Mathify: Evaluating Large Language Models on Mathematical Problem Solving Tasks

Avinash Anand
MIDAS Labs
IIIT-Delhi, Delhi, India
avinasha@iiitd.ac.in

Mohit Gupta
MIDAS Labs
IIIT-Delhi, Delhi, India
mohit22112@iiitd.ac.in

Kritarth Prasad
MIDAS Labs
IIIT-Delhi, Delhi, India
kritarth20384@iiitd.ac.in

Navya Singla
MIDAS Labs
IIIT-Delhi, Delhi, India
singlanavya01@gmail.com

Sanjana Sanjeev
MIDAS Labs
IIIT-Delhi, Delhi, India
sanjana21094@iiitd.ac.in

Jatin Kumar
MIDAS Labs
IIIT-Delhi, Delhi, India
jatin20206@iiitd.ac.in

Adarsh Raj Shivam
MIDAS Labs
IIIT-Delhi, Delhi, India
adarsh20274@iiitd.ac.in

Rajiv Ratn Shah
MIDAS Labs
IIIT-Delhi, Delhi, India
rajivratn@iiitd.ac.in

Abstract

The rapid progress in the field of natural language processing (NLP) systems and the expansion of large language models (LLMs) have opened up numerous opportunities in the field of education and instructional methods. These advancements offer the potential for tailored learning experiences and immediate feedback, all delivered through accessible and cost-effective services. One notable application area for this technological advancement is in the realm of solving mathematical problems. Mathematical problem-solving not only requires the ability to decipher complex problem statements but also the skill to perform precise arithmetic calculations at each step of the problem-solving process. However, the evaluation of the arithmetic capabilities of large language models remains an area that has received relatively little attention. In response, we introduce an extensive mathematics dataset called "**MathQuest**" sourced from the 11th and 12th standard Mathematics NCERT textbooks. This dataset encompasses mathematical challenges of varying complexity and covers a wide range of mathematical concepts. Utilizing this dataset, we conduct fine-tuning experiments with three prominent LLMs: LLaMA-2, WizardMath, and MAmmoTH. These fine-tuned models serve as benchmarks for evaluating their performance on our dataset. Our experiments reveal that among the three models, **MAmmoTH-13B** emerges as the most proficient, achieving the highest level of competence in solving the presented mathematical problems. Consequently, **MAmmoTH-13B** establishes itself as a robust and dependable benchmark for addressing NCERT mathematics problems.

NeurIPS'23 Workshop on Generative AI for Education (GAIED).

1 Introduction

Mathematical problem-solving represents a multifaceted cognitive skill, encompassing the comprehension of problem statements, identification of pertinent concepts and formulas, application of suitable strategies and algorithms, precise calculations, and the verification of solution validity and reasonableness. Traditionally, mathematical problem-solving has been imparted and assessed through conventional means such as textbooks, worksheets, and examinations, often affording limited feedback and learner guidance. Furthermore, these methods may not fully capture the diversity and intricacy of real-world mathematical challenges encountered by students.

In the era of rapid advancements in artificial intelligence and natural language processing (NLP), large language models (LLMs) have emerged as formidable tools for generating natural language text across a spectrum of domains and tasks [8]. LLMs, grounded in the transformer architecture [28], have the capacity to glean long-range dependencies and contextual representations from vast corpora of text data. These LLMs have showcased impressive proficiency in mathematical reasoning and problem-solving by leveraging their inherent understanding of arithmetic operations, algebraic principles, and symbolic manipulation. Nevertheless, existing LLMs grapple with substantial hurdles in tackling math word problems, particularly those necessitating intricate reasoning, multi-step arithmetic calculations, or domain-specific knowledge [9, 16, 33].

