Arithmetic Reasoning By GPT2

Viraat Saaran, Vishal Tyagi

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in various tasks but mathematical reasoning is still a challenging problem and an active area of research. This project explores the potential of LLMs in solving mathematical problems. Even the very large LLMs failed to answer the multi-step mathematical reasoning.We have performed training smaller LLMs, and to evaluate the results we have used the GSM8K dataset which contains the mathematical question and answer pairs. Based on the evaluations and initial results we concluded that mathematical reasoning is still a challenging area and increasing the number of parameters might help as demonstrated by other authors.

1. Introduction

In the relentless pursuit of artificial intelligence (AI) endowed with human-like intelligence, the capacity for mathematical reasoning stands as an indispensable component, as it drives ongoing efforts in the AI community to autonomously tackle math challenges. The development of autonomous math-solving capabilities appears to suggest not just a testing ground but also a pivotal catalyst for the growth of a more generalized and adept AI. As such, to optimize the growth of viable AI resources, it is crucial to delve into the intricate realms of textual comprehension, image interpretation, tabular analysis, symbolic manipulation, operational logic, and a nuanced grasp of world knowledge, which are vital in driving progress.

Recent years have reshaped the AI landscape, particularly in the realm of Large Language Models (LLMs). LLMs have earned recognition for masterful unraveling of difficult mathematical tasks, (Romera-Paredes et al., 2023, and Imani, et al., 2023) demonstrate their high-stakes accomplishments. Moreover, their innovative exploitations have rendered additional tools for LLM mathematical reasoning; facilitating insightful approaches focusing in the symphony of language and mathematical logics at large.

The increasingly multi-faceted vistas from advances towards mathematical knowledge in LLM-oriented research are notable. Diverse mathematical problem types pose a formidable challenge, exacerbated by the varied evaluation metrics, datasets, and settings employed in the assessment of LLM-oriented techniques (Testolin, 2023; Lu et al., 2023c) numerous studies present fundamental comprehension challenges: varying appraisals which defy the forging adequate tools necessary to judge gains, this significantly exacerbates extant epistemological research studies hindrances prevailing the present growth, LLM techniques deployed to tackling math problems yet evading universal frameworks.

A synthesis endeavor for shedding critical spot light onto variegated applications among LLM within a diverse discipline: to implement multi-dim rational analyses – navigating deeper factors comprising math problems integrated with inherent – varying resolution study themes devised methods; analysis solving tactics driving LLM mathematical analytical assessment with respect, Math capabilities, optimizing artificial LLM in general is key.

We looked deeply into the GSM8K dataset to understand its training and evaluation mechanism including the verifiers which helps in overall improvement of the model. Our Main contribution lies in understanding different LLMs and their strengths and weaknesses related to mathematical reasoning and giving step by step solutions. We explore various evaluation metrics and dataset that give more insight about the working of LLMs. We also planned to test the GPT3 over the GSM8k dataset while training it with a math world problem dataset.

2. Related Work

Notably, a significant gap in the existing literature on LLM applications within mathematical research, particularly in summarization, appears to exist to the best of our knowledge. Notable empirical findings from (Frieder et al., 2023a) provide valuable insight, as they compared two versions of ChatGPT, released on 9 January 2023 and 30 January 2023, respectively, and GPT-4 across math-related problems that span several areas, including proof generation, completion, a mathematical search engine capability, and assisted computation. Equity in human–LLM collaboration was underscored as Frieder et al. also put forward that such collaborative theorem proving generates added value to AI and the broader learning community. Conversely, Chang et al. (2023) analyzed the multifaceted applications of LLMs, which appeared underpinned by performance evaluations and research limitations in applied mathematical applications. Key considerations taken in our discourse - reflecting the diverse aspects relevant in this area at large – arise from i) fostering developments integral in broader applicability standards of LLMs applied within mathematical research contexts; ii) discussions engaging with current impeding factors influencing progress from not merely a narrow AI-centric perspective alone; and critically, refining and extending understandings from LLM methodology application to incorporate deeper comprehension of educational pedagogy underlying the ongoing work being performed – our presentation unfolds from this perspective.

