym.*

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive Psychology*, *3*, 430–454.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023, jan). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, *55*(9). Retrieved from `https://doi.org/10.1145/3560815` doi: 10.1145/3560815

OpenAI. (2023). *Gpt-4 technical report.*

Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., & Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, *372*(6547), 1209–1214. doi: 10.1126/science.abe2629

Ruggeri, K., Alí, S., Berge, M., Bertoldo, G., Bjørndal, L., Cortijos Bernabeu, A., ... Folke, T. (2020, 06). Replicating patterns of prospect theory for decision under risk. *Nature Human Behaviour*. doi: 10.1038/s41562-020-0886-x

Waldmann, M. R., & Hagmayer, Y. (2005). Seeing versus doing: Two modes of accessing causal knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*, 216.

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore-exploit dilemma. *Journal of Experimental Psychology: General*, *143*(6), 2074–2081. doi: 10.1037/a0038199

Zaller, I., Zorowitz, S., & Niv, Y. (2021). Information seeking on the horizons task does not predict anxious symptomatology. *Biological Psychiatry*, *89*(9), S217–S218.

Appendix I: Vignette-based prompts and responses

Decision-making: Heuristics and biases

Linda Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.

Q: Which option is the most probable?
- Option 1: Linda is a bank teller.
- Option 2: Linda is a bank teller and is active in the feminist movement.
- Option 3: Linda is a member of the NRA.

- **GPT-3 (original)**: Option 2.

- **GPT-3 (replicated)**: Option 2 is the most probable.

- **GPT-3.5**: Based on the information given, the most probable option is Option 2: Linda is a bank teller and is active in the feminist movement.

  This is because the information provided suggests that Linda is passionate about issues of discrimination and social justice. Her participation in anti-nuclear demonstrations suggests that she is likely to be politically engaged and involved in social activism. Additionally, her major in philosophy suggests that she has an interest in critical thinking and questioning societal norms.

  Given this background, it is reasonable to assume that Linda is likely to be active in causes related to social justice and equality. Thus, it is possible that she is a member of the feminist movement, which advocates for women's rights and works towards gender equality. The fact that she is a bank teller is not necessarily contradictory to her activism, as many individuals balance a professional career with social and political involvement.

  Option 3 is highly unlikely, as Linda's interests and values do not align with those of the National Rifle Association (NRA), which advocates for gun rights and opposes gun control laws.

- **GPT-3.5 with prompt engineering**: The most probable option is Option 1: Linda is a bank teller. This is because we have no additional information that suggests that Linda is also a feminist or a member of the NRA, whereas being a bank teller is a plausible occupation for someone with a degree in philosophy. This scenario is an example of the conjunction fallacy, where people tend to judge the likelihood of two events occurring together as greater than the likelihood of either event occurring alone.

- **GPT-4**: Based on the given information, Option 2: Linda is a bank teller and is active in the feminist movement is the most probable. This is because the description of Linda highlights her concern for social justice and discrimination issues, which aligns with the feminist movement's goals. The other two options do not match as closely with the information provided.

Cab A cab was involved in a hit-and-run accident at night. Two cab companies, the Green and the Blue, operate in the city. You are given the following data: 85% of the cabs in the city are Green and 15% are Blue. A witness identified the cab as Blue. The court tested the reliability of the witness under the circumstances that existed on the night of the accident and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

Q: What is the probability that the cab involved in the accident was Blue rather than Green?

- **GPT-3 (original)**: A: The probability that the cab involved in the accident was Blue rather than Green is 0.4.

- **GPT-3 (replicated)**: P(Blue|witness) = P(witness|Blue) * P(Blue) / P(witness)

  P(witness|Blue) = 0.8

  P(Blue) = 0.15

  P(witness) = P(witness|Blue) * P(Blue) + P(witness|Green) * P(Green)

  P(witness) = 0.8 * 0.15 + 0.2 * 0.85

  P(witness) = 0.23

  P(Blue|witness) = 0.8 * 0.15 / 0.23

  P(Blue|witness) = 0.5652173913043478

- **GPT-3.5**: To answer this question, we can use Bayes' theorem:

  P(Blue cab | witness identified cab as Blue) = P(witness identified cab as Blue | Blue cab) * P(Blue cab) / P(witness identified cab as Blue)

  We can calculate the three probabilities on the right-hand side of the equation as follows:

  P(witness identified cab as Blue | Blue cab) = 0.8 (given in the problem statement)

  P(Blue cab) = 0.15 (given in the problem statement)

  P(witness identified cab as Blue) = P(witness identified cab as Blue | Blue cab) * P(Blue cab) + P(witness identified cab as Blue | Green cab) * P(Green cab)

  To calculate P(witness identified cab as Blue | Green cab), we can use the fact that the witness fails to identify the color correctly 20% of the time. Since 85% of the cabs in the city are Green, the probability of the witness identifying a Green cab as Blue is 0.2 * 0.85 = 0.17.

  Therefore,

  P(witness identified cab as Blue) = 0.8 * 0.15 + 0.17 * 0.85 = 0.2455

  Substituting these values into Bayes' theorem, we get:

  P(Blue cab | witness identified cab as Blue) = 0.8 * 0.15 / 0.2455 = 0.492

  Therefore, the probability that the cab involved in the accident was Blue rather than Green is approximately 49.2%.

- **GPT-3.5 with prompt engineering**: 41%: To answer this question, we can use Bayes' theorem.

  Let's define:

  - A: the event that the cab involved in the accident was Blue. - B: the event that the witness identified the cab as Blue.

  We want to calculate P(A—B), the probability that the cab involved in the accident was Blue given that the witness identified the cab as Blue.

  Using Bayes' theorem, we have:

  P(A|B) = P(B|A) * P(A) / P(B)

  P(B|A) is the probability that the witness correctly identified the cab as Blue, which is given as 0.8. P(A) is the prior probability of the cab being Blue, which is 0.15. To calculate P(B), we need to consider two possibilities: the cab is either Green or Blue.

  - If the cab is Green, the probability that the witness identified it as Blue is 0.2, because the witness fails to identify the color 20% of the time.

  - If the cab is Blue, the probability that the witness identified it as Blue is 0.8, because the witness correctly identifies the color 80% of the time.

  So, we have:

  P(B) = P(B|Green) * P(Green) + P(B|Blue) * P(Blue)

  = 0.2 * 0.85 + 0.8 * 0.15

  = 0.29

  Therefore, we can calculate:

  P(A|B) = P(B|A) * P(A) / P(B)

  = 0.8 * 0.15 / 0.29

  = 0.41

  So the probability that the cab involved in the accident was Blue rather than Green, given that the witness identified the cab as Blue, is 0.41 or 41%.

- **GPT-4**: To find the probability that the cab involved in the accident was Blue rather than Green, we can apply Bayes' theorem. Let B be the event that the cab is Blue, G be the event that the cab is Green, and W be the event that the witness identified the cab as Blue.

