Question-Analysis Prompting Improves LLM Performance in Reasoning Tasks

Dharunish Yugeswardeenoo **Kevin Zhu** **Sean O'Brien**
Algoverse AI Research
dharyugi@gmail.com, kevin@algoverseacademy.com

Abstract

Although LLMs have the potential to transform many fields, they still underperform humans in reasoning tasks. Existing methods induce the model to produce step-by-step calculations, but this research explores the question: Does making the LLM analyze the question improve its performance? We propose a novel prompting strategy called Question Analysis Prompting (QAP), in which the model is prompted to explain the question in *n* words before solving. The value of *n* influences the length of response generated by the model. QAP is evaluated on GPT-3.5 Turbo and GPT-4 Turbo on arithmetic datasets GSM8K, AQuA, and SAT and commonsense dataset StrategyQA. QAP is compared with other state-of-the-art prompts including chain-of-thought (CoT), Plan and Solve Prompting (PS+) and Take A Deep Breath (TADB). QAP outperforms all state-of-the-art prompts on AQuA and SAT datasets on both GPT-3.5 and GPT-4. QAP consistently ranks among the top-2 prompts on 75% of the tests. A key factor of QAP performance can be attributed to response length, where detailed responses are beneficial when answering harder questions, but can negatively affect easy questions.

1 Introduction

Large language models (LLMs) have recently shown rapid improvement across a host of standard natural language processing (NLP) tasks, including arithmetic, commonsense and symbolic reasoning. (Brown et al., 2020) Although these models show improved ability to understand and generate text (OpenAI, 2023), their performance can still be further improved. One solution is to encourage the model to think step-by-step. Using chain-of-thought prompting (Wei et al., 2022), LLMs are given Q&A exemplars which are designed to elicit a structured step-by-step response from the model. Many newly developed strategies meant to improve LLM performance have been focused on sophisticating the model's step-by-step calculation (Gu et al., 2023). Despite SoTA prompts' remarkable success across various tasks, their accuracies can still be further improved. In this work, we explore ways to improve the model reasoning not only in the answer steps, but also how the model interprets the question itself. By making the model to explicitly interpret the question, we maximize its understanding of the question and minimize missed key information. This paper introduces Question-Analysis Prompting (QAP), a simple zero-shot prompting strategy that induces the model to first explain the question before solving. We include a configurable parameter within the prompt to examine how different word counts affect the quality of a model's response.

2 Prompt Design

The key principle behind QAP is that the model should reiterate the problem in its own words before solving. The benefit is that the model will be able to first think about what task it is trying to solve before it pursues the answer. Another principle is that we should be able to control how much the model explains so that we can adapt the prompt to different model sizes and problem complexities. The specific prompt used is as follows:

"Explain this problem to me in at least n words. Then solve for the answer."

In this work, we experiment with *n* = 25, 50, 100, 150, 200. The versions of these prompts are named QAPn. Although the model is not constrained to generating fewer than *n* tokens in its summary, we find that the number of tokens in the response correlates strongly with the choice of n. The correlation between *n* and median word count is 0.98. We show specific examples of the impacts of n in Figure 4.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 402–413
August 11-16, 2024 ©2024 Association for Computational Linguistics

