2023 International Conference on Intelligent Education and Intelligent Research (IEIR) | 979-8-3503-4289-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEIR59294.2023.10391224

Evaluation of Large Scale Language Models on Solving Math Word Problems with Difficulty Grading

1st Xin He
Faculty of Artificial Intelligence in Education
Central China Normal University
Wu Han,430070,China
xinhe@mails.ccnu.edu.cn

2nd Huikai Gao
Faculty of Artificial Intelligence in Education
Central China Normal University
Wu Han,430070,China
maxkrsz@mails.ccnu.edu.cn

3rd Jiaxing He
Faculty of Artificial Intelligence in Education
Central China Normal University
Wu Han,430070,China
h2021114247@mails.ccnu.edu.cn

4th Chao Sun*
Faculty of Artificial Intelligence in Education
Central China Normal University
Wu Han,430070,China
csun@ccnu.edu.cn

***Abstract*—With the advancement of artificial intelligence, large language models have achieved remarkable success, particularly in natural language processing. However, their performance in solving Math Word Problems (MWPs) has been challenged by the difficulty of these problems, especially in high-difficulty scenarios. This research aims to evaluate the answer accuracy of large language models, taking GPT-3.5 as an example, in MWP tasks, focusing on graded difficulty levels of the problems. We transform the existing math23K dataset into a math application problem dataset with multiple difficulty levels, aiming to reflect the diverse complexities of real-world application scenarios. Through systematic experimental analysis, we delve into the performance of these models on problems of varying difficulty and reveal the relationship between model performance and problem difficulty. Experimental results indicate that these large language models excel in solving simple and moderately low-difficulty problems but exhibit a general decline in performance when faced with moderately high and complicated problems. Furthermore, our study unveils that models are more susceptible to the complexity and ambiguity of problem descriptions, particularly in high-difficulty scenarios.**

***Index Terms*—Large Language Models; GPT; Math Word Problem; Automatic Answering**

I. Introduction

Natural Language Processing (NLP) is a critical branch of artificial intelligence dedicated to enabling computers to understand, process, and generate human-readable language. In recent years, with the rapid advancement of deep learning techniques, Large Language Models (LLMs) [1] have emerged, bringing significant breakthroughs and innovations to the field of NLP. Large language models are powerful tools built on deep learning technology, equipped with the capability to improve a computer's understanding and processing of natural language. Their core idea involves pre-training on massive amounts of textual data, allowing the model to acquire rich language knowledge and context comprehension. This pre-trained process equips the model with a general language understanding capability and subsequent fine-tuning can further enhance its performance for specific tasks.

The development history of large language models can be traced back to early statistical language models and neural network language models. The evolution of these models in the field of natural language processing has had a significant impact on today's language technologies and natural language generation tasks. Statistical language models [2] were one of the early methods used for processing natural language text. They employed statistical methods to estimate the probability relationships between words. These models provided a foundational framework for language generation, text classification, and information retrieval. However, traditional statistical language models faced challenges in modeling long-distance dependencies and handling large-scale corpora. With the rise of deep learning, neural network language models emerged. These models used neural networks, especially recurrent neural networks [3] (RNNs) and word embeddings, to model language sequences. This enabled the models to better handle long-distance dependencies, leading to improved language model performance. The real breakthrough in large language models came with the introduction of the Transformer model [4]. The Transformer introduced the self-attention mechanism, allowing the model to capture dependencies within sequences more effectively while also increasing parallelism. This in-

*Corresponding author

979-8-3503-4289-5/23/$31.00 ©2023 IEEE

novation led to significant success in tasks like machine translation and became the foundation for subsequent large language models.

In 2018, OpenAI released the Generative Pre-trained Transformer (GPT) [5], a large language model based on the Transformer architecture. GPT employed a pre-training and fine-tuning approach, enabling it to generate fluent and coherent text while excelling in various natural language processing tasks. This marked the rise of large language models, ushering in a new era in natural language processing.With the emergence of GPT, major companies invested in research and development efforts and introduced their own large language models. For example, Baidu introduced Wenxin Yiyuan, Google released PaLM, Alibaba Cloud launched Tongyi Qianwen, and MiniMax developed MIMO, among others. These models have played crucial roles in their respective domains and applications, driving continuous advancements in natural language processing technology.