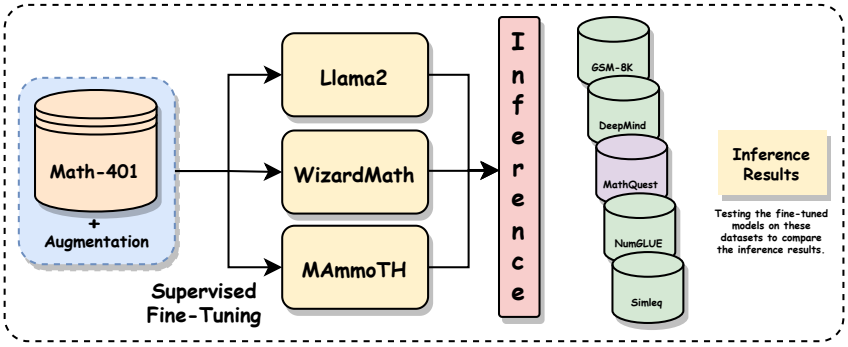

Figure 1: This figure shows the fine-tuning flow, the LLMs we use for fine-tuning, and the datasets we use for inference.

The advent of large language models (LLMs) has proven to be a boon in the field of education, as evidenced by recent studies [21, 25, 35]. These versatile models have ushered in a new era of learning possibilities, catering to individual student needs by considering their preferences, objectives, interests, and aptitudes. For instance, LLMs offer a tailored learning experience, providing personalized feedback, guidance, explanations, and recommendations [12]. Educators, too, find these models invaluable, as they simplify the creation of engaging learning materials such as quizzes, summaries, questions, and exercises [23]. Notably, LLMs can even generate multiple-choice questions based on provided text passages. Additionally, these models excel in enhancing language proficiency, aiding learners in vocabulary, grammar, pronunciation, and fluency [12]. Their versatility extends to assisting students and researchers in exploring new topics and extracting information from diverse sources. They effortlessly generate summaries [34], identify keywords, generate citations [13], and provide relevant links in response to queries.

This paper endeavors to tackle the challenges posed by mathematical problem-solving within the context of LLMs. To this end, we introduce MathQuest, a comprehensive mathematics dataset meticulously curated from the 11th and 12th standard Mathematics NCERT textbooks[1]. This dataset spans various levels of mathematical complexity and encompasses a wide array of mathematical concepts. We introduce this dataset because existing open-source datasets primarily consist of relatively straightforward mathematical problems. In contrast, standard mathematical problems can be significantly more complex. To equip Large Language Models (LLMs) with the ability to solve these intricate problems, we conduct fine-tuning on this dataset. Furthermore, we propose a novel approach for fine-tuning three preeminent LLMs: MAmmoTH [37], LLaMA-2 [27], and WizardMath [19]

[1] `https://ncert.nic.in/`

using our MathQuest dataset. Our evaluation encompasses not only the performance of these fine-tuned models on our dataset but also their proficiency on other openly accessible mathematical reasoning datasets. Our findings indicate that **MAmmoTH-13B** outshines its counterparts, emerging as the most adept and proficient in solving the mathematical challenges presented. Thus, **MAmmoTH-13B** establishes itself as a dependable and robust baseline for addressing NCERT mathematics problems.

2 Related Work

In this section, we delve into the existing literature, unveiling a diverse array of approaches utilizing Large Language Models (LLMs) for tackling mathematical problems.

Recent research has highlighted the potential of Large Language Models (LLMs) in education. They offer promise in automating question generation and supporting direct interactions within the learning environment [14]. Furthermore, investigations have explored few-shot prompting techniques over LLMs for addressing mathematical word problems [31, 38, 7]. The "chain-of-thought" prompting approach [31] leverages explicit intermediate reasoning steps to bolster the LLM's reasoning abilities. To mitigate arithmetic errors commonly observed in LLMs [17, 10], earlier studies [3] have explored the use of external calculators to execute operations generated by LLMs.