  We are given the following probabilities:

  P(B) = 0.15 (15% of the cabs are Blue)

P(G) = 0.85 (85% of the cabs are Green)

P(W|B) = 0.80 (the witness correctly identifies a Blue cab 80% of the time)

P(W|G) = 0.20 (the witness incorrectly identifies a Green cab as Blue 20% of the time)

We want to find P(B|W), the probability that the cab is Blue given that the witness identified it as Blue. Bayes' theorem states:

P(B|W) = P(W|B) * P(B) / P(W)

To find P(W), we can use the law of total probability:

P(W) = P(W|B) * P(B) + P(W|G) * P(G)

Plugging in the given probabilities:

P(W) = 0.80 * 0.15 + 0.20 * 0.85

P(W) = 0.12 + 0.17

P(W) = 0.29

Now we can find P(B|W):

P(B|W) = P(W|B) * P(B) / P(W)

P(B|W) = (0.80 * 0.15) / 0.29

P(B|W) $\approx$ 0.4138

So, the probability that the cab involved in the accident was Blue rather than Green is approximately 41.38%.

Hospital A certain town is served by two hospitals. In the larger hospital, about 45 babies are born each day, and in the smaller hospital, about 15 babies are born each day. As you know, about 50% of all babies are boys. The exact percentage of baby boys, however, varies from day to day. Sometimes it may be higher than 50%, sometimes lower. For a period of 1 year, each hospital recorded the days on which more than 60% of the babies born were boys.
Q: Which hospital do you think recorded more such days?
- Option 1: The larger hospital.
- Option 2: The smaller hospital.
- Option 3: About the same (i.e., within 5% of each other).

- **GPT-3 (original)**: A: Option 3.

- **GPT-3 (replicated)**: Option 1: The larger hospital.

- **GPT-3.5**: Let's start by calculating the expected number of days in a year where more than 60% of babies born are boys in each hospital.

  For the larger hospital, the expected number of boys born in a day is 45 * 0.5 = 22.5. Assuming a normal distribution for the number of boys born each day, the standard deviation is $\sqrt{45*0.5*0.5} = 3.36$. Therefore, the probability of more than 60% of babies being boys on any given day is the probability of observing more than 27 boys, which can be calculated using the normal distribution. This probability is very low, about 0.0013.

  Over a year of 365 days, we would expect about 0.0013 * 365 = 0.47 days where more than 60% of babies born are boys in the larger hospital.

  Similarly, for the smaller hospital, the expected number of boys born in a day is $15*0.5 = 7.5$, with a standard deviation of $\sqrt{15*0.5*0.5} = 1.94$. The probability of more than 60% of babies being boys on any given day is the probability of observing more than 9 boys, which is about 0.025.

  Over a year of 365 days, we would expect about 0.025 * 365 = 9.13 days where more than 60% of babies born are boys in the smaller hospital.

  Therefore, it is much more likely that the smaller hospital recorded more days where more than 60% of babies born were boys.

  Option 2: The smaller hospital is the most likely answer.

- **GPT-3.5 with prompt engineering**: To answer this question, we need to use the binomial distribution to calculate the probability of having more than 60% boys on a given day, given the number of babies born in each hospital.

  Let's define:

  - A: the event that more than 60% of babies born on a given day are boys.

  - B1: the event that the babies are born in the larger hospital.

  - B2: the event that the babies are born in the smaller hospital.

  We can calculate the probability of event A for each hospital using the binomial distribution:

  - For the larger hospital: $P(A|B1) =$sum$[k > 0.6 * 45](45 \text{ choose } k) * 0.5^{45}$

  - For the smaller hospital: $P(A|B2) =$sum$[k > 0.6 * 15](15 \text{ choose } k) * 0.5^{15}$

  Using a computer or calculator, we can find that P(A|B1) is approximately 0.000028 and P(A|B2) is approximately 0.19.

  Therefore, it is much more likely that the smaller hospital recorded more days on which more than 60% of babies born were boys. The answer is Option 2: The smaller hospital.

- **GPT-4**: Option 2: The smaller hospital.

  This question is related to the law of large numbers, which states that as the number of observations increases, the sample mean approaches the population mean. In this case, the population mean is the percentage of baby boys, which is 50%.

  In the larger hospital, there are more babies born each day (45) than in the smaller hospital (15). This means that the larger hospital's daily percentage of baby boys is more likely to be closer to the population mean of 50% due to the larger sample size. In contrast, the smaller hospital, with fewer babies born each day, is more likely to have a greater variation in the daily percentage of baby boys, making it more likely to record days with more than 60% of the babies born being boys.

Information search: Questions and hypothesis tests

Toma v1 In the last 6 days, Toma always arrived late to school. On the first day, Toma was late because he could not find his jacket. On the second day, Toma was late because his bicycle broke. On the third day, Toma was late because his bicycle broke. On the fourth day, Toma was late because he could not find his shoes. On the fifth day, Toma was late because he was watching TV. On the sixth day, Toma was late because he could not find his books. Today, on the seventh day, Toma was again late to school.

Q: Which of the following question would be better if wanted to find out why Toma was again late to school.
-Option 1: Were you late because you could not find something?
-Option 2: Were you late because your bicycle was broken?

- **GPT-3 (original)**: A: Option 1.

- **GPT-3 (replicated)**: Option 1 would be better if you wanted to find out why Toma was late to school on the seventh day.

- **GPT-3.5**: Option 1: Were you late because you could not find something? would be a better question to ask if you wanted to find out why Toma was again late to school.

  Based on the information provided, Toma has been late for different reasons on each of the past 6 days. It is possible that he was late on the seventh day for a completely different reason than the reasons he was late on the previous six days.

  Option 1 is a more general and open-ended question that allows Toma to explain any potential reason for being late, whether it is related to finding something or not. Option 2, on the other hand, assumes that the reason for being late is related to Toma's bicycle being broken, which may not be the case.

- **GPT-3.5 with prompt engineering**: Neither of the two options seems to be the best question to ask in this case, since Toma has been late for different reasons on different days. A better question would be:

  - Option 3: Why were you late today?

  This question allows Toma to explain the specific reason for being late on the seventh day, which could be different from any of the previous reasons.

  If I had to choose between the two options, Option 1: "Were you late because you could not find something?" would be slightly better than Option 2: "Were you late because your bicycle was broken?" because it is a more general question that could apply to any possible reason for being late, whereas Option 2 assumes that Toma's bicycle is still a potential reason for being late. However, it's still possible that Toma was late for a different reason on the seventh day, so Option 3 remains the best question to ask.

- **GPT-4**: Option 1: Were you late because you could not find something?

  This question is better for finding out why Toma was again late to school because it covers a broader range of potential reasons based on the past incidents. On three out of the six previous days, Toma was late due to not being able to find something (his jacket, shoes, or books). In comparison, Option 2 is more specific and only addresses one possible reason (broken bicycle) that caused Toma to be late on two out of the six previous days.

Toma v2 In the last 8 days, Toma always arrived late to school. On the first day, Toma was late because he could not find his jacket. On the second day, Toma was late because his bicycle broke.On the third day, Toma was late because his bicycle broke.On the fourth day, Toma was late because he could not find his shoes. On the fifth day, Toma was late because his bicycle broke. On the sixth day, Toma was late because his bicycle broke. On the seventh day, Toma was late because he was watching TV. On the eighth day, Toma was late because his bicycle broke.

Today, on the ninth day, Marcel was again late to school.

Q: Which of the following question would be better if wanted to find out why Toma was again late to school.

-Option 1: Were you late because you could not find something?

-Option 2: Were you late because your bicycle was broken?

- **GPT-3 (original)**: A: Option 2.
- **GPT-3 (replicated)**: Option 2 would be better if you wanted to find out why Toma was late to school again.
- **GPT-3.5**: Option 2: Were you late because your bicycle was broken? would be a better question to ask if you wanted to find out why Toma was again late to school.