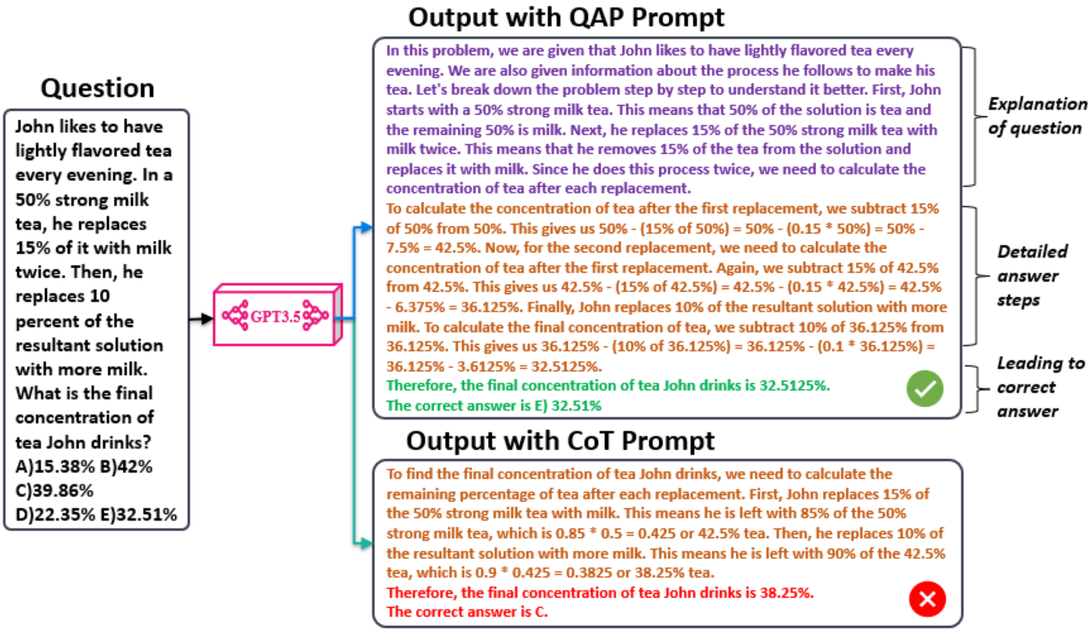

Figure 1: Example of QAP prompting - shows how the prompt triggers explanation of the question followed by an approach to solve the problem, detailed steps, finally leading to correct answer

3 Prompt Impact

In Figure 1, we highlight the structure of a standard QAP output. First, the model breaks down the question in its own words and provides detailed analysis on each event. Many of the steps highlighted in the explanation were shown in the calculation section. Compared to the CoT output, QAP encourages more sophistication in its response and thus reaches the correct answer.

4 Experimental Setup

4.1 Benchmarks

We evaluate the effectiveness of QAP on three arithmetic reasoning datasets. These include grade-school math questions from **GSM8K** (Cobbe et al., 2021), algebraic word problems from **AQuA** (Ling et al., 2017), and SAT math problems from **AGIEval** (Zhong et al., 2023). For commonsense reasoning, we evaluate on open-domain questions that require implicit reasoning, from **StrategyQA** (Geva et al., 2021). We evaluate on the test sets of all benchmarks.

4.2 Models

We specifically choose our models to observe the prompts' impacts across differences in model size. The smaller model is **GPT-3.5 Turbo** with version `gpt-3.5-turbo-0613`. Our larger model is **GPT-4 Turbo** with version `gpt-4-1106-preview` (OpenAI, 2023). For both of the models we used the OpenAI API [1] for running our experiments. The temperature and Top-K sampling was set to 0 to avoid randomness and keep consistency in the model's responses.

4.3 Prompts

For all datasets and models, we experiment with different variations of QAP. We utilize **QAP25**, **QAP50**, **QAP100**, **QAP150**, and **QAP200**. We compare the performance of QAP with the baseline (no prompt). Additionally we compare QAP with two different zero-shot prompts: **TADB** - "Take a deep breath and work on this problem step-by-step" (Yang et al., 2023) and **PS+** (Plan and Solve Plus) (Wang et al., 2023). Finally we also compare QAP with 8-shot chain-of-thought prompting.

4.4 Results

The results for GPT-3.5 Turbo and GPT-4 Turbo are shown in Table 1 and Table 2 respectively. General word counts are shown in Figure 7.