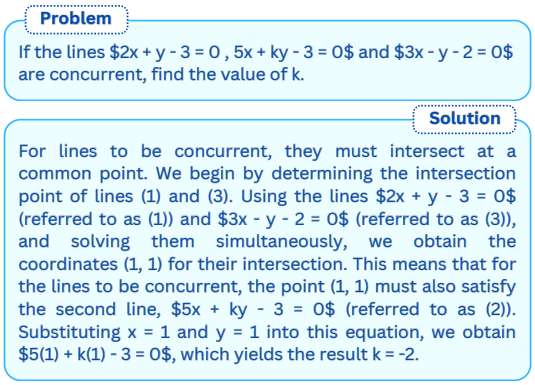

Figure 2: Our Dataset **MathQuest** Sample

Furthermore, [32] presents a novel method tailored for addressing elementary arithmetic and logical problems. This method concatenates the generated answer with the original problem statement, tasking the model with predicting the initial conditions to verify the accuracy of the answer. Notably, a subset of these approaches [6, 1] can function effectively with zero-shot prompts, offering a versatile approach to mathematical problem-solving. A specialized method, MathPrompter [11], targets the enhancement of arithmetic operations and reasoning capabilities of LLMs, particularly designed to facilitate mathematical problem-solving tasks.

Various approaches exist for enhancing mathematical problem-solving with Large Language Models (LLMs). Wang et al.'s self-consistency [30], built on the CoT framework, assesses multiple potential reasoning paths and selects answers via majority vote. [18] extend self-consistency by teaching a verifier to validate each step, while [20] use recent LLMs like GPT-3.5 to generate an output, provide feedback, and prompt the model for improvements. [29] evaluate pretrained language models on basic arithmetic expressions, including addition $(+)$ and subtraction $(-)$, and [24] expand the assessment to include multiplication $(*)$ operations within the language models' scope.

3 Dataset

For our research experiments, we employed the Math-401 dataset [36], which encompasses 401 samples of mathematical problems. This dataset encompasses a diverse range of mathematical operations, including addition $(+)$, subtraction $(-)$, multiplication $(*)$, division $(/)$, exponentiation, trigonometric functions $(\sin, \cos, \tan)$, logarithmic functions $(\log, \ln)$, and incorporates integers, decimals, and irrational numbers (π, e). Recognizing the limited sample size of this dataset for

effective learning by large language models, we expanded it through augmentation, resulting in a dataset size of $302,000$ samples. To construct our augmented dataset, we employed the ***SymPy*** Python library. This library allowed us to generate arithmetic mathematical equations along with their corresponding ground truth values. These equations covered basic arithmetic operators such as addition (+), subtraction (-), multiplication (*), and division (/). Furthermore, the dataset includes extensive arithmetic expressions with brackets, mimicking the complexity often encountered in real-world math word problems. Table 1 provides a comprehensive breakdown of the question types utilized in the creation of our augmented dataset. Furthermore, we evaluated our model's performance on four additional datasets: GSM-8K [4], DeepMind [26], NumGLUE [22], and SimulEq [15].

Type	Range	Decimal Places (1 - 4)	Variables	Count
Small Integer	[-20, 20]	×	(x, y)	65,000
Small Decimal	[-20, 20]	✓	(x, y)	35,000
Small Decimal + Integer	[-20, 20]	✓	(x, y)	39,000
Large Integer	[-1000, 1000]	×	(x, y)	39,000
Large Decimal	[-1000, 1000]	✓	(x, y)	25,000
Large Decimal + Integer	[-1000, 1000]	✓	(x, y)	25,000
3 Terms	[-100, 100]	✓	(x, y, z)	25,000
4 Terms	[-100, 100]	✓	(w, x, y, z)	49,000
Total	-	-	-	302,000

Table 1: The distribution of types of question in our augmented Math-401 dataset

3.1 Our Dataset: MathQuest

We have meticulously curated our own dataset, referred to as **MathQuest**, sourcing problems from high school mathematics NCERT books. MathQuest is a rich resource, encompassing word problems of varying complexities and spanning diverse mathematical concepts. Our dataset comprises a total of 14 overarching mathematical domains, including sets, trigonometry, binomial theorem, and more. The distribution of samples across these concepts is visually represented in Figure.3. Our dataset contains total of 223 samples. Notably, as depicted in the charts, the category of "Sequence and Series" boasts the highest number of problems within our dataset. To provide a glimpse of our dataset's structure, we present a sample from MathQuest in Figure.2.