  Based on the information provided, Toma has been late for different reasons on the past 8 days. However, he has been late on four of those days due to his bicycle breaking down, which suggests that this may be a recurring issue. Therefore, it would be reasonable to ask if Toma was late because his bicycle was broken again on the ninth day.

  Option 1 is still a general and open-ended question, but given that Toma has only been late once due to watching TV and the majority of his lateness has been related to his bicycle or not finding his belongings, it may not be the most relevant question to ask in this case.
- **GPT-3.5 with prompt engineering**: Given the pattern of Toma's lateness, it's likely that his bicycle breaking down is the reason he was late again on the ninth day. Therefore, Option 2: "Were you late because your bicycle was broken?" would be the better question to ask in this case. However, it's still possible that Toma was late for a different reason on the ninth day, so asking a more general question such as "Why were you late today?" might also be useful.
- **GPT-4**: Option 2: Were you late because your bicycle was broken?

  In this revised scenario, Toma was late due to his bicycle breaking on five out of the eight previous days. This makes it more likely that a broken bicycle could be the cause of Toma's tardiness again. Option 1 only addresses the reason for Toma being late on two out of the eight previous days (not finding something). Therefore, Option 2 is a better question to ask if you want to find out why Toma was late to school today.

Test A patient has a 0.8 probability of having Chamber-of-Commerce disease and a 0.2 probability of Elk's disease. (He surely has one or the other.) A tetherscopic examination yields a positive result in 90% of patients with Chamber-of-Commerce disease and in 20% of patients without it (including those with some other disease). An intraocular smear yields a positive result in 90% of patients with Elk's disease and in 10% of patients without it.

Q: Which test would you run on the patient?

- Option 1: A tetherscopic examination.
- Option 2: An intraocular smear.
- Option 3: Both are equally good.

- **GPT-3 (original)**: A: Option 1.
- **GPT-3 (replicated)**: Option 1: A tetherscopic examination.
- **GPT-3.5**: To determine which test to run on the patient, we need to consider which test would provide more information based on the probabilities given.

  Since the patient has a higher probability of having Chamber-of-Commerce disease (0.8) compared to Elk's disease (0.2), a positive result on a test for Chamber-of-Commerce disease is more likely to be correct.

From the information given, the tetherscopic examination has a higher probability of detecting Chamber-of-Commerce disease (90% chance of a positive result) compared to the intraocular smear for Elk's disease (90% chance of a positive result).

Therefore, Option 1: A tetherscopic examination would be the better test to run on the patient.

Option 3: Both are equally good is not a valid choice since the tetherscopic examination has a higher probability of detecting Chamber-of-Commerce disease, which has a higher probability of occurring in the patient.

- **GPT-3.5 with prompt engineering**: To determine which test to run, we need to consider which test is more likely to provide useful information about the patient's condition.

  If we run the tetherscopic examination, and the result is positive, there is a 90% chance the patient has Chamber-of-Commerce disease (true positive), and a 10% chance the patient does not have Chamber-of-Commerce disease (false positive). If the result is negative, there is a 20% chance the patient has Chamber-of-Commerce disease (false negative), and a 80% chance the patient does not have Chamber-of-Commerce disease (true negative).

  If we run the intraocular smear, and the result is positive, there is a 90% chance the patient has Elk's disease (true positive), and a 10% chance the patient does not have Elk's disease (false positive). If the result is negative, there is a 10% chance the patient has Elk's disease (false negative), and a 90% chance the patient does not have Elk's disease (true negative).

  Given the probabilities and potential outcomes, the better test to run would be the test with the higher true positive rate. In this case, the intraocular smear has a higher true positive rate (90%) than the tetherscopic examination (90% * 0.8 = 72%). Therefore, Option 2: "An intraocular smear" would be the better test to run.

- **GPT-4**: To determine which test is better to run on the patient, we should calculate the positive predictive value (PPV) of each test. The PPV is the probability that the patient has the disease, given a positive test result.

  Let C be the event that the patient has Chamber-of-Commerce disease, E be the event that the patient has Elk's disease, T+ be the event of a positive tetherscopic examination, and S+ be the event of a positive intraocular smear. We are given the following probabilities:

  $P(C) = 0.8$

  $P(E) = 0.2$

  $P(T+|C) = 0.9$

  $P(T+|E) = 0.2$

  $P(S+|E) = 0.9$

  $P(S+|C) = 0.1$

  First, let's calculate the PPV for the tetherscopic examination (PPV_T):

  $PPV_T = P(C|T+)$

  Using Bayes' theorem:

  $PPV_T = P(T+|C) * P(C)/P(T+)$

  To find P(T+), we can use the law of total probability:

  $P(T+) = P(T+|C) * P(C) + P(T+|E) * P(E)$

  Plugging in the given probabilities:

  $P(T+) = 0.9 * 0.8 + 0.2 * 0.2$

  $P(T+) = 0.72 + 0.04$

  $P(T+) = 0.76$

  Now we can find PPV_T:

  $PPV_T = P(T+|C) * P(C)/P(T+)$

  $PPV_T = (0.9 * 0.8)/0.76$

  $PPV_T \approx 0.9474$

  Now, let's calculate the PPV for the intraocular smear (PPV_S):

  $PPV_S = P(E|S+)$

  Using Bayes' theorem:

$PPV_S = P(S+|E) * P(E)/P(S+)$

To find P(S+), we can use the law of total probability:

$P(S+) = P(S+|E) * P(E) + P(S+|C) * P(C)$

Plugging in the given probabilities:

$P(S+) = 0.9 * 0.2 + 0.1 * 0.8$

$P(S+) = 0.18 + 0.08$

$P(S+) = 0.26$

Now we can find PPV_S:

$PPV_S = P(S+|E) * P(E)/P(S+)$

$PPV_S = (0.9 * 0.2)/0.26$

$PPV_S \approx 0.6923$

Comparing the two PPVs:

$PPV_T \approx 0.9474$

$PPV_S \approx 0.6923$

Option 1: A tetherscopic examination is the better test to run on the patient, as it has a higher positive predictive value.

Wason You are shown a set of four cards placed on a table, each of which has a number on one side and a colored patch on the other side. The visible faces of the cards show A, K, 4, and 7.

Q: Which cards must you turn over in order to test the truth of the proposition that if a card shows a vowel on one face, then its opposite face shows an even number?

- **GPT-3 (original)**: A: The cards that must be turned over are the A and the 7.

- **GPT-3 (replicated)**: The cards that must be turned over are the A and the 4.

- **GPT-3.5**: You need to turn over the A card and the 7 card.

  To test the proposition, you need to verify whether the opposite face of a vowel (A) shows an even number, and whether the opposite face of an odd number (7) shows a consonant.

  If the opposite face of the A card shows an odd number, then the proposition is false. If the opposite face of the 7 card shows a vowel, then the proposition is also false.

  Turning over the K and 4 cards would not provide any information about the truth of the proposition.

- **GPT-3.5 with prompt engineering**: You must turn over cards A and 7 to test the truth of the proposition.

- **GPT-4**: To test the truth of the proposition that if a card shows a vowel on one face, then its opposite face shows an even number, you need to check for cases that would violate this rule. In other words, you need to look for a card with a vowel on one face and an odd number on the other face.

  You should turn over the following cards:

  Card A: Since A is a vowel, you need to check if the other side has an even number, as per the proposition. If it does not have an even number, the proposition would be false.