Arithmetic Reasoning: On GPT-3.5 Turbo, a variant of QAP is the top performer in 2 out of 3 arithmetic tasks. QAP shows significant gains on AQuA and SAT. With GPT-4 Turbo, QAP performs the best in the same 2 out of 3 arithmetic tasks. This suggests that QAP may be more beneficial

[1] `https://platform.openai.com/docs/api-reference/chat`

Prompt	GSM8K	AQuA	SAT	StratQA
Baseline	78.7	52.8	70.9	**65.1**
QAP25	67.1	39.4	35.0	63.1
QAP50	77.8	50.0	52.7	61.4
QAP100	77.4	53.9	75.0	57.1
QAP150	78.5	**59.4**	**78.6**	53.2
QAP200	76.8	52.4	75.0	51.8
TADB	78.5	57.1	74.5	62.9
CoT	**79.0**	53.1	65.9	59.2
PS+	74.7	35.0	70.9	35.6

Table 1: Results for GPT-3.5 Turbo

Prompt	GSM8K	AQuA	SAT	StratQA
Baseline	95.3	78.7	96.8	76.3
QAP25	94.8	77.6	94.5	77.6
QAP50	93.4	**79.1**	95.9	76.9
QAP100	94.6	75.6	96.8	77.2
QAP150	94.7	78.0	97.3	77.6
QAP200	95.0	76.4	**98.2**	75.9
TADB	95.1	78.7	96.8	**78.0**
CoT	**95.6**	74.4	95.0	75.1
PS+	94.8	52.8	97.3	77.1

Table 2: Results for GPT-4 Turbo.

on questions involving algebraic and higher-level problem solving.

Commonsense Reasoning:. On StrategyQA, QAP consistently performs second-best when compared to other prompts. On both models, QAP25 is the highest QAP performer. This suggests that fewer-word explanations benefit commonsense reasoning. This is because too much explanation can cause the model to confuse a simple answer (shown in Figure 6. While there is a decline in performance as n increases on the 3.5 model, the larger GPT-4 Turbo model yields similar performances across all QAP variants.

5 Analysis

Question Difficulties Based On Baseline Performance: Within a given dataset, the difficulty of the individual question may vary. We propose a method to measure question difficulty based on performance with the baseline prompt. If the model can answer the problem correctly with the baseline prompt, then we consider the question to be *easy*; otherwise the question is *hard*. We analyze the performance of different prompts across "easy" and "hard" questions. Table 3 and Table 4 show that QAP consistently outperforms other prompts in the "hard" category.

Impact Of Word Counts On Question Difficulties: QAP generates higher word counts for both "easy" and "hard" questions (Table 5 and Table 6), despite performing lower on "easy" questions. Although more step-by-step thought processes are encouraged to avoid mistakes during reasoning, this suggests that *over*-explanation can negatively impact the model (also shown in Figure 5). Thus, the most suitable word count to solve a problem will vary from task to task; longer explanations are best suited to more complicated questions for which baseline prompting fails.

Downsides Of Smaller QAPs: Despite high performance on StrategyQA, QAP25 performs poorly on arithmetic datasets (mostly SAT and AQuA) using GPT-3.5 Turbo. Due to a small value of n, the model outputs are unfinished responses (i.e. the model stops midway through its reasoning steps) (shown in Figure 8). On SAT math, 51% of responses were incomplete for QAP25. On AQuA, 19% of responses were incomplete for QAP25.

6 Additional Studies

Placement of the prompt: In this evaluation, we studied the impact of prompt placement on performance using GSM8K dataset. Two options for prompt placement were considered: Q_Begin - adding the prompt before the question, and Q_End - adding the prompt after the question. Both placements provided similar results on GPT-3.5 and GPT-4. Results shown in the rest of the paper are based on Q_End.

No *N* Constraint: To test the effectiveness of adding the value of *N*, we first examine the prompt with just the phrase: "Explain this problem to me. Then solve for the answer". However, the model does not explain the question completely and in most cases directly starts solving the question. Its responses are no different than a response which used no given prompt. This shows that explicitly stating the minimum amount of words required is more likely to induce the model to explicitly generate an explanation of the question.