4 Methodology

This research aims to enhance the mathematical problem-solving capabilities of large language models. Initially, we observed that existing open-source models such as LLaMA-2 [27] and Vicuna [2] struggled with elementary mathematical tasks like simple addition and subtraction. This observation served as the catalyst for our research, motivating us to improve LLMs' proficiency in comprehending and accurately solving mathematical problems.

To achieve this, we adopted a instructive approach reminiscent of teaching mathematics to students. We commenced by imparting a clear understanding of fundamental operators such as $+, -, *, /$, gradually progressing to more advanced operators and expressions. Similarly, we endeavored to acquaint LLMs with the meanings of mathematical operators and expressions. To facilitate this process, we leveraged the Math-401 dataset [36], a valuable resource comprising 401 data samples consisting of basic mathematical questions and their corresponding answers. Given the dataset's limited size, we augmented it to introduce greater diversity and complexity, ensuring that the model could grasp and master advanced mathematical concepts during training.

For the fine-tuning process, we employed three prominent large language models: LLaMA-2 [27], WizardMath [19], and MAmmoTH [37]. LLaMA-2 [27] represents an upgraded version of LLaMA, refined through training on an enriched mixture of publicly available data. The enhancements encompass a 40% increase in the pre-training corpus size, a doubling of the model's context length, and the incorporation of grouped-query attention.

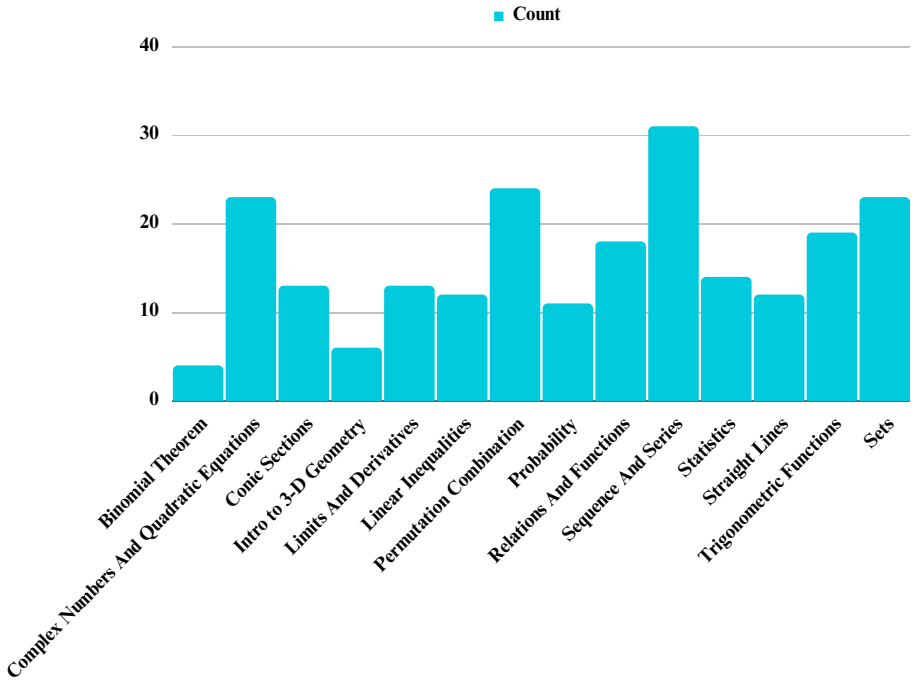

Figure 3: Distribution of Count of Samples of each Concept

WizardMath [19] introduces an innovative approach known as Reinforcement Learning from Evol-Instruct Feedback (RLEIF). This method combines Evol-Instruct and reinforced process supervision techniques to evolve GSM8k and MATH datasets. Subsequently, it fine-tunes the pre-trained LLaMA-2 model using the evolved data and reward models, resulting in the development of the WizardMath model.

