FROM PERCENTAGES TO FINANCE: TRANSFER LEARNING OF MATHEMATICAL SKILLS IN LLMS

Luca Zhou - 224400001
Sapienza University of Rome - The Chinese University of Hong Kong (Shenzhen)
Rome, Italy - Shenzhen, China
lucazhou2000@gmail.com

ABSTRACT

This project investigates the potential for positive transfer learning in fine-tuning small language models (LLMs) on mathematical reasoning tasks. Specifically, we explore whether training a model on specific mathematical skills such as percentage calculations, interest rate computations, and arithmetic word problems can improve its performance on unseen mathematical tasks. Due to the lack of publicly available datasets categorized by mathematical skill, we created synthetic datasets where each task is represented as a question-answer pair. Since LLMs struggle with numerical computation, the answer is expressed as a mathematical formula rather than a computed value, reducing the model's burden to perform arithmetic calculations and focusing it on learning formula generation. We fine-tuned the **Flan-T5-large** model using **AdaLoRa** to optimize training within limited computational resources. The training process incorporated instruction-following examples to improve generalization and reduce overfitting on synthetic data. Our experiments on four mathematical tasks show that positive transfer generally exists in mathematical reasoning. The code and datasets of this project are publicly available here.

1 INTRODUCTION

What is your research topic This research explores the potential for positive transfer learning in small language models (LLMs) when fine-tuned on specific mathematical tasks. We investigate whether training a model on particular mathematical skills such as percentage calculation, mean computation, financial problem-solving, and arithmetic word problems can enhance its performance on unseen mathematical tasks. The goal is to understand how well symbolic and mathematical reasoning skills transfer across related tasks in LLMs, even when computational resources are limited. We evaluate the Flan-T5-large (Chung et al., 2024) model fine-tuned using AdaLoRA (Zhang et al., 2023), focusing on formula generation rather than numerical computation. This approach tests the model's ability to generalize symbolic representations after fine-tuning rather than numeric computation.

Why the task is important Mathematical reasoning is a critical capability for language models, underpinning various real-world applications such as financial forecasting, educational tools, and scientific research assistants. While large-scale LLMs have shown impressive performance across a wide range of natural language tasks, they often struggle with mathematical reasoning, particularly when calculations are involved. Understanding whether training on one mathematical skill can improve performance on others could reveal key insights into the transfer learning capabilities of LLMs. Furthermore, demonstrating positive transfer in a resource-constrained setup, such as using small models and efficient fine-tuning methods like AdaLoRA, highlights the feasibility of applying LLMs to specialized domains without requiring extensive computational infrastructure. This work contributes to advancing the understanding of symbolic reasoning in LLMs and improving their performance in mathematical tasks through targeted fine-tuning.

What you did and what you achieved In this project, we employed **Flan-T5-large** (Chung et al., 2024) ($\approx 790M$ parameters), an instruction-tuned version of T5, pre-trained on diverse tasks using

task-specific prompts, making it well-suited for tasks involving symbolic reasoning and formula generation. We fine-tuned it on four simple mathematical reasoning tasks:

- Percentage calculation
- Mean calculation
- Financial-specific problems
- Arithmetic word problems

Due to the absence of public datasets categorized by mathematical skills, we created synthetic datasets where answers were represented as mathematical formulas rather than computed values, refer to table 1 for sample training examples per task. This approach allowed the model to focus on learning symbolic representations rather than performing numerical calculations. Due to limited computational resources, the model size is upper-bounded by 1 billion parameters. Furthermore, we applied **AdaLoRA** (Zhang et al., 2023) for parameter-efficient fine-tuning, optimizing only the query and value parameters of the attention mechanism, accounting for 0.896% of total parameters only. During evaluation, we included few-shot demonstrations into the question prompt to guide the model, though these demonstrations were not included in the fine-tuning phase. Experimental results showed traces of positive transfer across tasks: **fine-tuning the model on some mathematical skills often enhanced its performance on others in a zero-shot fashion, and adding training data from additional tasks often improves the performance on the target task**. This suggests that small language models can also effectively benefit from positive transfer in mathematical reasoning. Since this project is a proof of concept, we expect further improvements in positive transfer learning for larger models and more complex mathematical tasks, where data derive from real sources.

2 Related work

Mathematical reasoning with language models has gained significant attention in recent years. Early works such as GPT-3 (Brown, 2020) and T5 (Raffel et al., 2020) demonstrated the potential of LLMs in performing various NLP tasks, though their performance on mathematical tasks remained limited due to a lack of numerical reasoning capabilities. Models like Minerva (Lewkowycz et al., 2022) and GPT-f (Polu & Sutskever, 2020) specifically addressed mathematical problem-solving by incorporating symbolic reasoning into their objectives, showcasing improved performance on benchmarks such as MATH (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021). However, these models require billions of parameters and extensive computational resources, making them less feasible in constrained environments.