By contrast, related experiments on simultaneous advancements employing contemporary LLM technology come from a study done almost concurrently to our efforts in (Liu et al., 2023b). Crucial demarcators establishing the basis and resultant differences marking this current narrative with attendant discussion as opposed to more established efforts primarily falls short through an offering delving less into critical considerations variously relevant across various performance, namely looking at i and ii main problems taken fully cognizant of perspectives drawn from multilateral potential scope existing across AI training to math education. Considering these areas, the inherent focal limitations which come from AI systems lacking human context demand dedicated attention.

	Name	Size	Level	Note
Q-A	CMath (Wei et al., 2023)	1.7K	E	Chinese; grade 1-6
	Sat-Math (Zhong et al., 2023)	220	H	Multi-choice
Question-Equation-Answer	SVAMP (Patel et al., 2021)	1K	E	Three types of variations
	ASDiv (Miao et al., 2020)	2.3K	E	Problem type and grade level annotated
	MAWPS (Koncel-Kedziorski et al., 2016)	3.3K	E	Extension of AddSub, MultiArith, etc.
	ParaMAWPS (Raiyan et al., 2023)	16K	E	Paraphrased, adversarial MAWPS
	SingleEq (Koncel-Kedziorski et al., 2015)	508	E	
	AddSub (Hosseini et al., 2014)	395	E	Only addition and subtraction
	MultiArith (Roy and Roth, 2015)	600	E	Multi-step reasoning
	Draw-1K (Upadhyay and Chang, 2017)	1K	E	
	Math23K (Wang et al., 2017)	23K	E	Chinese
	Ape210K (Zhao et al., 2020)	210K	E	Chinese
	K6 (Yang et al., 2023)	600	E	Chinese; grade 1-6
	CM17K (Qin et al., 2021)	17K	M H	Chinese; grade 6-12
Question-Rationale-Answer	Carp (Zhang et al., 2023a)	4.9K	M	Chinese
	GSM8K (Cobbe et al., 2021)	8.5K	M	Linguistically diverse
	MATH (Hendrycks et al., 2021)	12.5K	H	Problems are put into difficulty levels 1-5
	PRM800K (Lightman et al., 2023)	12K	H	MATH w/ step-wise labels
	MathQA (Amini et al., 2019)	37K	C	GRE examinations; have quality concern
	AQuA (Ling et al., 2017)	100K	C	GRE&GMAT questions
	Arb (Sawada et al., 2023)	105	C	Contest problems and university math proof
	Ghosts (Frieder et al., 2023b)	709	C	
	TheoremQA-Math (Chen et al., 2023b)	442	C	Theorem as rationale
	LILA (Mishra et al., 2022)	132K	H	Incorporates 20 existing datasets
	Math-Instruct (Yue et al., 2023)	260K	H	Instruction-following style
	TabMWP (Lu et al., 2023b)	38K	H	Tabular MWP; below the College level

Table 1: Datasets for Math Word Problems.
E= Elementary, M= Middle School, H= High School, C= College, H= Hybrid

Figure 1. Related works

3. Methodology

3.1. Dataset

The GSM8K dataset comprises 8.5K high-grade elementary mathematics problems crafted by human problem designers, which we subsequently divided into 7.5K training problems and 1K test problems, thus facilitating a comprehensive evaluation of language model performance. These problems necessitate between 2 and 8 steps to solve, and their solutions generally involve a sequence of fundamental arithmetic operations (+, , ×, ÷) to ultimately arrive at the correct answer, a task that a skilled middle school student should be able to accomplish. In creating the GSM8K dataset, we adhered to the following guiding principles:

