Improving Bengali Math Solutions with Multi-Agent TIR

1st Abir Ahmed
Dept. of Software Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
abirahmedsohan6@gmail.com

2nd Farzine
Dept. of Software Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
farzine07@student.sust.edu

3rd Siam Arefin
Dept. of Software Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
siamarefin2000@gmail.com

4th Tahera Jannat Mitu
Dept. of Software Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
taherajannat1971@gmail.com

***Abstract*—Mathematical problem-solving using large language models (LLMs) has seen significant advancements, yet there remains a scarcity of models focused on Bangla language math problems. To address this gap, we developed a solution using the Qwen/Qwen2.5-32B-Instruct-GPTQ-Int4 model, integrating Tool-Integrated Reasoning (TIR) prompting with a multi-agent framework specifically adapted for Bangla. Our work is based on structured problem-solving by decomposing complex mathematical questions into intermediate Python code, improving the model's accuracy and efficiency in a low-resource language context. Testing on a diverse set of 100 math problems yielded an 81% success rate, demonstrating the model's robustness and effectiveness in Bangla mathematical reasoning. This work not only advances AI capabilities in the Bangla language but also provides valuable resources for the Bangla AI community, encouraging further research and development in this underrepresented area.**

***Index Terms*—Tool-Integrated Reasoning (TIR), Qwen-32B model, Bangla math solver**

I. Introduction

Artificial Intelligence (AI) has transitioned from a speculative concept to a transformative force across various domains. A critical aspect of AI's evolution is its capacity for complex problem-solving, particularly in mathematical reasoning. While significant progress has been made in languages with abundant resources, there remains a notable gap in AI's ability to tackle mathematical challenges in low-resource languages such as Bengali. This limitation hinders the development of AI tools that can assist Bengali-speaking students and researchers in comprehending and solving intricate mathematical problems. Bengali is one of the world's most widely spoken languages, with approximately 228 million native speakers and an additional 37 million second-language speakers.

In this study, we address this gap by developing an AI model capable of solving mathematical problems presented in Bengali. Our approach involves curating a comprehensive dataset of Bengali math problems, implementing advanced natural language processing techniques to interpret and process the language, and employing sophisticated machine learning algorithms to solve the problems accurately. We also incorporate a feedback loop to refine the model's performance iteratively.

By establishing benchmarks in complex mathematical reasoning within the Bengali language, our work aims to enhance AI's problem-solving capabilities and provide valuable educational tools for the Bengali-speaking community. This endeavor contributes to the field of AI and promotes inclusivity in academic resources, ensuring that technological advancements benefit a broader spectrum of learners.

II. Related Work

The development of AI models capable of solving mathematical problems has seen significant advancements, with recent methodologies focusing on combining structured reasoning and code execution to improve problem-solving capabilities. NuminaMath 7B TIR [1], for instance, is a language model fine-tuned specifically to tackle complex mathematical tasks. This model uses a two-stage training approach, beginning with supervised fine-tuning on a dataset of math problems paired with solutions that employ Chain of Thought (CoT) prompting [2]. CoT enables the model to reason through multi-step solutions, promoting logical consistency and enhancing interpretability. The second fine-tuning stage involves training on a synthetic dataset that emphasizes Tool-Integrated Reasoning (TIR), where each problem is decomposed into a series of rationales, executable Python code, and outputs. This TIR approach enables the model to perform intermediate calculations, improving its accuracy and ability to handle complex mathematical reasoning. Despite its advantages, the model has limitations in tackling highly advanced problems, particularly in geometry, due to its lack of visual processing capabilities.

Another notable model in this area is DeepSeekMath 7B [3], which pushes the boundaries of mathematical reasoning in

open-source AI models by focusing on extensive pre-training using a custom-built dataset specifically curated for mathematics. Known as the DeepSeekMath Corpus, this dataset comprises over 120 billion tokens selected for mathematical content from Common Crawl data. To enhance the model's mathematical reasoning, DeepSeekMath introduces a novel reinforcement learning technique called Group Relative Policy Optimization (GRPO), which is a variant of Proximal Policy Optimization (PPO). GRPO reduces computational requirements by eliminating the need for a critic model, making training more efficient and memory-friendly. This combination of extensive pre-training and GRPO has enabled DeepSeekMath to demonstrate robust performance on a variety of mathematical benchmarks, approaching the capabilities of larger, proprietary models in complex mathematical tasks.