Lastly, the MAmmoTH [37] models are trained using the MathInstruct dataset, meticulously curated for instructional tuning. MathInstruct is constructed from a compilation of 13 mathematical datasets, including six newly curated rationales. It encompasses a hybrid of chain-of-thought (CoT) and program-of-thought (PoT) rationales, ensuring comprehensive coverage of diverse mathematical domains. The entire fine-tuning process is outlined in Figure. 1.

Model	**# of Params**	**Accuracy**					
		GSM-8K	**DeepMind**	**NumGLUE**	**SimulEq**	**Math-401***	**MathQuest**
LLaMA-2	7B	16.0	46.0	37.0	11.0	10.0	10.4
LLaMA-2	13B	22.0	50.0	42.0	15.0	10.0	14.1
WizardMath	7B	61.0	51.0	54.0	27.0	6.0	14.6
WizardMath	13B	65.0	55.0	70.0	36.0	8.0	14.3
MAmmoTH	7B	43.0	49.0	54.0	23.0	11.0	12.2
MAmmoTH	13B	44.0	48.0	56.0	26.0	14.0	18.1

Table 2: Exact Match Accuracy results on the set of 100 samples of 5 datasets and our dataset MathQuest **Before** fine-tuning on Math-401 dataset. (*) refers to the set of Math-401 we augmented for fine-tuning.

5 Experiments

In this section, we delve into the details of our conducted experiments, outlining the experimental setup and the utilized hyper-parameters. Our research objective revolves around the creation of a

high school-level mathematical dataset, encompassing questions of varying complexities and diverse concepts, followed by the establishment of robust baselines for solving mathematical problems.

To achieve this, we conducted experiments involving three prominent large language models: LLaMA-2 [27], WizardMath [37]. We performed these experiments on both the 7B and 13B variants of these large language models (LLMs). Our experiments were executed in two stages. In the first stage, we directly loaded the original model weights and carried out inference on our designated test set. In the second stage, we undertook the fine-tuning of these models using the Math-401 [36] dataset as a crucial step in the process.

The Math-401 [36] dataset initially comprised 401 elementary mathematical equations paired with their corresponding results. To enhance its comprehensiveness and diversity, we performed data augmentation by introducing more intricate equations involving operators such as addition ($+$), subtraction ($-$), multiplication ($*$), division ($/$), as well as parentheses ($()$). This augmentation process aimed to create a more generalized and versatile dataset. Subsequently, we proceeded to fine-tune the Large Language Models (LLMs) using this augmented Math-401 [36] dataset.

Model	**# of Params**	**Accuracy**					
		GSM-8K	**DeepMind**	**NumGLUE**	**SimulEq**	**Math-401***	**MathQuest**
LLaMA-2	7B	30.0	46.0	45.0	15.0	17.0	10.6
LLaMA-2	13B	42.0	51.0	54.0	16.0	24.0	20.3
WizardMath	7B	64.0	55.0	52.0	29.0	15.0	16.01
WizardMath	13B	68.0	56.0	70.0	38.0	10.0	20.1
MAmmoTH	7B	56.0	50.0	62.0	24.0	16.0	18.5
MAmmoTH	13B	67.0	51.0	64.0	34.0	18.0	**24.0**

Table 3: Exact Match Accuracy Results on the set of 100 samples of 5 datasets and our dataset MathQuest **After** fine-tuning on Math-401 dataset. (*) refers to the set of Math-401 we augmented for fine-tuning.

The dataset was split into training (241,600 samples), validation (30,200 samples), and test (30,200 samples) subsets. We used the AdamW optimizer, a well-recognized technique, to enhance model performance. This optimization step was crucial for achieving the results in our study.