- High quality: We deliberately eschewed potentially error-prone data scraping techniques and instead relied on human problem writers, implementing rigorous quality control procedures to minimize errors and opting to retain only the highest-quality problems; our ensuing quality control process revealed a probable error rate of less than 2 percent based on worker agreement.
- High diversity: A hallmark of our dataset is its elevated level of diversity among problems, achieved through the deliberate avoidance of linguistic templates that could potentially yield artificially similar problems; this ensures that performance on the held-out test is a far more meaningful indicator of language model performance.
- Moderate difficulty: By selecting a problem distribution that proved challenging but not entirely intractable for state-of-the-art language models, we established a diagnostic regimen capable of capturing the scaling trends of disparate models and methods; although a number of early algebraic concepts could theoretically be applied to solve the problems, in practice, most can be resolved by straightforward arithmetic operations without requiring explicit variable declarations.
- Natural language solutions: To derive insights from large language models, we compiled the problems' solutions in natural language, rather than recording their structures in precise mathematical expressions. Per our explicit instructions, problem writers extensively expatiated on their deliberations with diverse, often colloquial linguistic styles.

3.2. Models

3.3. GPT-2

The **Generative Pre-trained Transformer 2 (GPT-2)** is OpenAI's second foundational large language model, leveraging pre-training across 8 million web pages to establish its architecture. Phased deployment commenced in February 2019, with full operationalization of the 1.5-billion-parameter framework achieved by November 5, 2019.

3.3.1. Overview

- **Model Type**: GPT-2 is a decoder-only transformer model designed for text generation.
- **Parameter Size**: It has 1.5 billion parameters.
- **Training Objective**: The model was trained using a causal language modeling (CLM) objective.

3.3.2. Performance

- **Evaluation Metrics**: GPT-2 has been evaluated on various benchmarks, demonstrating significant performance improvements over its predecessor, GPT-1.
- **Capabilities**: It showcases strong performance in tasks such as text generation, translation, summarization, and question answering.

3.3.3. Use Cases

- **Text Generation**: GPT-2 is widely used for generating coherent and contextually relevant text.
- **Downstream Tasks**: It can be fine-tuned for various tasks such as sentiment analysis, summarization, and more.

3.3.4. Availability

- **Access**: GPT-2 is available on platforms like Hugging Face, allowing researchers and developers to use and fine-tune the model.
- **License**: The model is released under an open license, promoting reproducible and responsible research.

3.4. OPT-350M

The **OPT-350M** model is part of the **Open Pre-trained Transformer (OPT)** series developed by Meta AI. Here are some key details about OPT-350M:

3.4.1. Overview

- **Model Type**: OPT-350M is a decoder-only transformer model designed for text generation.
- **Parameter Size**: It has 350 million parameters.
- **Training Objective**: The model was trained using a causal language modeling (CLM) objective, similar to GPT-3.

3.4.2. Performance

- **Evaluation Metrics**: OPT-350M has been evaluated on various benchmarks, including **ARC**, **HellaSwag**, **MMLU**, **TruthfulQA**, **WinoGrande**, and **GSM8K**.
- **Capabilities**: It demonstrates strong performance in zero-shot and few-shot learning tasks, making it suitable for a variety of natural language processing applications.

3.4.3. Use Cases

- **Text Generation**: The model can be used for generating coherent and contextually relevant text.
- **Downstream Tasks**: It can be fine-tuned for specific tasks such as sentiment analysis, summarization, and question answering.

3.4.4. Availability

- **Access**: OPT-350M is available on platforms like Hugging Face, allowing researchers and developers to use and fine-tune the model.
- **License**: The model is released under an open license, promoting reproducible and responsible research.