To overcome these limitations, parameter-efficient fine-tuning methods like LoRA and AdaLoRA have emerged as practical solutions for adapting models without retraining all parameters. LoRA has been successfully applied to instruction-tuned models such as Flan-T5 and LLaMA (Touvron et al., 2023), enabling fine-tuning on specialized tasks while using only a fraction of the model's trainable parameters. Additionally, Hu et al. (2022) have shown that multi-task learning on related tasks can lead to better generalization. This work extends these ideas by applying AdaLoRA to mathematical reasoning tasks with a focus on symbolic formula generation, demonstrating that even small models can benefit from inter-task transfer under resource constraints.

3 PIPELINE

The pipeline followed in this project is depicted in figure 1. Next, we delve into the methods adopted in data preparation, training, and evaluation separately.

3.1 Method

Approach We approach the task of mathematical formula generation as a sequence-to-sequence language modeling task, where the model generates mathematical formulas in text form given a mathematical reasoning question. This allows us to leverage the pre-trained Flan-T5-large model's text-generation capabilities for symbolic reasoning.

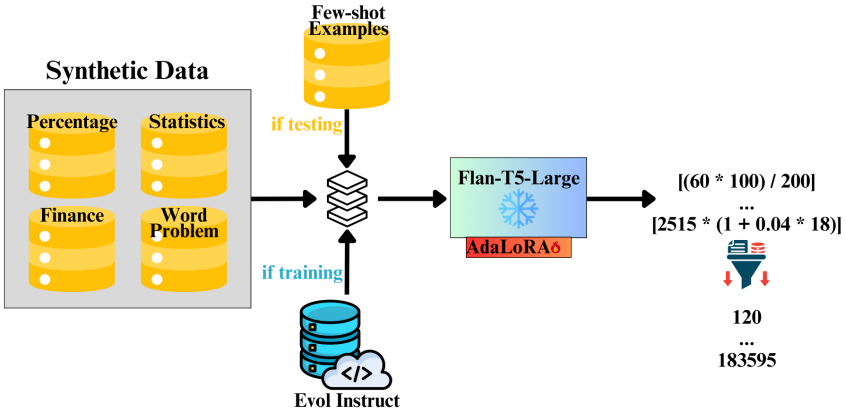

Figure 1: The overall pipeline of this project

Training Data To create the training data, we design randomized templates for each mathematical task, including:

- **Percentage**: simple percentage (500 examples), percentage change (500 examples);
- **Mean computation** given random numbers (1000 examples);
- **Financial-specific** problems: loan with interest rate (500 examples), risk assessment (500 examples), currency exchange (500 examples);
- **Word problems** about money balance (1000 examples).

Each template defines a question structure with placeholders for randomly generated numbers. During data generation, these placeholders are populated with sampled numeric values to create diverse question-answer pairs. We enclose the expected answer with square brackets so that the model learns to generate formulae between them, facilitating result parsing during evaluation. Refer to table 1 for sample training examples.

Since synthetic datasets generated this way suffer from limited linguistic diversity and facilitate overfitting, we consistently incorporate examples from the **Evol Instruct** 70k (Xu et al., 2023) dataset into the training datasets. These examples consist of general instruction-following tasks, ensuring the model retains its native instruction-following capabilities while reducing overfitting to synthetic templates.

PEFT Given the restrictive computational environment in Google Colab, we adopt parameter-efficient fine-tuning (PEFT) using the **AdaLoRA** framework. Like all LoRA-based methods, it freezes the entire model and only trains lightweight matrices to be applied to the selected parameters. AdaLoRA dynamically adjusts the ranks of these matrices during training according to their importance scores, enabling efficient adaptation while minimizing memory and computation overhead. In our case, we opt to apply AdaLoRA to the query (Q) and value (V) matrices in the attention layers, leaving only 0.896% of parameters learnable. This approach is particularly suitable for tasks requiring fine-grained symbolic reasoning, where full model fine-tuning would be computationally prohibitive.

3.2 Evaluation

We evaluate the model's performance and transfer learning through two key evaluation metrics:

Formula Accuracy We compare the formula generated by the model with the expected correct formula. However, we do not directly compare the formulae as strings. Instead, We compute their exact mathematical values and compare them. This metric measures how accurately the model has learned to produce the correct symbolic representation given a mathematical problem. A comparison results in a hit (1) when the values match and a miss (0) otherwise. When the output generated by

Task	Question	Answer (Formula)
Percentage	What is the formula for computing 41% of 89? Use the template [(percentage * base) / 100].	[(41 * 89) / 100]
Mean Calculation	Compute the formula for the mean of the following numbers: 51, -23, 48. Use the template [(sum of values) / number of values].	[(51 - 23 + 48) / 3]
Financial Question	If a principal of $1000 earns 5% interest annually, what is the formula for the amount after 3 years with simple interest? Use the template [principal + (principal * rate * time)].	[1000 + (1000 * 0.05 * 3)]
Word Problem	Luca buys 3 apples at $2/unit. Please tell me the formula for computing the overall balance. Remember to use negative signs for spending and positive for earning.	[(3 * 2) + (2 * 1.5)]

Table 1: Sample training examples

the model cannot be interpreted mathematically, that counts as a miss as well. The final accuracy of an experiment is the average of the binary hit-miss vector:

$$\text{Average Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(y_{\text{pred}}^{(i)} = y_{\text{true}}^{(i)}\right) \tag{1}$$

Where N is the sample size, y_{pred} and y_{true} are the predicted and true values respectively, interpreted from the formulae.