7 Related Work

In one-shot and few-shot prompting, the model is given one or more input/output examples which will serve as a demonstration for it to solve the problem using in-context learning (Mahabadi et al., 2022). QAP is a zero-shot prompt. In zero-shot

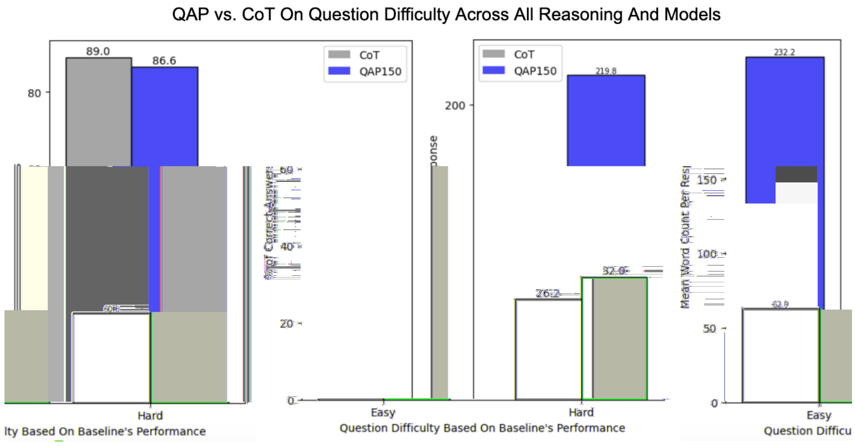

Figure 2: We consider difficulty of the problem based on baseline's results. E.g., an incorrect answer is "hard" and a correct answer is "easy". Left chart shows accuracy within each difficulty. Right chart shows mean (average) word count for within each difficulty. All results for each prompt are shown in Table 6 and Table 4

.

prompting the model does not receive exemplars, but is given a specially crafted instruction on how to approach the task (Kojima et al., 2022).

Chain of Thought: Chain-of-thought reasoning is a notable few-shot (zero-shot also exists (Yang et al., 2023) example in which the model is shown how to express its reasoning steps (Wei et al., 2022). This approach was highly effective as the model would replicate these exemplars, and their accuracies improved drastically. CoT encouraged the model to think step-by-step, and this concept would be a repeating theme among other zero-shot counterparts.

TADB: Among different variants of Zero-Shot CoT, the TADB prompt (Yang et al., 2023) was derived using an optimization objective to find instructions that would maximize task accuracy. The eventual prompt was "Take a deep breath, and work on this problem step by step". TADB is an example of how the wording of a prompt can drastically impact responses.

Plan and Solve Prompting Plus: Another zero-shot prompt is Plan-and-Solve Prompting (Wang et al., 2022). There were two versions to this prompt. The first simply asked the model devise a plan and solve step-by-step. The second version (PS+) extended the prompt by specifically asking the prompt to extract relevant variables and their corresponding numerals and to calculate intermediate results. We used PS+ on our experiments. One difference between PS+ and QAP is that PS+ prompt is more specific to math datasets since it instructs to extract variables, intermediate results, etc., whereas QAP is more general. Also, PS+ prompts the model to understand the problem, but it is not clear if model should output anything specific to the question itself. In contrast, QAP explicitly instructs the model to explain the problem in *n* words.

Question Decomposition: Question Decomposition (Radhakrishnan et al., 2023) strategy causes the model to break down the question by creating sub-questions. The model answers each of these sub-questions and it ties together all the sub-answers into a final answer. It considers two methods for decomposition, Factored Decomposition and CoT Decomposition. In factored decomposition each sub-question is answered in a separate context. CoT decomposition is an intermediate between factored decomposition and CoT. It enforces one context for sub-question, sub-answer and the answer to the original question. The analysis of question decomposition shows reduced bias and ignored reasoning, improves the faithfulness of a model-generated reasoning over CoT while retaining the performance gains of CoT.

8 Conclusion

In this paper, we explored the approach of Question-Analysis Prompting to improve LLM accuracy across math and commonsense reasoning. The prompt focuses on how the model interprets the task given, and whether restating the question in its own words can further sophisticate its answer steps. The ability of this prompting method to perform well in diverse model types, tasks difficulty, and type of tasks seems promising. We plan to extend this work further by combining QAP with other prompt strategies, applying decoding strategies and evaluating multi-modal tasks.