  Card 7: Since 7 is an odd number, you need to check if the other side has a consonant. If it has a vowel, the proposition would be false, as it would contradict the rule that a vowel must have an even number on the other side.

  You don't need to turn over cards K and 4:

  Card K: K is a consonant, so the proposition doesn't make any claim about what should be on the other side. Therefore, it is not relevant to testing the proposition.

  Card 4: 4 is an even number, but the proposition only concerns cards with a vowel on one side. Turning this card over won't provide any information about the truth of the proposition.

  So, you must turn over cards A and 7 to test the truth of the proposition.

Deliberation: The Cognitive Reflection Test

CRT1 A bat and a ball cost 1.10 in total. The bat costs 1.00 more than the ball.
Q: How much does the ball cost?

- **GPT-3 (original)**:A: The ball costs $0.10.

- **GPT-3 (replicated)**: The ball costs $0.10.

- **GPT-3.5**: Let's assume the ball costs x dollars.

  Since the bat costs 1 dollar more than the ball, it would cost x+1 dollars.

  Together, the bat and ball cost 1.10 dollars.

  So we can set up the equation:

  x + (x+1) = 1.10

  Simplifying the equation, we get:

  2x + 1 = 1.10

  Subtracting 1 from both sides, we get:

  2x = 0.10

  Dividing by 2, we get:

  x = 0.05

  Therefore, the ball costs 0.05 dollars.

- **GPT-3.5 with prompt engineering**: Let's call the cost of the ball "x". Then, according to the problem, we know that the cost of the bat is $1.00 more than the ball, so it would be "x + $1.00".

  We also know that the total cost of both items is $1.10. So we can set up the following equation:

  x + (x + $1.00) = $1.10

  Simplifying the equation, we get:

  2x + $1.00 = $1.10

  Subtracting $1.00 from both sides, we get:

  2x = $0.10

  Dividing both sides by 2, we get:

  x = $0.05

  Therefore, the ball costs $0.05.

- **GPT-4**: Let the cost of the ball be x. According to the problem, the bat costs x + 1.00. The total cost of the bat and the ball is 1.10. We can set up the following equation:

  x + (x + 1.00) = 1.10

  Combining the terms:

  2x + 1.00 = 1.10

  Subtract 1.00 from both sides:

  2x = 0.10

  Now, divide both sides by 2:

  x = 0.05

  The ball costs 0.05 (5 cents).

CRT2 If it takes 5 machines 5 minutes to make 5 widgets, how long would it take 100 machines to make 100 widgets?

- **GPT-3 (original)**: A: 100 machines would make 100 widgets in 100 minutes.

- **GPT-3 (replicated)**: It would take 100 machines 100 minutes to make 100 widgets.

- **GPT-3.5**: Assuming all the machines are equally efficient, we can set up a proportion to solve for the time it would take 100 machines to make 100 widgets:

  5 machines make 5 widgets in 5 minutes, so 1 machine makes 1 widget in 5 minutes.

  Therefore, 100 machines make 100 widgets in (100 widgets)/(100 machines x 1 widget/machine x 5 minutes/widget) = 20 minutes.

  So it would take 100 machines 20 minutes to make 100 widgets.

- **GPT-3.5 with prompt engineering**: If it takes 5 machines 5 minutes to make 5 widgets, then each machine takes 5 minutes to make one widget.

  Therefore, to make 100 widgets, we need 100 machines, and each machine takes 5 minutes to make one widget.

  So, it will take 100 machines 5 minutes to make 100 widgets.

- **GPT-4**: We know that 5 machines can make 5 widgets in 5 minutes. This means that in those 5 minutes, each machine makes 1 widget (5 widgets / 5 machines = 1 widget per machine).

  Now, if we have 100 machines, each machine can still make 1 widget in 5 minutes. Therefore, 100 machines can make 100 widgets in the same 5 minutes.

CRT3 In a lake, there is a patch of lily pads. Every day, the patch doubles in size.
Q: If it takes 48 days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?

- **GPT-3 (original)**: A: It would take 24 days for the patch to cover half of the lake.

- **GPT-3 (replicated)**: It would take 24 days for the patch to cover half of the lake.

- **GPT-3.5**: Since the patch doubles in size every day, we can work backwards to figure out how many days it takes for the patch to cover half the lake.

  On the day before the lake is fully covered, the patch must have covered half the lake. Let's call this day 47.

  On day 46, the patch covered half the area of the lake on day 47.

  On day 45, the patch covered half the area of the lake on day 46.

  We can continue this pattern until we reach day 1, when the patch covered half the area of the lake on day 2.

  So, it takes 47 days for the patch to cover half the lake.

- **GPT-3.5 with prompt engineering**: Since the patch doubles in size every day, we can work backwards to find how many days it would take for the patch to cover half the lake. If it takes 48 days to cover the entire lake, on the 47th day the patch would have covered half the lake.

  Why the 47th day? Because the patch doubles in size every day, so on the day before it covers half the lake, it must have covered half of the half-lake, i.e., one-quarter of the full lake. Working backwards, on the 46th day it would have covered one-eighth of the full lake, on the 45th day one-sixteenth of the full lake, and so on.

  Therefore, it would take 47 days for the patch to cover half of the lake.

- **GPT-4**: Since the patch of lily pads doubles in size every day, the day before it covers the entire lake, it would cover half of the lake.

  So, it would take 48 days - 1 day = 47 days for the patch to cover half of the lake.

Causal reasoning: Blickets, interventions, and counterfactuals

Blickets Blickets are objects that turn on a machine. I have two objects, A and B, and a machine. My goal is to find out if any or both of these objects are a blicket. When I put object A on the machine, the machine turned on. When I put object A and object B on the machine, the machine turned on. When I put object B on the machine, the machine did not turn on.

Q: Do you think object A is a blicket?
Q: Do you think object B is a blicket?
Q: Why is object B not a blicket?

- **GPT-3 (original)**: A: Yes, I think object A is a blicket.

  A: No, I think object B is not a blicket.

  A: I think object B is not a blicket because it did not turn on the machine when it was placed on the ma- chine by itself.

- **GPT-3 (replicated)**: From the information given, it seems that object A is a blicket and object B is not. It's possible that object B is not a blicket because it needs to be used in conjunction with object A in order to work.

- **GPT-3.5**: Based on the information given:

  Object A turned on the machine when it was put on it, so it is a blicket.

  Object B did not turn on the machine when it was put on it alone, so it is not a blicket.

  When object A and object B were put together on the machine, the machine turned on, which suggests that object A is enough to turn on the machine, and object B is not necessary. Therefore, the answers to the questions are:

  Yes, object A is a blicket.

  No, object B is not a blicket.

  Object B is not a blicket because it did not turn on the machine when it was put on it alone. It seems that it requires the presence of object A to turn on the machine.

- **GPT-3.5 with prompt engineering**: Based on the given information, it seems likely that object A is a blicket, as it caused the machine to turn on by itself and when combined with object B. Object B, on the other hand, is likely not a blicket, as it did not cause the machine to turn on by itself, but only when combined with object A.

  Object B could fail to be a blicket for various reasons, such as not having the required weight, shape, or material properties needed to activate the machine, or having some kind of mechanism that prevents it from turning on the machine by itself.

- **GPT-4**: A: Yes, object A is a blicket. When you put object A on the machine, it turned on, which indicates that object A is a blicket.