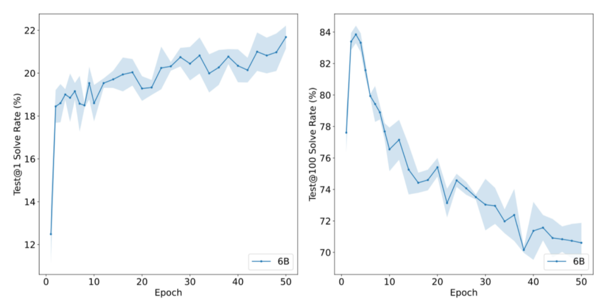


Figure 2. Methodology

3.5. Verification

Training solutions will be evaluated based on their ability to reach the correct final answer, with 'correct' or 'incorrect' labels assigned accordingly. However, some solutions may arrive at the correct final answer using flawed reasoning, resulting in false positives. To address this, we will adopt the following verifier training approach:

1. By finetuning a model (the "generator") for 2 epochs on the training set, we will enable it to acquire basic skills in this domain.

2. 100 completions are then sampled from the generator for each training problem, and their solutions are labeled as correct or incorrect.

3. The verifier is consequently trained for a single epoch on this dataset. A 2-epoch training epoch was chosen to optimize the generator's training process—beyond this point, solution diversity tends to collapse.

To prevent overfitting, generator and verifier models are trained separately, though combining them is theoretically possible. We elect to keep the model size identical for both the generator and verifier, and as such, unless otherwise specified, utilize the same model configuration. Enhancing its primary function, the verifier also trains using the generator's language modeling objective, serving as a valuable auxiliary objective.

Testing procedure includes the following steps: we sample 100 completions for each test problem, subsequently ranking them using the verifier; ultimately, we return the solution with the highest verifier score. For comparison purposes, the figure below delves into the efficacy of verification vis-a-vis finetuning for both the 6B and 175B model sizes, clearly demonstrating that for larger datasets, a verifier can significantly enhance model performance. Conversely, when working with smaller datasets, verification doesn't seem to bring a notable advantage—this is, in all likelihood, an outcome of a propensity to overfit the correct answer. As indicated by efficiency gains at scale, using verification produces distinctly better outcomes once sufficient data is made available to optimize solution accuracy.

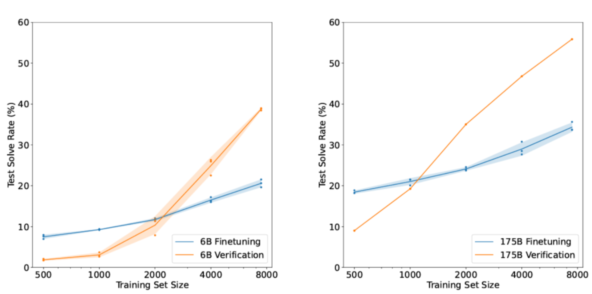


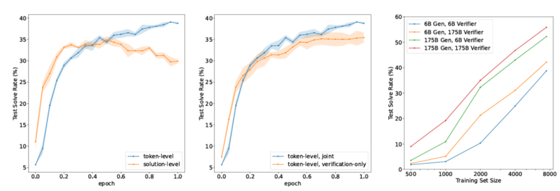
Figure 3. Verification

3.6. Verification Ablation

Given two training approaches for verifiers, we can either condition a single scalar prediction on the entire solution generation or opt for a token-by-token prediction. The latter approach is our default, constituting a token-level value function. Figure A compares the performance of both methods, denominated as "solution level" and "token level." Contrary to expectations, predicting the value function after each token proves a more complex task compared to judging a full completion, given the increased noise level. While early training is slower, the token-level verifier eventually outperforms the solution-level verifier, continually demonstrating improvement versus the solution-level verifier's potential for overfitting. The provision of a full value function implies an auxiliary signal, facilitating thorough reasoning throughout solution generations, in addition to simply recalling correct answers.

Selecting an objective in training verifiers presents possibilities. Ablations in Figure B weigh an integrated language modeling objective during verification against singular use of a verification objective. Findings indicate combining both yields strict performance improvements. An inclusion of this language modeling objective proves reasonable and prudent. Evidential outcomes become even more notable when looking into component variability - specifically training size vs verifier, as figures in C indicate, better options for permuting variable elements. Fact-checking processes performed by verifiers display some reliance on some fixed "factors-checking-to discriminate subelements," apparently without invoking some substantial portions subelements found overall verification-processes as conjectured through various experiments during studies.