The Qwen2.5-72B-Instruct [4] model represents another significant advancement, designed with a large parameter count to handle complex tasks in mathematics and coding. Instruction-tuned and equipped with enhanced adherence to expert-level instructions, Qwen2.5-72B can manage extended contexts up to 131,072 tokens while generating long outputs (up to 8,192 tokens). Its methodology integrates specialized architectures, such as RoPE and SwiGLU, and is highly effective in managing structured data, including tables and JSON outputs. This model also supports 29 languages, enhancing its applicability across linguistic contexts, and demonstrates considerable capabilities in both language understanding and mathematical problem-solving, especially for multilingual applications.

The Deep Neural Solver for Math Word Problems [5], developed by Tencent AI Lab, employs a unique hybrid approach to address the challenges of math word problem-solving. The model combines a sequence-to-sequence (seq2seq) neural network with a similarity-based retrieval model. The seq2seq component directly translates word problems into equation templates, allowing the model to generate solutions without requiring feature engineering. The retrieval model complements this approach by identifying problems in the training data that are similar to the test problem and applying their equation templates, thereby enhancing the model's accuracy. By dynamically choosing between the seq2seq and retrieval methods based on similarity scores, the hybrid design improves overall performance, particularly on problems that closely resemble those in the training data.

III. Methodology

A. Dataset

This work utilizes two key datasets, **NuminaMath-CoT** and **NuminaMath-TIR** [6], each designed to support structured mathematical reasoning. Both datasets are available under the Creative Commons NonCommercial (CC BY-NC 4.0) license.

- **NuminaMath-CoT**: The NuminaMath-CoT dataset includes approximately 860,000 math problems, each presented in a Chain of Thought (CoT) format. Problems are sourced from diverse backgrounds, such as high school math exercises, international math olympiads, and synthetic datasets. The solutions are formatted to provide a step-by-step reasoning path, enabling models to break down complex problems through sequential steps.
- **NuminaMath-TIR**: NuminaMath-TIR contains about 72,540 problems focused on Tool-Integrated Reasoning (TIR), where solutions are developed using Python libraries like `SymPy` and `NumPy`. This dataset emphasizes a code-executed reasoning approach. Solutions were generated through an iterative process using GPT-4, with each answer validated against a reference to ensure accuracy.

To broaden the dataset's usability, we also experiment with **translated Bangla versions** of both datasets, allowing for application in low-resource language contexts.

B. Model Selection

In developing an effective solution for Bengali mathematical problem-solving, we conducted a series of experiments across multiple models and strategies. Each model was tested with a variety of approaches, including Chain of Thought (CoT) and multi-agent Tool-Integrated Reasoning (TIR). We describe our model selection process and findings below.

1) Initial Experimentation with DeepSeek-AI: We began our experiments with the `deepseek-ai/deepseek-math` series of pre-trained models, evaluating their performance in both English and Bengali. The performance on Bengali data was poor, while results for English were moderate, indicating potential limitations of these models in handling Bengali mathematical contexts, possibly due to limited multilingual capabilities.

2) Transition to AI-MO/NuminaMath Variants: To improve performance, we next experimented with variants from the `AI-MO/NuminaMath` series. Under similar setups, these models showed an improvement in English problem-solving accuracy. The models demonstrated enhanced reasoning and interpretability for complex problems, suggesting that they might be better suited for structured mathematical datasets compared to the initial DeepSeek-AI models. Despite this, performance on Bengali problems remained suboptimal.

3) Enhanced Results with Qwen/Qwen2.5-Math: We achieved the most promising results with the `Qwen/Qwen2.5-Math` variant. This model exhibited significantly better performance on Bengali problems, likely owing to its multilingual training and advanced contextual understanding. The Qwen2.5-Math model's architecture and scale were also advantageous for handling the nuanced

Model	K	Depth	Tem	Dtype	Score
extbf{Qwen/Qwen2.5-32B-Instruct-GPTQ-Int4}	\textbf{16}	\textbf{6}	\textbf{0.8}	\textbf{full}	\textbf{81}
Qwen2.5-7B-Instruct	64	6	0.8	half	77
Qwen2.5-32B-Instruct-INT4	16	5	0.8	half	76
Qwen2.5-7B-Instruct	32	6	0.8	half	75
AI-MO/NuminaMath-7B-TIR-GPTQ	16	5	0.8	full	73
Qwen2.5-32B-Instruct-AWQ	16	5	0.8	half	72
Qwen2.5-Math-1.5B-Instruct	32	4	0.8	half	61
AI-MO/NuminaMath-7B-TIR	4	4	0.7	full	60
AI-MO/NuminaMath-7B-TIR	4	4	0.7	half	60
AI-MO/NuminaMath-7B-CoT	4	4	0.7	half	40

Fig. 1. Overview of experimented models and their hyperparameters.

language structure and mathematical complexity present in our dataset.