Automated mathematical problem-solving has long been a highly regarded research topic within the field of machine learning. The field has made significant strides with the advent of large language models. Many individuals have conducted corresponding tests to evaluate their ability to solve mathematical problems [6]. The emergence of large language models has brought new possibilities for machines to address mathematical questions, sparking extensive research and discussions. Many researchers have extensively tested the capabilities of these models in solving mathematical problems, with the aim of exploring their potential in the field of mathematics. Some of these studies have indicated that large language models exhibit remarkable zero-shot accuracy in many mathematical problems [7].

For large language models solving mathematical problems, researchers have introduced two criteria for evaluating their answer outputs: verifiability and conciseness [8]. Accuracy in addressing mathematical problems has consistently been a focal point for researchers, emphasizing the correctness and readability of the answers still awaiting exploration and study. While large language models have demonstrated potential in solving mathematical problems, their accuracy remains a central focus for researchers. The correctness of answers and the readability of responses require further research and exploration.

Furthermore, researchers are actively exploring methods to further enhance large language models, aiming to improve their performance in solving mathematical problems and other domains. This may involve innovations in better pre-training methods, richer embeddings of mathematical knowledge, more effective mathematical reasoning techniques, and other areas. Thus, the development of large language models in the field of mathematical problem-solving remains a challenging yet promising area with opportunities for advancement in addressing a wide range of application scenarios.

The main contributions of this paper are as follows:

1) Testing the ability of GPT to solve MWP of different difficulty levels.
2) Summarizing the current performance of GPT in solving MWP.
3) Analyzing the primary reasons for deficiencies in GPT's MWP solving capabilities.

II. Related Work

Large language models like GPT, with billions or even tens of trillions of parameters, represent outstanding achievements in the field of deep learning. Through extensive pre-training on massive text data, they possess remarkable text generation and language understanding capabilities. To validate their potential, researchers have conducted extensive testing and evaluation in various domains, including text generation quality, language understanding tasks, knowledge and commonsense understanding, fake news detection, interpretability, multilingual support, and transfer learning. These studies not only contribute to model improvement but also drive their application and development across a wide range of use cases.

For instance, Tom B. Brown et al. (Language Models are Few-Shot Learners) [9] conducted comprehensive tests on GPT, assessing its logical reasoning abilities, robustness, and more. While they examined various aspects of GPT's performance, they did not delve deeply into its ability to solve problems, including mathematical problems.

To evaluate GPT's problem-solving abilities, Vedant Gaur et al. (Reasoning in Large Language Models Through Symbolic Math Word Problems) [8] introduced two criteria for assessing the correctness and conciseness of GPT's answer outputs. The article explored the normativity of GPT's outputs under these criteria and discussed methods for optimizing GPT's output in this context.

In summary, research on GPT and other large language models' abilities to solve Math Word Problems (MWPs) is still an evolving field [6] [10]. As large language models continue to develop, research in the domain of MWPs is expected to make further breakthroughs and enter new phases of advancement.

III. Methodology

In order to systematically evaluate the performance of large language models (primarily GPT-3.5) on the Math Word Problems dataset, math23k [11], we adopted a difficulty-stratified approach. Specifically, we categorized the 3385 data points into five difficulty levels, corresponding to Level 1, Level 2, Level 3, Level 4 and Level 5 as outlined in Table 1 below. The purpose of this step was to construct a dataset containing questions of varying difficulty levels.

TABLE I
THE ACCURACY OF GPT IN SOLVING DIFFICULTY-TIERED QUESTIONS

Dataset	Level 1	Level 2	Level 3	Level 4	Level 5
Difficulty Value	0-0.1	0.1-0.15	0.15-0.2	0.2-0.4	0.4-0.5
Size	582	1351	569	838	46

TABLE I categorizes the dataset into five difficulty levels based on the difficulty coefficient of each problem, labeled as

level 1 to level 5. Each difficulty level encompasses a diverse set of mathematical problems, covering various aspects of the mathematical domain. In this paper, we meticulously selected a specific number of problem samples for experimental research corresponding to these five distinct difficulty levels.