Results shown through verification-effect metrics comparison (over the generator size - though comparable results in the presence smaller verifiers may still ensue efficiency benefits according to these experiments).



(a) Comparison between a verifier trained to predict correctness after every token (token-level) and one trained to predict correctness after only the final token (solution-level)

(b) Comparison between a verifier trained jointly to predict correctness and perform language modeling (joint) and one trained only to predict correctness (verification-only)

(c) Performance when varying the size of the generator and the verifier in isolation. Increasing the size of the generator has a larger impact than increasing the size of the verifier.

Figure 4. Verification Ablation

4. Experiments and Results

4.1. Training Details : GPT2

The training configuration parameters are present in the final checkpoint files. We have used pretrained GPT2 model and fine tuned it with GSM8K Dataset. We have trained GPT2 model on both the test, test-socratic dataset.

The training is done on different epochs ranging from 10 epochs to 50 epochs. The time taken to pretrain the model on this dataset is from 50 minutes to 5 hours on horeka cluster using 4 gpu.

4.2. Training Details: OPT*350M

The parameters for training configurations are also present in final checkpoint folders on github. We have used pre-trained OPT model and fine-tuned it on GSM8K Dataset. Since this is a large model compared to GPT2 trainig it was more complex and time and resource consuming. THe time taken to train 10 epochs took around 2.5 hours and for 20 epochs around 4-5 hours. on horeka cluster using 4 gpu.

4.3. Results

These are the loss values after testing the trained model on test and test-socratic dataset.

	gpt2_10	gpt2_20	gpt2_50
test	7.40	7.84	10.04
test_socratic	7.41	7.84	10.06

	opt350m_10	opt350m_20
test	10.15	10.89

Figure 5. Results for the training.

Please see the supplementary for detailed results.

5. Challenges

Using GPT-2 for the GSM8K dataset presents several challenges:

5.1. 1. Complexity of Mathematical Reasoning

- **Lack of Mathematical Intuition**: GPT-2, like many language models, lacks inherent mathematical intuition. It struggles with understanding and solving complex math problems that require multi-step reasoning.

- **Error Propagation**: Small mistakes in intermediate steps can propagate, leading to incorrect final answers. This is particularly problematic for math problems, where accuracy in each step is crucial.

5.2. 2. Dataset Limitations

- **Data Representation**: Properly formatting and representing mathematical problems and solutions in a way that GPT-2 can understand is challenging. The model needs to accurately interpret symbols, equations, and word problems.

- **Training Data**: GSM8K, while extensive, may still lack the diversity required for the model to generalize well to all types of math problems. This can lead to overfitting on the training data.

5.3. 3. Model Architecture

- **Parameter Size**: GPT-2, with its 1.5 billion parameters, may not be large enough to fully capture the complexity of mathematical reasoning required for GSM8K. Larger models like GPT-3 or GPT-4 may be more suitable but come with increased computational costs.

- **Contextual Understanding**: GPT-2's ability to maintain long-term context is limited. Math problems often require understanding and retaining information over multiple steps, which can exceed the model's context window.

5.4. 4. Training Challenges

- **Hyperparameter Optimization**: Fine-tuning GPT-2 requires careful optimization of hyperparameters like learning rate, batch size, and training epochs. Incorrect settings can lead to poor performance.

- **Resource Intensive**: Fine-tuning and training large language models like GPT-2 require significant computational resources, including powerful GPUs and substantial memory.

5.5. 5. Evaluation and Verification

- **Performance Metrics**: Standard performance metrics like accuracy, precision, recall, and F1-score may not fully capture the model's ability to solve complex math problems. Evaluating intermediate steps and reasoning is also necessary.

- **Verification**: Ensuring the correctness of generated solutions is challenging. Verification systems need to be in place to check and validate the answers produced by the model.

5.6. 6. Post-Processing

- **Error Analysis**: Identifying and addressing the types of errors made by GPT-2 is crucial for improving its performance. This requires thorough analysis and iterative refinements.