4) Methodological Approach: Chain of Thought vs. Multi-Agent TIR: For each model, we tested two primary reasoning approaches:

- **Chain of Thought (CoT)**: Here, we prompted models to generate step-by-step solutions independently.
- **Multi-Agent TIR**: This approach involved a multi-agent setup where agents collaborated to solve problems individually, with the final output selected through majority voting.

The multi-agent TIR approach consistently outperformed CoT across all models, enhancing problem-solving accuracy through explicit intermediate reasoning steps. We further experimented by incorporating CoT as additional context within the multi-agent TIR prompts. However, this hybrid approach resulted in a performance drop, suggesting that passing CoT as context may introduce noise or distract the model from the structured reasoning steps in TIR.

5) Context Generation through Vectorized Datasets: To enhance CoT relevance, we vectorized both the **NuminaMath-CoT** and **NuminaMath-TIR** datasets and used cosine similarity to identify and incorporate problem-specific contexts. Unfortunately, this context addition did not yield improved performance, as both individual CoT and context-based CoT diminished accuracy.

6) Final Model Configuration: Ultimately, we achieved the best results by focusing solely on multi-agent TIR without any problem-specific CoT context. This approach, combined with selective hyperparameter tuning, resulted in a significant improvement in Bengali problem-solving performance, providing a robust solution for our low-resource language setup.

IV. Results and Discussion

Our experiments demonstrate the Qwen-32B model's effectiveness in solving Bengali math problems, achieving high accuracy with structured reasoning. Table I summarizes the key results across various models and reasoning approaches.

The Multi-Agent TIR prompting approach outperformed Multi-Agent TIR with CoT context, achieving 81% accuracy with Qwen-32B due to its structured problem-solving and

TABLE I
PERFORMANCE SUMMARY

Model	Reasoning Approach	Accuracy (%)
Qwen/Qwen2.5-32B	**Multi-Agent TIR**	**81%**
Qwen/Qwen2.5-32B	Multi-Agent TIR + CoT	65%
AI-MO/NuminaMath-7B	Multi-Agent TIR	74%

consensus-driven approach. In contrast, adding CoT as an external context within TIR did not improve accuracy, suggesting that context-free TIR provides a more effective and streamlined solution.

Key Insights:

- **Model Performance:** Qwen-32B excelled in Bangla for both accuracy and reasoning depth, especially when applied with TIR, highlighting the importance of multilingual capabilities.
- **Reasoning Approach:** The Multi-Agent TIR setup improved accuracy by enabling focused problem decomposition and consensus voting, particularly effective in Bengali's low-resource language setting.
- **Computational Trade-off:** While TIR added minor computational overhead, the accuracy gains outweighed the increased response time.

These findings underscore the potential of structured reasoning approaches, like Multi-Agent TIR, in enhancing AI's ability to tackle complex mathematical problems in low-resource languages.

V. CONCLUSION

In this study, we investigated the performance of the Qwen-32B model for solving Bengali math problems. By utilizing the Multi-Agent TIR prompting approach, we achieved an impressive accuracy of 81%, outperforming all other approaches. The results highlight the effectiveness of structured problem-solving techniques in enhancing reasoning accuracy, particularly in a low-resource language like Bengali. Our findings demonstrate the model's potential to solve complex mathematical problems and contribute to the advancement of AI for languages with limited resources.

We plan to fine-tune the model in future work using a larger, more diverse Bengali dataset. This fine-tuning would further refine the model's ability to handle a broader range of math problems in Bengali, improving its robustness and accuracy. Additionally, exploring hybrid reasoning strategies combining Multi-Agent TIR prompting with other advanced techniques like Tree of Thought, Algorithm of Thought, etc could enhance problem-solving capabilities, particularly for more challenging mathematical tasks. Further experimentation with different pre-trained models and reasoning frameworks will also be crucial in optimizing performance.

REFERENCES

[1] E. Beeching, S. C. Huang, A. Jiang, J. Li, B. Lipkin, Z. Qina, K. Rasul, Z. Shen, R. Soletskyi, and L. Tunstall, "Numinamath 7b tir." https://huggingface.co/AI-MO/NuminaMath-7B-TIR, 2024.

[2] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv preprint arXiv:2210.03493*, 2022.

[3] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[4] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[5] 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, 2017.

[6] J. LI, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. C. Huang, K. Rasul, L. Yu, A. Jiang, Z. Shen, Z. Qin, B. Dong, L. Zhou, Y. Fleureau, G. Lample, and S. Polu, "Numinamath." [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.