  A: No, object B is not a blicket. When you put object B on the machine alone, the machine did not turn on, which indicates that object B is not a blicket.

  A: Object B is not a blicket because it does not turn on the machine when placed on it by itself. In the scenario where both object A and object B were on the machine and the machine turned on, it is likely due to the presence of object A, which is a blicket.

Intervene I am allergic to flowers. When my friend gave me red flowers, I did not sneeze. When my friend gave me yellow flowers, I did not sneeze. When my friend gave me blue flowers, I sneezed.

Q: Imagine a bouquet containing red, yellow, and blue flowers. Which flowers would need to be removed such that I would not show an allergic reaction?

- **GPT-3 (original)**: A: The blue flowers would need to be removed.

- **GPT-3 (replicated)**: The blue flowers would need to be removed.

- **GPT-3.5**: Since you did not show an allergic reaction to both red and yellow flowers, we can assume that you are not allergic to them. However, you did sneeze when you received blue flowers, indicating that you are allergic to them. Therefore, the blue flowers would need to be removed from the bouquet for you to not show an allergic reaction.

- **GPT-3.5 with prompt engineering**: Based on the information given, it seems that you are only allergic to blue flowers. Therefore, the blue flowers would need to be removed from the bouquet in order for you to not show an allergic reaction.

- **GPT-4**: Based on the information provided, you would need to remove the blue flowers from the bouquet to prevent an allergic reaction, as they are the only flowers that made you sneeze. The red and yellow flowers did not cause an allergic reaction, so they can remain in the bouquet.

Mature Imagine there are four different pills: pill A, pill B, pill C, and pill D. Pill A can kill people. Pill B can kill people. Pill C cannot kill people. Pill D cannot kill people.

Q: A man took pill A and pill B and he died. If he had not taken pill A, could he still have died, yes or no?
Q: A man took pill B and pill C and he died. If he had not taken pill B, could he still have died, yes or no?
Q: A man took pill A and pill D and he died. If he had not taken pill D, could he still have died, yes or no?
Q: A man took pill C and pill D and he did not die. If he had not taken pill D, could he have died, yes or no?

- **GPT-3 (original)**: A: Yes

  A: No

  A: Yes

  A: No

- **GPT-3 (replicated)**: 1. Yes

  2. Yes

  3. Yes

  4. No

- **GPT-3.5**: If the man had not taken pill A, he could still have died from the effects of pill B.

  If the man had not taken pill B, he may not have died from pill C, but it is not clear from the information given whether he could still have died from other causes or from taking pill A or D.

  If the man had not taken pill D, he could still have died from the effects of pill A.

  It is not clear from the information given whether the man could have died from taking pill C, with or without taking pill D.

- **GPT-3.5 with prompt engineering**: A: Yes

  A: Yes

  A: Yes

  A: No

  Sure, here are the explanations for each question:

  Q: A man took pill A and pill B and he died. If he had not taken pill A, could he still have died, yes or no?

  A: Yes. Since pill B can also kill people, it is possible that pill B alone was enough to cause the man's death.

  Q: A man took pill B and pill C and he died. If he had not taken pill B, could he still have died, yes or no?

  A: Yes. Pill C cannot kill people, but there may have been other factors that caused the man's death.

  Q: A man took pill A and pill D and he died. If he had not taken pill D, could he still have died, yes or no?

  A: Yes. Since pill A can kill people, it is possible that pill A alone was enough to cause the man's death.

  Q: A man took pill C and pill D and he did not die. If he had not taken pill D, could he have died, yes or no?

  A: No. Pill C cannot kill people, so even if the man had not taken pill D, he still would not have died.

- **GPT-4**: Q: A man took pill A and pill B and he died. If he had not taken pill A, could he still have died, yes or no?

  A: Yes, he could still have died because he also took pill B, which can kill people.

  Q: A man took pill B and pill C and he died. If he had not taken pill B, could he still have died, yes or no?

  A: No, he would not have died because pill C cannot kill people, and it would be the only pill he took.

  Q: A man took pill A and pill D and he died. If he had not taken pill D, could he still have died, yes or no?

  A: Yes, he could still have died because he also took pill A, which can kill people.

  Q: A man took pill C and pill D and he did not die. If he had not taken pill D, could he have died, yes or no?

  A: No, he would not have died because pill C cannot kill people, and it would be the only pill he took.

Adversarial vignettes

Black Cab A cab was involved in a hit-and-run accident at night. Two cab companies, the Green and the Blue, operate in the city. You are given the following data: 85% of the cabs in the city are Green and 15% are Blue. A witness identified the cab as Blue. The court tested the reliability of the witness under the circumstances that existed on the night of the accident and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

Q: What is the probability that the cab involved in the accident was black?

- **GPT-3 (original)**: A: The probability that the cab involved in the acci- dent was black is 0.2.
- **GPT-3 (replicated)**: There is no information given about the probability that the cab involved in the accident was black, so we cannot calculate this.
- **GPT-3.5**: The problem does not provide any information about the proportion of black cabs in the city, so it is not possible to calculate the probability that the cab involved in the accident was black.
- **GPT-3.5 with prompt engineering**: We don't have any information about the proportion of black cabs in the city, so we can't calculate the probability that the cab involved in the accident was black. However, we can calculate the probability that the cab was actually Blue, given that the witness identified it as Blue.

  Let B be the event that the cab involved in the accident was Blue, and W be the event that the witness identified the cab as Blue. We want to calculate P(B — W).

  Using Bayes' theorem, we have:

  P(B | W) = P(W | B) * P(B) / P(W)

  We know that P(B) = 0.15, and we can calculate P(W) using the law of total probability:

  P(W) = P(W | B) * P(B) + P(W | not B) * P(not B) = 0.8 * 0.15 + 0.2 * 0.85 = 0.29

  We also know that the witness correctly identified each color 80% of the time, so P(W — B) = 0.8 and P(W — not B) = 0.2.

  Substituting all these values, we get:

  P(B | W) = 0.8 * 0.15 / 0.29 $\approx$ 0.41

  So the probability that the cab involved in the accident was actually Blue, given that the witness identified it as Blue, is about 0.41.

- **GPT-4**: The problem statement does not provide any information about black cabs operating in the city. It only mentions Green and Blue cab companies. Based on the information provided, it is impossible to calculate the probability of the cab involved in the accident being black.

Reverse Wason You are shown a set of four cards placed on a table, each of which has a number on one side and a colored patch on the other side. The visible faces of the cards show 4, 7, A, and K.

Q: Which cards must you turn over in order to test the truth of the proposition that if a card shows a vowel on one face, then its opposite face shows an even number?

- **GPT-3 (original)**: A: The A and the K.
- **GPT-3 (replicated)**: The cards that must be turned over are the A and the K.
- **GPT-3.5**: You would need to turn over the A and the 7 cards.
- **GPT-3.5 with prompt engineering**: You need to turn over the card with the letter "A" and the card with the number "7".

  To test the truth of the proposition, we need to see if there is a vowel on one face and an even number on the other face. The cards with the numbers 4 and K do not need to be turned over because they do not have vowels on their faces. However, the card with the letter "A" needs to be turned over to check if there is an even number on the other face. Similarly, the card with the number "7" needs to be turned over to check if there is a vowel on the other face.

- **GPT-4**: To test the truth of the proposition that if a card shows a vowel on one face, then its opposite face shows an even number, you need to check for cases that would violate this rule. In other words, you need to look for a card with a vowel on one face and an odd number on the other face.