Subsequently, we conducted tests on various large language models designed for Math Word Problems using this difficulty-stratified dataset [12]. We determined the accuracy of answering these questions by comparing the experimental data's answers to the standard answers [13]. The comparison of accuracy not only helped us assess the performance of large language models in Math Word Problems but also allowed us to evaluate their abilities in logical reasoning and language comprehension.

This approach was instrumental in providing a more accurate evaluation of model performance across different difficulty levels. Ultimately, our research findings will offer valuable insights into understanding how large language models perform when confronted with problems of varying difficulty. They will also serve as a valuable reference for future improvements and optimizations of these models.

IV. Experiments

This research's experimental design aims to evaluate the answer accuracy of large language models in solving Math Word Problems (MWP), with a particular focus on the difficulty stratification of the dataset. We have taken a series of measures to gain a deeper understanding of the performance of the large language model GPT-3.5 when handling MWP of varying difficulty levels. The main steps and methods of the experiment are as follows:

A. Dataset Construction

We selected 3385 mathematical problems from the math23k dataset to construct a multi-difficulty-level mathematical application problem dataset, encompassing five difficulty tiers: easy, low, moderate, high-medium, and high. This dataset comprises problem samples from various sources, including educational materials, standardized tests, and mathematical competitions, with each problem accompanied by a reference answer. We conducted meticulous screening and annotation to ensure the diversity and representativeness of the problems, thus guaranteeing the accuracy of our evaluation. Table II consists of sample questions for each difficulty level.

B. Data Preprocessing

We conducted preprocessing on the collected problems, including tasks like text cleaning, tokenization, and stemming, to ensure the consistency and usability of the problem text. Furthermore, we categorized these MWPs into different difficulty levels, dividing them into five distinct levels: simple, low-medium, medium, high-medium, and high. This stratification process was designed to ensure that we have a multi-tiered evaluation system, covering various problems from easy to complex.

The specific procedure for difficulty grading is as follows: We employ various operations to establish different indicators for each mathematical question and use these indicators to assess the difficulty level of each question. Different difficulty coefficient ranges correspond to different difficulty levels. Referring to TABLE I, questions with difficulty coefficients between 0 and 0.10 are classified as Level 1, questions with difficulty coefficients between 0.10 and 0.15 are classified as Level 2, questions with difficulty coefficients between 0.15 and 0.20 are classified as Level 3, questions with difficulty coefficients between 0.20 and 0.40 are classified as Level 4, and questions with difficulty coefficients between 0.40 and 0.5 are classified as Level 5.

C. Model Selection for MWP Evaluation

In order to comprehensively assess the performance of large language models in the Math Word Problems (MWP) task, we have meticulously selected three leading models: GPT-3.5 [9], MWPBERT [14], and Graph2Tree [15]. These models represent different natural language processing techniques and architectures. Each of these models has undergone pre-training and fine-tuning to adapt to the requirements of solving mathematical problems. Through this diverse model selection, we are able to conduct a comprehensive evaluation of their strengths in handling mathematical problems of various difficulty levels, providing deeper insights and understanding for addressing complex mathematical challenges in the future.

D. Experimental Setup

We conducted experiments with different large language models for each difficulty level. For each Math Word Problem (MWP) [16], we used the problem description as input and instructed the large language model to generate an answer. Subsequently, we compared the generated answer with the pre-annotated correct answer. We divided the experiments into sub-experiments corresponding to different difficulty levels, evaluating each model's performance on easy, moderate, moderately challenging, and highly challenging problems, respectively. Each sub-experiment included a set of problem samples with relatively uniform difficulty levels to ensure fairness and comparability of the experimental results.

E. Performance Evaluation

We employed accuracy as the performance metric to evaluate the accuracy of answers generated by the three large language models across various difficulty levels. Accuracy reflects the proportion of generated answers that match the correct answers. We conducted a thorough analysis of the answer accuracy of the models at different difficulty levels.The table below presents the accuracy scores for each model.