- **Human-in-the-Loop**: Incorporating human feedback can help refine the model's performance, but this process is time-consuming and resource-intensive.

By addressing these challenges through data augmentation, model architecture improvements, optimized training techniques, continuous evaluation, and robust post-processing methods, the performance of GPT-2 on the GSM8K dataset can

6. Conclusion

The GSM8K dataset can be used as a base dataset for fine-tuning the GPT-2 model. By leveraging GSM8K, researchers can enhance GPT-2's performance on mathematical reasoning tasks. This involves optimizing hyperparameters like temperature and beam size to improve the model's problem-solving accuracy. The GSM8K dataset provides a robust framework for training and evaluating LLMs, making it an essential tool for advancing AI capabilities in mathematical reasoning.

Several strategies can be employed to improve the performance of GPT-2 on the GSM8K dataset:

1. **Data Augmentation**
   - **Increase Training Data**: Incorporate more math problems to provide a wider variety of examples for the model to learn from.
   - **Synthetic Data**: Generate synthetic math problems that mimic the structure of the GSM8K dataset to enhance the training set.
2. **Model Architecture**
   - **Larger Models**: Use larger versions of the GPT series, such as GPT-3 or GPT-4, which have more parameters and can potentially understand and solve problems more effectively.
   - **Ensemble Methods**: Combine predictions from multiple models to improve accuracy and robustness.
3. **Training Techniques**
   - **Curriculum Learning**: Start training with simpler problems and gradually increase the complexity, helping the model build a strong foundation before tackling more difficult problems.
   - **Transfer Learning**: Fine-tune the model on related tasks before training on GSM8K to leverage pre-existing knowledge.
4. **Evaluation and Feedback**
   - **Continuous Evaluation**: Regularly assess the model's performance on a validation set and use the results to adjust training strategies.
   - **Human-in-the-Loop**: Incorporate human feedback to refine and improve the model's problem-solving approach.
5. **Post-Processing**
   - **Verification Systems**: Implement verification systems to check the correctness of the solutions generated by the model and correct any mistakes.
   - **Error Analysis**: Analyze the types of errors the model makes and develop strategies to address these specific weaknesses.

7. Future Works

Further progress in this area can be achieved by using college-level datasets, notably GHOSTS and its counterparts. This dataset is developed by first creating a collection of 709 prompts from various sources, and subsequently evaluating the models on (subsets of) these data points. We rate the corresponding outputs provided by the models and collect statistics, such as error types, output lengths, or the stability of the answer under prompt engineering. This yields a total of 1636 ratings by human experts.

Evaluating GPT-4 involved utilizing the miniGHOSTS dataset, comprising 170 carefully chosen prompts that capture the essence of model performance across various subdatasets. This distilled set was selected by applying a heuristic method, where 10 prompts from each GHOSTS subdataset file were chosen to match the mean rating and standard deviation of the original file, thus retaining the most relevant characteristics; more details on this process are available in Appendix H. By leveraging miniGHOSTS, the resource-intensive evaluation process for new language models, including GPT-4, can be efficiently reduced. Efficiency was further optimized by extracting one representative question from each file in the miniGHOSTS dataset, resulting in the microGHOSTS dataset, consisting of 14 questions.

Given the benefits of this streamlined approach, microGHOSTS ultimately enables rapid pre-screening of language models, with questions specifically selected to represent the mathematical problems that a model would typically struggle with. To support consistent evaluation, the microGHOSTS dataset is accompanied by reference solutions, explanations, and LLM failure modes for raters, taking into account that some may not possess in-depth mathematical knowledge.

8. References

1. Ahn J., Verma R., Lou R., Liu D., Zhang R.and Yin W., ”Large Language Models for Mathematical Reasoning: Progresses and Challenges,” https://arxiv.org/abs/2402.00157, 2024.