For fine-tuning, we employed QLora [5], an efficient approach that maximizes memory efficiency and minimize computation cost using 4-bit quantization in a pretrained language model, resulting in Low Rank Adapters (LoRA). Each model underwent 10 epochs of fine-tuning with a learning rate of 3×10^{-4}. Post fine-tuning, we assessed the models using the same test set employed for pre-fine-tuning inference. The results, summarized in Table. 3, serve to highlight the enhancements achieved in mathematical problem-solving capabilities before and after fine-tuning.

5.1 Evaluation Metric

We compared all model variants to evaluate the quality of the generated solutions. To measure performance, we assessed the accuracy in matching the generated answers to the actual solutions for five open-source datasets: GSM-8K, DeepMind, SimulEq, NumGLUE, and Math-401. These datasets provide ground truth answers for exact match accuracy calculation.

6 Results & Discussion

In this section, we present the outcomes of our experiments in the domain of mathematical problem-solving. Our study encompasses evaluations conducted on our proprietary dataset, MathQuest, as well as five other publicly available datasets. This paper establishes baseline performance metrics for the task using our MathQuest dataset. To gauge the effectiveness of Large Language Models (LLMs) across diverse datasets, we utilize exact match accuracy as a benchmark metric.

We organize our results into two distinct setups: **before fine-tuning** and **after fine-tuning** the models, with the primary aim of evaluating the model's learning capabilities. Table. 2 presents the exact

match accuracy of three models across two variants, 7B and 13B, before fine-tuning, on five datasets and our dataset MathQuest. To summarize these findings, referring to Table. 2, the performance of all the models is notably lower on the SimulEq dataset, as well as on our augmented dataset, Math-401. This discrepancy can be attributed to the presence of intricate problems within these datasets, which often require additional knowledge, such as questions like "Number of red color cards in a deck of 52 cards." Consequently, Table.3 provides a detailed overview of the accuracy results following the fine-tuning process. In summary, the accuracy of all models showed significant improvement after undergoing fine-tuning on our diverse and complex question-answer dataset. Notably, models with 13B parameters exhibited higher accuracy compared to those with 7B parameters.

The key takeaways from Table. 2, and Table. 3 reveal that the best-performing model is **MAmmoTH-13B** for our dataset MathQuest, exhibiting the highest accuracy among all models after fine-tuning, at 24.0%. Additionally, it's noteworthy that both MAmmoTH 7B and 13B generated outputs with precision up to two decimal places, indicating their accuracy. From Table 3, It is evident that our dataset, MathQuest, poses a greater challenge due to its complexity and diversity, resulting in lower accuracy compared to other datasets.

7 Conclusion

In summary, our approach enhances Large Language Models (LLMs) in acquiring vital reasoning skills for precise mathematical problem-solving. We introduce tailored question-answer pairs in our **MathQuest** dataset, encompassing single or multiple mathematical operators and expressions. These supportive simple and complex problems guide the model toward incremental problem-solving. Our primary aim is to provide illustrative examples that improve solution accuracy and clarity. Our results demonstrate significant enhancements in both solution precision and comprehensibility, promising valuable support for educators and students seeking effective mathematical problem-solving capabilities.

While our research establishes a robust foundation for advancing mathematical problem-solving with Generative LLMs, further refinements and optimizations are essential to extend its applicability across a broader range of scenarios. Ultimately, our work contributes to advancing conceptual understanding and numerical problem-solving in high school-level mathematical question-answering, offering valuable assistance to students and professionals grappling with complex questions through LLMs.

8 Limitations

While our proposed solution can successfully solve basic mathematical problems, it occasionally encounters challenges when dealing with complex mathematical problems that involve retaining variable values for use in subsequent equations.