  You should turn over the following cards:

Card A: Since A is a vowel, you need to check if the other side has an even number, as per the proposition. If it does not have an even number, the proposition would be false.

Card 7: Since 7 is an odd number, you need to check if the other side has a consonant. If it has a vowel, the proposition would be false, as it would contradict the rule that a vowel must have an even number on the other side.

You don't need to turn over cards 4 and K:

Card 4: 4 is an even number, but the proposition only concerns cards with a vowel on one side. Turning this card over won't provide any information about the truth of the proposition.

Card K: K is a consonant, so the proposition doesn't make any claim about what should be on the other side. Therefore, it is not relevant to testing the proposition.

So, you must turn over cards A and 7 to test the truth of the proposition.

Wrong CRT A bat and a ball cost 1.10 in total. The bat costs 1.00 more than the bat. Q: How much does the ball cost?

- **GPT-3 (original)**: A: The ball costs $0.10.

- **GPT-3 (replicated)**: A: The ball costs $0.10.

- **GPT-3.5**: Let's assume that the ball costs x. According to the problem, we know that the bat costs 1.00 more than the ball, which means that the bat costs x+1.

  We also know that the total cost of the bat and ball is 1.10.

  Therefore, we can set up the following equation:

  x + (x+1) = 1.10

  Simplifying:

  2x + 1 = 1.10

  Subtracting 1 from both sides:

  2x = 0.10

  Dividing both sides by 2:

  x = 0.05

  Therefore, the ball costs 0.05 dollars.

- **GPT-3.5 with prompt engineering**: Let's call the cost of the ball "b" and the cost of the bat "1.00 + b" (since we know the bat costs $1 more than the ball). We also know that the total cost of the bat and ball is $1.10. So we can set up the equation:

  b + (1.00 + b) = 1.10

  Simplifying this equation, we get:

  2b + 1.00 = 1.10

  2b = 0.10

  b = 0.05

  Therefore, the ball costs $0.05.

- **GPT-4**: There seems to be a mistake in your question. It should say "The bat costs 1.00 more than the ball" instead of "The bat costs 1.00 more than the bat." Here is the corrected question and its answer:

  A bat and a ball cost 1.10 in total. The bat costs 1.00 more than the ball. How much does the ball cost?

  Let the cost of the ball be x. According to the problem, the bat costs x + 1.00. The total cost of the bat and the ball is 1.10. We can set up the following equation:

  x + (x + 1.00) = 1.10

  Combining the terms:

  2x + 1.00 = 1.10

  Subtract 1.00 from both sides:

  2x = 0.10

Now, divide both sides by 2:

x = 0.05

The ball costs 0.05 (5 cents).

Immature Blicket Imagine a box that makes a sound if certain objects are placed on it. When I put a red object on it, the box made a sound. When I put a green object on it, the box made a sound. When I put a blue object on it, the box did not make a sound. When I put a yellow object on it, the box did not make a sound.

Q: I put a green and a yellow object on the box and it made a sound. Would it still have made a sound if I had not put the yellow object on it?

Q: Would the box make a sound if I only put a green object on it?

Q: But if there are two objects on the box, a green and a yellow one, and I remove the yellow one, then there would only be a green object on it. Why does that not turn on the machine?

- **GPT-3 (original)**: A: No, the box would not have made a sound if you had not put the yellow object on it.

  A: Yes, the box would make a sound if you only put a green object on it.

  A: The box would not make a sound if there is only a green object on it.

- **GPT-3 (replicated)**: Yes, the box would still have made a sound if the yellow object had not been placed on it.

  Yes, the box would make a sound if you only put a green object on it. The box is likely programmed to only make a sound when a certain object is placed on it. In this case, the box is programmed to make a sound when a red, green, or blue object is placed on it.

  It does not matter how many objects are on the box, as long as the correct object is present, the box will make a sound.

- **GPT-3.5**: Based on the information given, we can infer the following:

  The box makes a sound if either a red object or a green object is placed on it.

  The box does not make a sound if either a blue object or a yellow object is placed on it.

  When a green and a yellow object are placed on the box together, it makes a sound.

  Q1: If you had only put a green object on the box, it would have made a sound because the box makes a sound when a green object is placed on it.

  Q2: Yes, the box would make a sound if you only put a green object on it, based on the first observation that the box makes a sound if a green object is placed on it.

  Q3: This is because the box is designed to make a sound only if certain objects are placed on it together. In this case, the box makes a sound only if both a green object and a yellow object are placed on it together. Removing one of the objects will prevent the box from making a sound. So, when you remove the yellow object, the condition for the box to make a sound is no longer satisfied, even though a green object is still on the box.

- **GPT-3.5 with prompt engineering**: 1. No, if the box made a sound when a green and yellow object were placed on it, and it did not make a sound when a yellow object was placed on it alone, then the yellow object is necessary to trigger the sound.

  2. Yes, according to the given information, the box made a sound when a green object was placed on it.

  3. It's possible that the box has a specific combination of objects that triggers the sound, rather than simply the presence of a single object. It's also possible that the yellow object is necessary to complete a circuit or mechanism inside the box that allows it to produce the sound when the green object is present.

- **GPT-4**: Q: I put a green and a yellow object on the box and it made a sound. Would it still have made a sound if I had not put the yellow object on it?

  A: Yes, it would still have made a sound. Based on the information provided, the box made a sound when a green object was placed on it.

  Q: Would the box make a sound if I only put a green object on it?

  A: Yes, the box would make a sound if you only put a green object on it. In the given examples, when you put a green object on the box, it made a sound.

  Q: But if there are two objects on the box, a green and a yellow one, and I remove the yellow one, then there would only be a green object on it. Why does that not turn on the machine?

  A: Based on the information provided, the box should still make a sound when only a green object is on it. If you remove the yellow object and leave the green one, the box should still make a sound.

Adversarial vignettes - Extensions

Quadruple Lily In a lake, there is a patch of lily pads. Every day, the patch quadruples in size. Q: If it takes 48 days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?

- **GPT-3**: It would take 24 days for the patch to cover half of the lake.
- **GPT-3.5**: If the patch quadruples in size every day, we can think of this as doubling in size twice. So after one day, the patch will be four times its original size. After two days, it will be sixteen times its original size, and so on. Thus, if it takes 48 days for the patch to cover the entire lake, it must take 47 days for the patch to cover half the lake (since on the 48th day it will double in size one last time to cover the whole lake).
- **GPT-3.5 with prompt engineering**: If the patch of lily pads quadruples in size every day, we can determine the number of times it needs to quadruple to cover half of the lake.

  Since it takes 48 days for the patch to cover the entire lake, we can work backwards to find out when it would cover half of the lake.

  Starting with day 47, the patch would be 1/4th the size of the lake (since it quadruples each day). On day 46, it would be 1/16th the size, on day 45 it would be 1/64th, and so on.

  To find the day when the patch covers half of the lake, we need to determine when the patch is 1/2 or 50% of the lake's size. Let's calculate:

  1/64 * 1/64 * 1/64 * ... * 1/64 = 1/2

  The number of times we need to multiply 1/64 by itself to get 1/2 is log base 1/64 of 1/2. Using logarithms, we can find that:

  log base 1/64 of 1/2 = log(1/2) / log(1/64) ≈ 6.0206

  Therefore, it would take approximately 6.0206 days for the patch to cover half of the lake.
- **GPT-4**: If it takes 48 days for the patch of lily pads to cover the entire lake, and the patch quadruples in size every day, we can work backwards to find out how long it would take for the patch to cover half of the lake.