9 Limitations

There are a few limitations of QAP. First, LLMs are sensitive to the prompt's word choice, particularly for zero-shot prompts. As a result so small changes to the prompt wording can impact the model's performance. For example, the current QAP prompt asks the model to "solve" for the answer. While this works well for math tasks, it may not be optimal for commonsense tasks. Secondly, the results in this paper are based on four datasets and a single class of aligned models; further results should evaluate on more diverse and multi-modal datasets, as well as a greater variety of models. Finally, more robust methods (e.g., based on a classifier) to determine the choice of the parameter n should be investigated to go beyond manual selection.

10 Ethics

We experiented on three arithmetic datasets: GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), and AGIEval SAT Math (Zhong et al., 2023). For commonsense reasoning, used StrategyQA (Geva et al., 2021). GSM8K use the MIT License code, while AQUA and StrategyQA use the Apache-2.0 code. QAP and the prompts used in this work do not jeopardize the safety of others. They do not include any wording which may deem offensive to any individual or group.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. 2023. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Marzieh Saeidi, Lambert Mathias, Veselin Stoyanov, and Majid Yazdani. 2022. Perfect: Prompt-free and efficient few-shot learning with language models. *arXiv preprint arXiv:2204.01172*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson E. Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, John Kernion, Kamil.e Lukovsiut.e, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkat Chandrasekaran, Zac Hatfield-Dodds, Jared Kaplan, Janina Brauner, Sam Bowman, and Ethan Perez. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *ArXiv*, abs/2307.11768.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

A Appendix

A.1 Analysis of Accuracy Based On Question Difficulty

Performance of prompts on problems categorized into easy and hard - where easy problems are those where baseline prompt leads to a correct answer and hard problems are those where baseline prompt leads to a wrong answer. For each category the % of correct answers are calculated by number of correct answers(per prompt) over the total number of problems in that category (easy or hard)

Prompt	Easy	Hard
QAP25	84.7	30.1
QAP50	90.0	36.7
QAP100	91.5	39.5
QAP150	92.3	43.2
QAP200	91.1	41.3
TADB	93.6	34.9
CoT	92.6	35.0
PS+	88.2	31.5

Table 3: Accuracy for Arithmetic Reasoning

Prompt	Easy	Hard
QAP25	89.5	24.3
QAP50	87.7	24.6
QAP100	83.8	26.9
QAP150	81.4	27.0
QAP200	80.0	25.0
TADB	91.3	20.3
CoT	85.8	27.3
PS+	70.6	21.1

Table 4: Accuracy for Commonsense Reasoning

A.2 Analysis of Word Count based on Question Difficulty

Median word count generated by various prompts on all datasets and models categorized into easy and hard - where easy problems are those where baseline prompt leads to a correct answer and hard problems are those where baseline prompt leads to a wrong answer.

Prompt	Easy	Hard
QAP25	94.6	126.7
QAP50	123.6	158.5
QAP100	200.4	229.6
QAP150	224.4	257.9
QAP200	270.0	301.0
TADB	146.3	214.5
CoT	99.4	128.3
PS+	197.8	216.3

Table 5: Mean word count for Arithmetic Reasoning

Prompt	Easy	Hard
QAP25	36.9	38.7
QAP50	71.5	73.8
QAP100	183.8	192.3
QAP150	215.8	220.4
QAP200	268.8	274.6
TADB	37.5	58.0
CoT	29.1	30.9
PS+	162.4	179.0

Table 6: Mean word count for Commonsense Reasoning

A.3 Example Explanations

GSM8K	AQUA	StrategyQA
Q: Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make? A: To solve this problem, we need to calculate the increased value of the house after the repairs...	Q: Of the 200 students in a school, at least 45% attended the prom night and at least 35% took part in the debating session. What is the maximum number of students who could have neither attended the prom night nor the debating session? A: This problem is asking us to find the maximum number of students who did not attend either the prom night or the debating session. We are given that at least 45% of the 200 students attended the prom night, and at least 35% took part in the debating session. To solve this problem, we need to find the maximum number of students who attended either the prom night or the debating session, and then subtract that from the total number of students (200) to find the maximum number of students who did not attend either event.	Q: Would a Monoamine Oxidase candy bar cheer up a depressed friend? (False) A: No, a Monoamine Oxidase (MAO) candy bar would not cheer up a depressed friend. MAO inhibitors are a type of medication used to treat depression, but they are only available in prescription form and should be taken under the supervision of a healthcare professional. Consuming a candy bar containing MAO inhibitors without proper medical guidance can be dangerous and potentially harmful. Therefore, the answer is False.