2. Cobbe K., Kosarraju V., Bavarian M., Chen M., Jun H., Kaiser L., Plappert M., Tworek J., Hilton J., Nakano R., Hesse C., Schulman J., ”Training Verifiers to Solve Math Word Problems,” https://arxiv.org/abs/2110.14168, 2021.

3. Chang Y., Wang X., WU Y., Yang L., ZHU K., Chen H., YI X., Wang C., YE W., Zhang Y., Chang Y., YU P., Yang Q., Xie X., ”TA Survey on Evaluation of Large Language Models,” https://arxiv.org/abs/2307.03109, 2023.

4.OpenAI, ”GPT-4 Technical Report,” arXiv preprint arXiv:2309.07778,https://arxiv.org/abs/2303.08774, 2023.

5. Jiang A., Sablayrolles A., Roux A., Mensch A., Savary B., Bamford C., Chaplot D., Casas D., Hanna E., Bressand F., Lengyel G., Bour G., Lample G., Lavaud Lelio, Saulnier L., Lachaux Marie., Stock P., Subramanian S., Yang S., Antoniak S., Scao T., Gervet T., Lavril T., Wang T., Lacroix T., Sayed W., ”Mixtral of Experts,” https://arxiv.org/abs/2401.04088 [eess.IV], 2024.

6. Frieder S., Pinchetti L., Chevalier A., Griffiths R., Salvatori T., Lukasiewixz T., Petersen P., Berner J., ”Mathematical Capabilities of ChatGPT,” https://arxiv.org/abs/2301.13867, 2023.

References

9. Supplementary

9.1. Gpt50 socratic Result

Prompt: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? Response: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? How many apples does Jenny give to her friend? ** Jenny gives 7 x 7 = ¡¡7*7=42¿¿42 apples to her friend. How many apples does Jenny have left? ** Jenny has 23 - 42 = ¡¡23-42=3¿¿3 apples left. 3

Prompt: If one box can hold 6 oranges, how many oranges can 9 boxes hold? Response: If one box can hold 6 oranges, how many oranges can 9 boxes hold? How many oranges can each box hold? ** Each box can hold 6 x 9 = ¡¡6*9=54¿¿54 oranges. How many oranges can each box hold? ** Therefore, each box can hold 54/6 = ¡¡54/6=9¿¿9 oranges. 9

Prompt: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? Response: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? How many chocolate bars does Sarah have? ** Sarah has 3/4*3/4=¡¡3/4*3/4=2¿¿2 chocolate bars. How many chocolate bars does Sarah have left? ** She has 2-1=¡¡2-1=1¿¿1 chocolate bar left. 1

Prompt: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? Response: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? How many students are in each group? ** Each group has 30 students / 5 students/group = ¡¡30/5=4¿¿4 students/group. How many groups will there be? ** So there will be 4 groups * 5 students/group = ¡¡4*5=20¿¿20 groups. 20

Prompt: Tom’s age is twice that of his brother’s age. If his brother is 12 years old, how old is Tom? Response: Tom’s age is twice that of his brother’s age. If his brother is 12 years old, how old is Tom? How old is Tom’s brother? ** Tom’s brother is 12 - 2 = ¡¡12-2=10¿¿10 years old. How old is Tom’s brother’s age? ** So his age is 10 * 2 = ¡¡10*2=20¿¿20 years old. How old is Tom’s brother? ** Therefore, Tom’s age is 20 +

Prompt: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? Response: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? How much money did Viraat spend? ** Viraat spent 20 x 30 = ¡¡20*30=600¿¿600 dollars. How much money did Viraat have left? ** So, Viraat had 93 - 600 = ¡¡93-600=90¿¿90 dollars left. 90

9.2. Gpt20 Result

Prompt: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? **Response**: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? Jenny has 23 - 7 = ¡¡23-7=15¿¿15 apples left. Jenny has 15 / 7 = ¡¡15/7=2¿¿2 apples left. 2