  Since the patch quadruples in size daily, we can divide the size of the patch by 4 to find out the size it was one day prior. On day 48, the patch covers the entire lake (let's say this is 1 or 100%). Therefore, on day 47, the patch would cover 1/4 of the lake (or 25%), because 1/4 * 4 = 1.

  Now, we need to find when the patch would cover half of the lake (or 50%). Since 25% is half of 50%, we can move one more day back. On day 46, the patch would be half the size it was on day 47 (25%). Half of 25% is 12.5%.

  So, it would take 46 days for the patch to cover half of the lake.

Unproductive Machines Q: If it takes 5 machines 5 minutes to make 0 widgets, how long would it take 100 machines to make 100 widgets?

- **GPT-3** : A: If it takes 5 machines 5 minutes to make 0 widgets, it would take 100 machines 100 minutes to make 100 widgets.
- **GPT-3.5**: Assuming that the relationship between the number of machines, time and number of widgets is linear, we can set up a proportion to solve for the time it would take 100 machines to make 100 widgets:

  5 machines can make 0 widgets in 5 minutes, so 5 machines can make 1 widget in 25 minutes (5 x 5). Therefore, 5 machines can make 100 widgets in 2500 minutes (25 x 100).

  To find out how long it would take 100 machines to make 100 widgets, we can set up the following proportion:

  5 machines : 2500 minutes = 100 machines : x minutes

  where x is the time it would take 100 machines to make 100 widgets.

  Simplifying the proportion:

  5 * x = 100 * 2500 x = 50,000 minutes

  Therefore, it would take 100 machines 50,000 minutes, or approximately 34 days and 18 hours, to make 100 widgets.
- **GPT-3.5 with prompt engineering**: If it takes 5 machines 5 minutes to make 0 widgets, we can use this information to determine the time it would take for a different number of machines to produce a specific number of widgets.

  In this case, we have 100 machines and we want to know how long it would take for them to make 100 widgets. Let's set up a proportion to solve the problem:

5 machines / 5 minutes = 100 machines / x minutes

To find the value of x, we can cross-multiply and solve for it:

5 * x = 5 * 100

x = 500

Therefore, it would take 100 machines approximately 500 minutes to make 100 widgets.

- **GPT-4**: There seems to be a mistake in the question. If 5 machines take 5 minutes to make 0 widgets, it means that the machines are not producing any widgets at all. In this case, no matter how many machines you have or how much time you give them, they will still produce 0 widgets. Therefore, it is not possible to determine the time it would take for 100 machines to make 100 widgets based on the given information.

Half Day In a lake, there is a patch of lily pads. Every half day, the patch doubles in size.

Q: If it takes 48 days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?

- **GPT-3**: It would take 24 days for the patch to cover half of the lake.

- **GPT-3.5**: Since the patch doubles in size every half day, after 1 day it would have covered four times the original area. Therefore, after 2 days, it would have covered 16 times the original area. We can see a pattern emerging here: after n days, the patch will have covered 2^{4n} times the original area.

  If it takes 48 days for the patch to cover the entire lake, it would take 47 half-days for the patch to cover half of the lake, because after 47 half-days, the patch will have covered $2^{4\times47} = 2,147,483,648$ times the original area, which is half of the area of the full lake. Therefore, the answer is 23.5 days.

- **GPT-3.5 with prompt engineering**: If the patch of lily pads doubles in size every half day, we can determine the number of times it needs to double to cover half of the lake.

  Since it takes 48 days for the patch to cover the entire lake, we can work backwards to find out when it would cover half of the lake.

  Starting with day 47.5, the patch would be 1/2 the size of the lake (since it doubles every half day). On day 47, it would be 1/4th the size, on day 46.5 it would be 1/8th the size, and so on.

  To find the day when the patch covers half of the lake, we need to determine when the patch is 1/2 or 50% of the lake's size. Let's calculate:

  $(1/2)^2 * (1/2)^2 * (1/2)^2 * ... = 1/2$

  The number of times we need to multiply $(1/2)^2$ by itself to get 1/2 is log base (1/2) of 1/2. Using logarithms, we can find that:

  log base (1/2) of 1/2 = 1

  Therefore, it would take approximately 1 half day for the patch to cover half of the lake.

- **GPT-4**:If the patch of lily pads doubles in size every half day and it takes 48 days for the patch to cover the entire lake, we can determine how long it would take for the patch to cover half of the lake.

  Since the patch doubles in size every half day, there are 2 growth cycles per day. Therefore, there are 48 days * 2 = 96 half-day growth cycles in total.

  To find out when the patch covers half of the lake, we need to go back by one growth cycle. So, on the 95th half-day growth cycle, the patch would cover half of the lake.

  Now, we need to convert this back into days: 95 half-day growth cycles / 2 = 47.5 days.

  So, it would take 47.5 days for the patch to cover half of the lake.

Appendix II: Deliberation plots

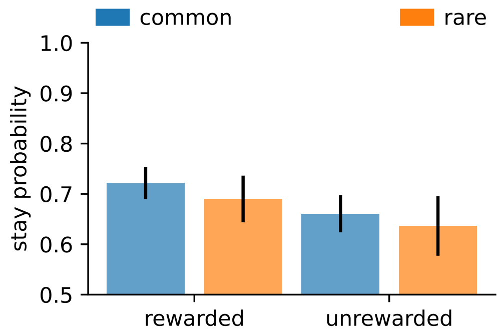

Figure 7: Deliberation two step task results from replication on GPT-3 on 25 runs with 20 trials each. $p>0.2$

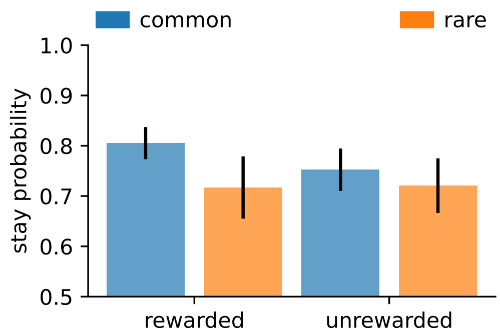

Figure 8: Deliberation two step task results from GPT-3 with prompt engineering on 25 runs with 20 trials each. $p>0.2$

Appendix III: Causal reasoning GPT-3.5 and prompt engineering responses

Causal-chain

GPT3.5

- **do(B=1)**: Based on the previous research, it is likely that substance B will cause the production of substance A, and substance A will likely cause the production of substance C. Therefore, if substance B is added to the new casks, it is reasonable to expect that substance C will also be present in those casks. Out of the 20 new casks, we can estimate that on average, all 20 casks will contain substance C.

- **do(B=0)**: Based on the given information, if substance B is removed, substance A is not likely to be produced, which means substance C is also not likely to be produced. Therefore, on average, none of the new casks will contain substance C. The answer is 0.

- **B=1**: Based on the previous research, it is likely that substance A will be present in the casks where substance B is present. And since substance A likely causes the production of substance C, it is also likely that substance C will be present in these casks. Out of the 20 new casks, we can estimate that on average, around 15 casks will contain substance C. This is based on the fact that in the previous observations, 15 out of 20 casks contained

- **B=0**: Based on the previous information, if substance B is absent, substance A is also likely to be absent, which means substance C is unlikely to be produced. Therefore, on average, very few or none of the new casks will contain substance C. As an estimate, we can say that 1 or 2 out of the 20 new casks may contain substance C on average.