As shown in Table III, we tested the question making performance of GPT using MWP datasets that have been difficulty graded, following up on what was said above, the difficulty is categorized into one to five grades, where grade one has 581 questions, grade two has 1351 questions, grade three has 569 questions, grade four has 838 questions, and

TABLE II
EXAMPLE QUESTIONS WITH DIFFICULTY STRATIFICATION IN MWP

Difficulty Levels	Example Questions	Answer
Level 1	In 4 days, a total of 212 trees were planted. What is the approximate number of trees planted per day on average?	53
Level 2	The school is planning to build a wall. On the first day, the workers delivered 4560 bricks, which is 960 more than the second day. How many bricks were delivered in total over the two days?	8160
Level 3	Mary brought some money to buy exercise books. If each exercise book costs 0.3 yuan, she can buy 24 books. If she chooses another type of exercise book that costs an additional 0.1 yuan per book, how many books can she buy with the money she has?	18
Level 4	A year ago, Anna deposited her pocket money into a bank with an annual interest rate of 3.78%. After one year, she had a total of 207.56 yuan, including the principal and interest. How much money did Anna deposit in the bank one year ago?	200
Level 5	In a box, there are an unknown number of red and white balls. If you draw 1 red ball and 1 white ball at a time, and when there are no more red balls left, there are 50 white balls remaining. If you draw 1 red ball and 3 white balls at a time, and when there are no more white balls left, there are 50 red balls remaining. How many red and white balls were originally in the box?	250

grade five has 46 questions. We also used two other MWP solvers, MWPBERT and Graph2Tree, to solve the difficulty graded dataset and get the accuracy rate for comparison experiments.

Overall, we can see that GPT has a higher accuracy compared to MWPBERT and Graph2Tree, which indicates the advantage of large language models in solving mathematical problems. Compared to other solvers, large language models have a better ability to comprehend mathematical questions and provide scientifically logical solutions. However, it can also be observed that the current accuracy of large language models still falls short of being fully reliable [17]. Therefore, there is still a significant need for further research on the application of large language models in solving MWP.

TABLE III
THE ACCURACY OF GPT IN SOLVING DIFFICULTY-TIERED QUESTIONS

Dataset	Level 1	Level 2	Level 3	Level 4	Level 5
Size	582	1351	569	838	46
GPT-3.5	0.8314	0.6972	0.5533	0.3213	0.1556
MWPBERT	0.8	0.758	0.55	0.27	0.065
Graph2Tree	0.747	0.682	0.504	0.198	0.02

F. Experiment Results

n this study, we observed a decrease in the accuracy of large language models when answering difficult math application questions. However, comparing the other two solver models, the accuracy rate is almost equal at low-difficulty questions, although the accuracy rate is lower than that of MWPBERT when answering level 2 questions, which we believe may be due to the fact that the GPT's understanding of the Chinese semantics is likely to be ambiguous, especially since level 2 is the level with the most number of questions in the dataset, with 1,351 questions, which may amplify this problem. However, when it comes to the more difficult questions, although the accuracy rates are not ideal, it is clear that the large language model has a higher accuracy rate, which signals that the large language model has a great potential for answering more difficult math questions. In order to gain a deeper understanding of these performance differences, we conducted further analysis to reveal the underlying causes of these differences. We found that in high-difficulty problems, the model is more susceptible to the complexity and ambiguity of the problem description. This means that the problem formulation may contain more implicit information and semantic ambiguity, posing greater challenges for the model to accurately understand and transform the problem.

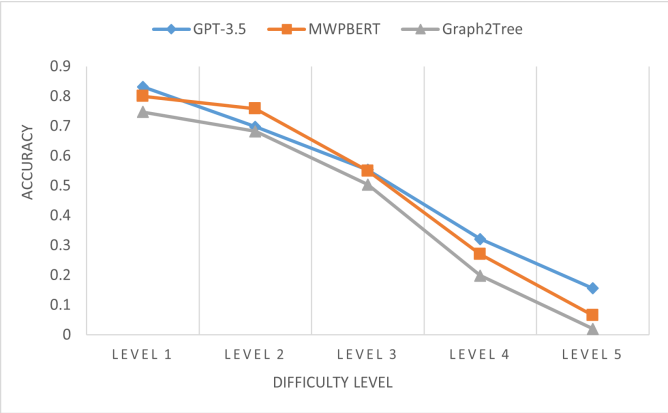

Fig. 1. Elliptic Paraboloid

V. CONCLUSION

Based on the experimental results, we can observe that large language models exhibit impressive performance in the field of Math Word Problems (MWP), significantly when solving simple and moderately low-difficulty problems. However, for moderately high and high-difficulty problems, the models' performance tends to be poorer, with a noticeable decrease

in answer accuracy. This results suggests that there are limitations in the models' capabilities when dealing with complex problems.