Prompt: If one box can hold 6 oranges, how many oranges can 9 boxes hold? **Response**: If one box can hold 6 oranges, how many oranges can 9 boxes hold? Let x be the number of oranges in a box. The number of oranges in a box is 9 x 6 = ¡¡9*6=36¿¿36 oranges. The number of oranges in a box is x + 36 = ¡¡36+36=72¿¿72 oranges. 72

Prompt: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? **Response**: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? She has 3/4*1/2=¡¡3/4*1/2=1¿¿1/2 of the chocolate bar. She has 1/2*1/2=¡¡1/2*1/2=1¿¿1/2 of the chocolate bar. 1

Prompt: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? **Response**: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? There are 30 x 5 = ¡¡30*5=150¿¿150 students in the classroom. There are 150 - 30 = ¡¡150-30=90¿¿90 students in the classroom. 90

Prompt: Tom's age is twice that of his brother's age. If his brother is 12 years old, how old is Tom? **Response**: Tom's age is twice that of his brother's age. If his brother is 12 years old, how old is Tom? Tom is 12-2=¡¡12-2=10¿¿10 years old. Tom's brother is 10+2=¡¡10+2=14¿¿14 years old. 14

Prompt: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? **Response**: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? Viraat has 93 - 20 = ¡¡93-20=42¿¿42 dollars. He has 42 / 30 = ¡¡42/30=3¿¿3 dollars left. 3

9.3. GPT-20-Socratic-Result

Prompt: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? Response: Jenny has 23 apples. She gives 7 apples to her friend. How many apples does Jenny have left? How many apples does Jenny have left? ** Jenny has 23 - 7 = ¡¡23-7=21¿¿21 apples left. How many apples does Jenny have left? ** Jenny has 21 - 7 = ¡¡21-7=14¿¿14 apples left. 14

Prompt: If one box can hold 6 oranges, how many oranges can 9 boxes hold? Response: If one box can hold 6 oranges, how many oranges can 9 boxes hold? How many oranges can 9 boxes hold? ** 9 boxes can hold 6 oranges x 6 oranges = ¡¡9*6=36¿¿36 oranges. How many oranges can 9 boxes hold? ** 9 boxes can hold 36 oranges x 9 boxes = ¡¡36*9=96¿¿96 oranges. 96

Prompt: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? Response: Sarah has 3/4 of a chocolate bar. She eats 1/2 of what she has. How much of the chocolate bar does she have left? How many chocolate bars does Sarah have? ** Sarah has 3/4*1/2=¡¡3/4*1/2=2¿¿2 chocolate bars. How many chocolate bars does Sarah have left? ** She has 2*1/2=¡¡2*1/2=2¿¿2 chocolate bars.

Prompt: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? Response: A classroom has 30 students. If the teacher wants to split them into groups of 5, how many groups will there be? How many students are in the group? ** The number of students in the group is 30-5 = ¡¡30-5=20¿¿20 students. How many students are in the group? ** The number of students in the group is 20-5 = ¡¡20-5=10¿¿10 students. How many students are in the group? ** The

Prompt: Tom's age is twice that of his brother's age. If his brother is 12 years old, how old is Tom? Response: Tom's age is twice that of his brother's age. If his brother is 12 years old, how old is Tom? How old is Tom? ** Tom is 12 years old because 12 = ¡¡12=12¿¿12 How old is Tom's brother? ** His brother is 12 years old because 12 = ¡¡12=12¿¿12 12

Prompt: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? Response: Viraat has 93 dollars. He spent 20 dollars on sweets. He get his 30 dollars back from someone else. How much money does Viraat Have left? How much money did Viraat have after 20 dollars? ** After 20 dollars, Viraat had 93 - 20 = ¡¡93-20=42¿¿42 dollars. How much money did Viraat have after 30 dollars? ** After 30 dollars, Viraat had 42 - 20 = ¡¡42-20

	gpt2_10	gpt2_20	gpt2_50
test	7.40	7.84	10.04
test_socratic	7.41	7.84	10.06

	opt350m_10	opt350m_20
test	10.15	10.89
test_socratic	10.59	11.10

Figure 6. Results for the training.