Prompt Engineering

- **do(B=1)**: Based on the previous information, it is likely that substance B causes the production of substance A, and substance A causes the production of substance C. Therefore, if substance B is added to the new casks, it is likely that substance C will also be present. Looking at the previous observations, we can see that in all cases where substance B and substance A were present, substance C was also present. Therefore, we can expect that all 20 new casks will contain substance C on average.
- **do(B=0)**: Based on the information provided, it is likely that substance A causes the production of substance C. Therefore, if substance B is removed from the new casks, it is expected that substance A will not be present, and consequently, substance C will also not be present. Therefore, the expected number of new casks that will contain substance C is zero.
- **B=1**: Based on the previous information, it is likely that substance A causes the production of substance C. Therefore, if substance B is present in the new casks, it is likely that substance A and subsequently substance C will also be present. Looking at the previous casks, all the casks that had substance B also had substance A, and most of the casks that had substance A also had substance C. Therefore, it is reasonable to assume that most of the new casks with substance B will
- **B=0**: Based on the information provided, it is likely that substance A is not present in the new casks since substance B, which is necessary for its production, is absent. Therefore, substance C, which is likely produced by substance A, is also unlikely to be present in the new casks. Therefore, on average, none of the new casks will contain substance C.

Common-cause

GPT3.5

- **do(B=1)**: Based on the previous observations and the additional information, it is likely that substance A is present in the casks that contain substance B. And since substance A likely causes the production of substance C, it is reasonable to assume that most of the new casks that contain substance B will also contain substance C. Therefore, on average, it is expected that around 18 of the new casks (out of 20) will contain substance C.
- **do(B=0)**: Based on the previous observations and the additional information, it can be inferred that substance A is necessary for the production of both substance B and substance C. Therefore, if substance B is removed from the new casks, it is likely that substance C will also be absent.Thus, on average, none of the new casks will contain substance C.
- **B=1**: Based on the previous observations and the additional information, it is likely that all 20 of the new casks containing substance B will also contain substance C. Therefore, the average number of new casks containing substance C is 20.
- **B=0**: Based on the previous observations, substance A is present in 7 out of the 20 casks (Casks 2, 5, 8, 9, 10, 13, and 15). Since substance A likely causes the production of substance C, we can expect that these 7 casks will also contain substance C. Therefore, on average, we can expect that 7 out of the 20 new casks will contain substance C.

Prompt Engineering

- **do(B=1)**: Based on the previous information that substance A likely causes the production of both substance B and substance C, and assuming that the addition of substance B in the new casks will trigger the production of substance C, it is reasonable to expect that all 20 new casks will contain substance C. Therefore, the average number of new casks containing substance C is 20.
- **do(B=0)**: Based on the previous observations, we know that substance A likely causes the production of substance C. Therefore, if we remove substance B, we can still expect to find substance C in the casks where substance A is present. Out of the 20 new casks, we know that substance A was present in 10 of the previous casks. Therefore, on average, we can expect to find substance C in approximately 10 of the new casks.
- **B=1**: Based on the previous information, if substance B is present, it is likely that substance A is also present, which in turn is likely to cause the production of substance C. Therefore, on average, I would expect around 15 of the 20 new casks to contain substance C.
- **B=0**: Based on the previous information, if substance B is absent, it is still likely that substance C will be present if substance A is present. Therefore, the number of new casks that will contain substance C on average can be estimated by looking at the proportion of casks in the previous observations that had substance A and substance C present, regardless of the presence of substance B. Out of the 20 previous casks, 10 had substance A and C present. Therefore, on average, we can expect 10/20 or 0.5 of the new casks to contain substance C. So, the answer is 0.5 or 50

Appendix IV: Information Search logistic Regression Plots

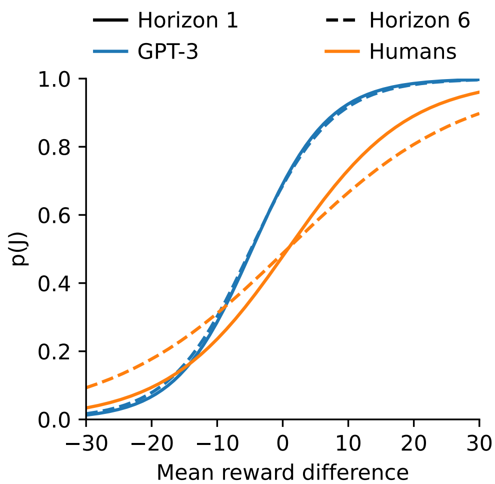

Figure 9: GPT-3 replication under equal.

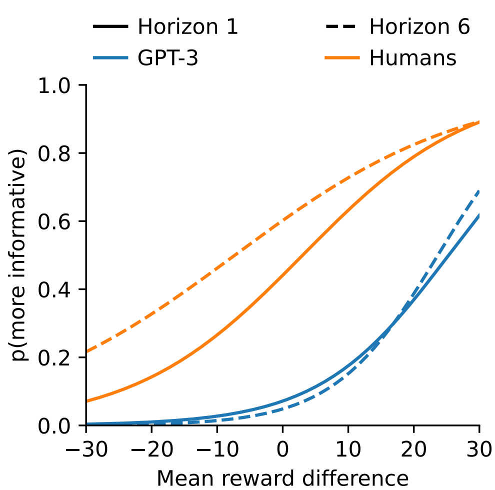

Figure 10: GPT-3 replication under unequal.

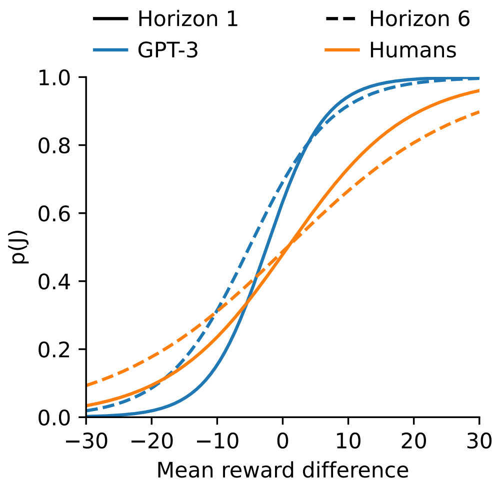

Figure 11: GPT-3 with prompt engineering under equal.

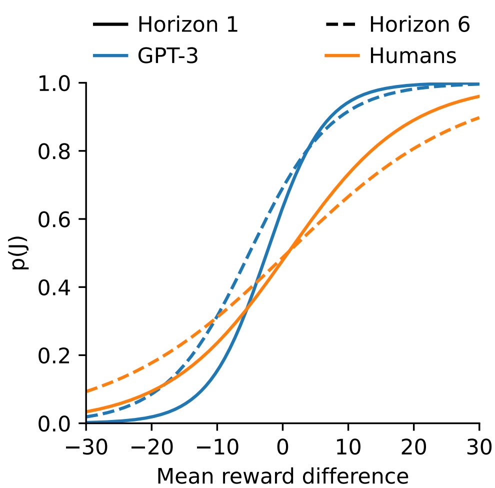

Figure 12: GPT-3 with prompt engineering replication under unequal.