Figure 3: Examples of QAP inducing explanations of the question on GSM8K, AQuA, and StrategyQA. The prompts include QAP50, QAP150, QAP50 respectively. Pink highlights key phrases (math reasoning) and orange highlights represents useful background information (commonsense reasoning).

A.4 Impact of Changing *n*

QAP200	QAP100
Q: Karen is paddling her canoe up a river against the current. On a still pond, Karen can paddle 10 miles per hour. The river flows in the opposite direction at 4 miles per hour. If the river is 12 miles long, how many hours will it take Karen to paddle up it? **(2 hours)** A: To solve this problem, we need to understand the concept of relative velocity. Relative velocity is the velocity of an object with respect to another object. In this case, we need to find Karen's velocity relative to the river. Karen's speed on a still pond is given as 10 miles per hour. However, when she paddles against the current of the river, her speed will be reduced. The river flows in the opposite direction at a speed of 4 miles per hour. Therefore, Karen's velocity relative to the river will be the difference between her speed and the speed of the river. To calculate Karen's velocity relative to the river, we subtract the speed of the river from her speed on a still pond: Karen's velocity relative to the river = Karen's speed on a still pond - Speed of the river = 10 miles per hour - 4 miles per hour = 6 miles per hour Now that we know Karen's velocity relative to the river, we can calculate the time it will take her to paddle up the river. We can use the formula: Time = Distance / Velocity In this case, the distance Karen needs to paddle is given as 12 miles (the length of the river), and her velocity relative to the river is 6 miles per hour. Plugging these values into the formula, we get: Time = 12 miles / 6 miles per hour = 2 hours Therefore, it will take Karen 2 hours to paddle up the river against the current.	Q: Karen is paddling her canoe up a river against the current. On a still pond, Karen can paddle 10 miles per hour. The river flows in the opposite direction at 4 miles per hour. If the river is 12 miles long, how many hours will it take Karen to paddle up it? **(2 hours)** A: To solve this problem, we need to consider the speed of Karen's canoe and the speed of the river current. Karen's canoe can paddle at a speed of 10 miles per hour on a still pond. However, when she paddles against the current of the river, her effective speed will be reduced. The river flows in the opposite direction at a speed of 4 miles per hour. To determine Karen's effective speed, we need to subtract the speed of the river current from her paddling speed. So, her effective speed against the current will be 10 - 4 = 6 miles per hour. Now, we can calculate the time it will take Karen to paddle up the river. The distance she needs to cover is 12 miles. We can use the formula: time = distance / speed. Therefore, the time it will take Karen to paddle up the river is 12 miles / 6 miles per hour = 2 hours. So, it will take Karen 2 hours to paddle up the 12-mile long river against the current.

Figure 4: This comparison shows how responses vary when changing n. This is only the answer portion. This was experimented on QAP50 and QAP20 on GSM8K on AQuA. **Blue** represents a QAP200 section which provides more detail than QAP100's (**Red**) response on the same step. **Green** represents a section that QAP200 had that QAP100 did not have at all.