Another limitation of our proposed work is the partial enhancement of reasoning abilities in LLMs for solving mathematical problems. However, it still falls short in dealing with complex expressions that include nested brackets within equations. The reason could be limited training dataset size, we will try to increase our training data in future research. We intend to address this limitation in our future work, wherein we plan to incorporate recent prompting techniques and further enhance LLMs reasoning abilities for these types of problems.

9 Acknowledgement

Dr. Rajiv Ratn Shah is partly supported by the Infosys Center for AI, the Center of Design and New Media, and the Center of Excellence in Healthcare at Indraprastha Institute of Information Technology, Delhi. We gratefully thank Dr. Astha Verma and Mr. Naman Lal for their guidance and continuous support during our research. Their knowledge and insightful feedback significantly influenced the direction and quality of our research. We appreciate their time, devotion, and willingness to share information, which all contributed considerably to the accomplishment of this job. Their encouragement and constructive talks were a continual source of motivation for us, and we consider ourselves fortunate to have benefited from their wisdom and leadership.

References

[1] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2022.

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[6] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, Roman Wang, Nikhil Singh, Taylor L. Patti, Jayson Lynch, Avi Shporer, Nakul Verma, Eugene Wu, and Gilbert Strang. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32), aug 2022.

[7] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023.

[8] Muhammad Usman Hadi, Qasem Al-Tashi, Rizwan Qureshi, Abbas Shah, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Shaikh, Naveed Akhtar, Jia Wu, and Seyedali Mirjalili. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects, 07 2023.

[9] Joy He-Yueya, Gabriel Poesia, Rose E. Wang, and Noah D. Goodman. Solving math word problems by combining language models with symbolic solvers, 2023.

[10] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[11] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models, 2023.

[12] Jaeho Jeon and Seongyong Lee. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt - education and information technologies, May 2023.

[13] Shing-Yun Jung, Ting-Han Lin, Chia-Hung Liao, Shyan-Ming Yuan, and Chuen-Tsai Sun. Intent-controllable citation text generation. *Mathematics*, 10:1763, 05 2022.

[14] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023.

[15] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics.

[16] Frank Lester. Thoughts about research on mathematical problem- solving instruction. *Mathematics Enthusiast*, 10:245–278, 01 2013.

[17] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022.

[18] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier, 2023.

[19] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2023.

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

[21] Silvia Milano, Joshua A McGrane, and Sabina Leonelli. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334, 2023.

[22] Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[23] Steven Moore, Richard Tong, Anjali Singh, Zitao Liu, Xiangen Hu, Yu Lu, Joleen Liang, Chen Cao, Hassan Khosravi, Paul Denny, et al. Empowering education with llms-the next-gen interface and content generation. In *International Conference on Artificial Intelligence in Education*, pages 32–37. Springer, 2023.

[24] Matteo Muffo, Aldo Cocco, and Enrico Bertino. Evaluating transformer language models on arithmetic operations using number decomposition, 2023.

[25] Anastasia Olga, Akash Saini, Gabriela Zapata, Duane Searsmith, Bill Cope, Mary Kalantzis, Vania Castro, Theodora Kourkoulou, John Jones, Rodrigo Abrantes da Silva, et al. Generative ai: Implications and applications for education. *arXiv preprint arXiv:2305.07605*, 2023.

[26] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models, 2019.

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut

Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[29] Cunxiang Wang, Boyuan Zheng, Yuchen Niu, and Yue Zhang. Exploring generalization ability of pretrained language models on arithmetic and logical reasoning, 2021.

[30] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[32] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification, 2023.

[33] Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. An empirical study on challenging math problem solving with gpt-4, 2023.

[34] Le Xiao and Xiaolin Chen. Enhancing llm with evolutionary fine tuning for news summary generation, 2023.

[35] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. Practical and ethical challenges of large language models in education: A systematic literature review. *arXiv preprint arXiv:2303.13379*, 2023.

[36] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. How well do large language models perform in arithmetic tasks?, 2023.

[37] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning, 2023.

[38] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023.