This situation primarily stems from the complexity of high-difficulty problems, where problem descriptions may involve implicit information, nested semantic structures, and other intricate elements, placing higher demands on the models' deep understanding and accurate extraction of crucial information. Additionally, high-difficulty problems often come with semantic ambiguity, forcing models to choose among multiple interpretations without sufficient context to definitively determine the correct answer. Insufficient domain knowledge and data imbalances in training data for high-difficulty problems may also cause a decline in performance.

Therefore, further research and improvements in handling the complexity and ambiguity of problem descriptions are needed, including the integration of richer domain knowledge and more balanced training data. These efforts will help enhance the performance of large language models on high-difficulty mathematical application problems. This issue also reminds us to exercise caution when using these models in practical education and applications, especially when dealing with high-difficulty mathematical application problems.

VI. Future

Currently, GPT models may not be able to provide accurate answers to complex mathematical problems because they are trained based on large-scale textual data [18], and mathematical problems usually require specialized mathematical knowledge and reasoning ability to solve them.GPT is currently more adept at natural language processing and has some shortcomings in expressing mathematical symbols.

From our research, it is evident that GPT may struggle to comprehend ambiguous questions and lacks the ability to self-correct errors in certain problems, leading to a cascade of incorrect responses. Currently, GPT achieves high accuracy in solving simple mathematical problems, but it falls short when faced with more challenging ones. Moreover, even in solving relatively easier mathematical problems, GPT can produce incorrect answers due to various factors such as failure to understand or misinterpret the question, or errors in the application of mathematical formulas. Therefore, at present, GPT can only be considered a relatively better MWP solver, and further research is needed to delve deeper into this area.

GPT is currently better at natural language processing and has some shortcomings in expressing mathematical notation.However, with the popularization of large language models, GPT and even large language models will develop gradually. In the future, GPT should be able to solve math problems with more readable answers, with smooth logic and no errors in mathematical notation. Of course, the most important thing is to achieve almost correct answer results.

VII. Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant 62177025.

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models."

[2] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems*, vol. 13, 2000.

[3] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[5] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[6] S. Frieder, L. Pinchetti, A. Chevalier, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, and J. Berner, "Mathematical capabilities of ChatGPT."

[7] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners."

[8] V. Gaur and N. Saunshi, "Reasoning in large language models through symbolic math word problems."

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] P. Shakarian, A. Koyyalamudi, N. Ngu, and L. Mareedu, "An independent evaluation of ChatGPT on mathematical word problems (MWP)."

[11] Y. Wang, X. Liu, and S. Shi, "Deep neural solver for math word problems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 845–854, Association for Computational Linguistics.

[12] Z. Liang, J. Zhang, and X. Zhang, "Analogical math word problems solving with enhanced problem-solution association."

[13] Z. Yuan, H. Yuan, C. Tan, W. Wang, and S. Huang, "How well do large language models perform in arithmetic tasks?."

[14] Z. Liang, J. Zhang, L. Wang, W. Qin, Y. Lan, J. Shao, and X. Zhang, "MWP-BERT: Numeracy-augmented pre-training for math word problem solving."

[15] S. Li, L. Wu, S. Feng, F. Xu, F. Xu, and S. Zhong, "Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem."

[16] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. v. d. Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models."

[17] D. Zhang, L. Wang, L. Zhang, B. T. Dai, and H. T. Shen, "The gap of semantic parsing: A survey on automatic math word problem solvers," vol. 42, no. 9, pp. 2287–2305.

[18] S. Zheng, Y. Zhang, Y. Zhu, C. Xi, P. Gao, X. Zhou, and K. C.-C. Chang, "GPT-fathom: Benchmarking large language models to decipher the evolutionary path towards GPT-4 and beyond."