A.5 Large value of n for simple problems hurts the performance

QAP50	QAP200
Q: A gecko eats 70 crickets every three days. The first day she eats 30% of the crickets. The second day she eats 6 less than the first, and the third day she finishes up the remaining crickets. How many crickets does she eat on the third day? Explain this problem to me in at least 50 words. Then solve for the answer. A: To solve this problem, we need to find out how many crickets the gecko eats on the third day. We know that the gecko eats 70 crickets every three days. On the first day, she eats 30% of the crickets, which is 0.3 * 70 = 21 crickets. On the second day, she eats 6 less than the first day, so she eats 21 - 6 = 15 crickets. Therefore, on the third day, she eats the remaining crickets, which is 70 - 21 - 15 = 34 crickets. So the gecko eats 34 crickets on the third day. Therefore, the answer is 34.	Q: A gecko eats 70 crickets every three days. The first day she eats 30% of the crickets. The second day she eats 6 less than the first, and the third day she finishes up the remaining crickets. How many crickets does she eat on the third day? Explain this problem to me in at least 200 words. Then solve for the answer. A: To solve this problem, we need to break it down into three parts: the first day, the second day, and the third day. On the first day, the gecko eats 30% of the total number of crickets. To find this, we need to calculate 30% of 70. To do this, we multiply 70 by 0.30, which gives us 21. So, on the first day, the gecko eats 21 crickets. On the second day, the gecko eats 6 less than the first day. So, we subtract 6 from 21, which gives us 15. Therefore, on the second day, the gecko eats 15 crickets. Now, we need to find out how many crickets are left for the third day. To do this, we subtract the total number of crickets eaten on the first and second days from the total number of crickets. So, we subtract 21 and 15 from 70, which gives us 34. ***Therefore, on the third day, the gecko eats 34 crickets.*** In summary, the gecko eats 21 crickets on the first day, 15 crickets on the second day, and 34 crickets on the third day. To express this as a final answer, we add up the number of crickets eaten on each day: 21 + 15 + 34 = 70. Therefore, the gecko eats a total of 70 crickets on the third day.

Figure 5: Example in which over-explanation can negatively impact a response. QAP50 acquires the correct answer (34), but QAP200 does not. In fact, QAP200 reaches the correct answer, but additional explanation leads to a wrong answer.

QAP25	QAP50
Q: Would a geographer use biochemistry in their work? A: No, a geographer would not typically use biochemistry in their work. Geographers study the Earth's physical features, climate, and human activities, while biochemistry focuses on the chemical processes in living organisms. False	Q: Would a geographer use biochemistry in their work? A: A geographer typically focuses on studying the Earth's physical features, climate patterns, and human interactions with the environment. Biochemistry, on the other hand, is a branch of science that deals with the chemical processes and substances occurring within living organisms. While there may be some overlap between the two fields, ***it is unlikely that a geographer would extensively use biochemistry in their work.*** However, there are certain scenarios where a geographer might encounter biochemistry. For example, if a geographer is studying the impact of pollution on a particular ecosystem, they may need to analyze the chemical composition of soil or water samples to understand the extent of contamination. In this case, basic knowledge of biochemistry would be helpful in interpreting the results. In conclusion, while biochemistry may not be a core component of a geographer's work, there can be instances where some understanding of biochemistry is beneficial. Therefore, the statement can be considered True.

Figure 6: Example in which over-explanation negatively impacts a commonsense reasoning response. The comparison shows that more words can confuse the model.

A.6 Word Counts for all datasets with GPT-3.5 and GPT-4

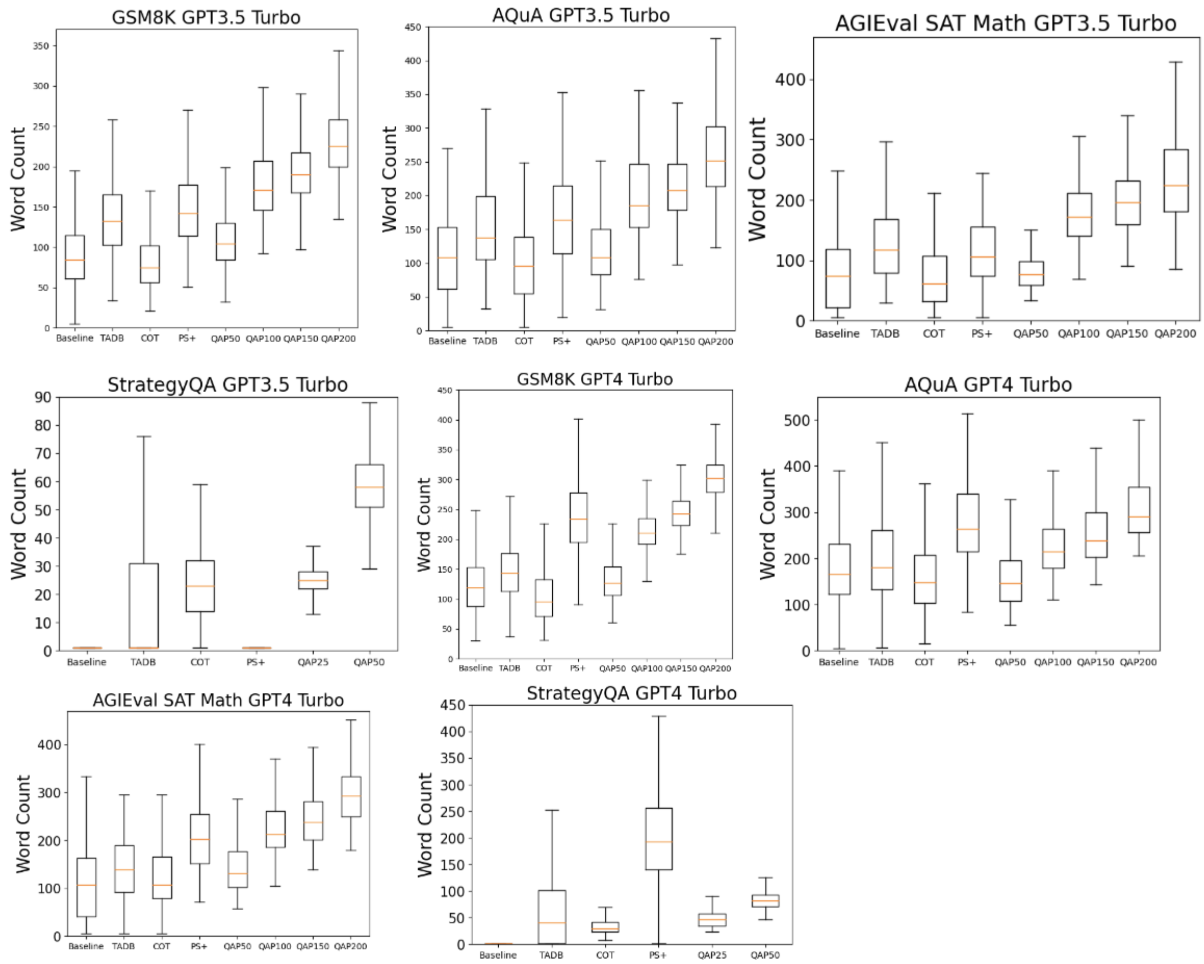

Figure 7: Median word counts in response for all datasets using GPT-3.5 Turbo and GPT-4 Turbo

A.7 QAP25 Unfinished Response

QAP25
Q: $$m = \frac{\left(\frac{r}{1{,}200}\right)\left(1+\frac{r}{1{,}200}\right)^N}{\left(1+\frac{r}{1{,}200}\right)^N - 1}P$$ The formula above gives the monthly payment m needed to pay off a loan of P dollars at r percent annual interest over N months. Which of the following gives P in terms of $m, r,$ and N ? Answer Choices: (A)$P = \frac{\left(\frac{r}{1{,}200}\right)\left(1+\frac{r}{1{,}200}\right)^N}{\left(1+\frac{r}{1{,}200}\right)^N - 1}m$ (B)$P = \frac{\left(1+\frac{r}{1{,}200}\right)^N - 1}{\left(\frac{r}{1{,}200}\right)\left(1+\frac{r}{1{,}200}\right)^N}m$ (C)$P = \left(\frac{r}{1{,}200}\right)m$ (D)$P = \left(\frac{1{,}200}{r}\right)m$
A: The formula gives the monthly payment needed to pay off a loan. We need to find the formula for the loan amount P in terms of m, r, and N.

Figure 8: Example in which QAP25 outputs an unfinished response on the SAT dataset.