Cross-task Generalization To assess positive transfer, we evaluate both the zero-shot and multi-task transfer learning. Specifically, in **zero-shot transfer learning**, we evaluate the model's performance on tasks it was not fine-tuned on, whereas in **multi-task transfer learning**, we examine whether fine-tuning with additional data from other tasks benefits the target task. To conduct this analysis, we fine-tune and test the following models:

- 1 Pretrained baseline
- 4 Task-specific experts
- 1 Multi-task model fine-tuned jointly on all four tasks
- 3 Multi-task models fine-tuned jointly on all but one task

Zero-shot positive transfer occurs whenever a model not fine-tuned on the target task outperforms the pretrained baseline on that task. Multi-task positive transfer occurs when fine-tuning a model on the target task benefits from adding training data from other tasks. Few-shot prompting is adopted during evaluation but not in fine-tuning, see table 2 for sample test examples.

4 EXPERIMENTAL SETTING

All our experiments leverage a pretrained Flan-T5-large model ($\approx 790M$ parameters) loaded from Hugging Face. As synthetic data lacks linguistic diversity, we always include in the training dataset a random portion of Evol Instruct examples, of which the size is equivalent to 30% of that of the training dataset. The goal is to retain the instruction following the ability of the model. The fine-tuning objective is the cross-entropy loss function, where irrelevant output tokens are padded with -100 to prevent them from contributing to the loss. For all experiments, we consistently use the AdamW optimizer and fine-tune for 6 epochs with weight decay of $1e^{-5}$ and learning rate scheduling, starting from $4e^{-4}$ and shrinking by a factor of 10 at epochs 4 and 6. Because the batch size is constrained to 4, gradient accumulation is applied on every 4 batches to simulate a batch size of 16. We applied AdaLoRA as the PEFT method to the query (Q) and value (V) matrices in attention layers, configuring it with rank = 24, alpha = 32, and dropout = 0.1. Overall, we fine-tune one expert model per task, one multi-task model on all tasks, and one multi-task model for each left-out task

Task	Question	Answer (Formula)
Percentage	39% of -45 has formula [(39 * -45) / 100]. 98% of -30 has formula [(98 * -30) / 100]. 88% of 14 has formula [(88 * 14) / 100]. 30% of 36 has formula [(30 * 36) / 100]. 74% of 32 has formula	[(74 * 32) / 100]
Mean Calculation	The mean of 23, 18, 53, 54, 49 has formula [(23+18+53+54+49) / 5]. The mean of 9, -74, 58, -46 has formula [(9-74+58-46) / 4]. The mean of -69, -94, -73, -89 has formula	[(-69-94-73-89) / 4]
Financial Question	Loan of $9602 with 12% interest over 14 years has formula [9602 * (1 + 0.12 * 14)]. Loan of $4213 with 10% interest over 5 years has formula [4213 * (1 + 0.1 * 5)]. Loan of $6823 with 16% interest over 20 years has formula [6823 * (1 + 0.16 * 20)]. Loan of $2401 with 14% interest over 2 years has formula	[2401 * (1 + 0.14 * 2)]
Word Problem	Luca buys 3 bottles of water at $12/unit has formula [- (12 * 3)]. Luca sells 6 phones at $999/unit has formula	[(999 * 6)]

Table 2: Sample test examples

(i.e. for each task, we fine-tune one multi-task model jointly on all other three tasks). During testing, we generate different but the same number of synthetic examples as in training, with between 2 and 5 few-shot demonstrations in the question prompt.

5 Results & Discussion

We divide this analysis into single-task and multi-task transfer learning, which differs by the number of tasks the model is fine-tuned on. In the former case, experts are evaluated on a target task but are fine-tuned on a different task, whereas in the latter, the models are fine-tuned jointly on multiple tasks.

Single-task Transfer Learning This experiment revolves around zero-shot transfer learning. The experimental outcome is depicted in a matrix form, where columns represent the evaluation task and rows represent the single-task expertise. See table 3.

Expert/Test Task	Percentage	Mean	Finance	Word Problem
Percentage	40.09%	3.31%	58.96%	80.56%
Mean	8.16%	92.76%	73.82%	48.74%
Finance	5.85%	0.17%	92.12%	80.17%
Word Problem	16.07%	3.38%	68.16%	92.59%
Pretrained	20.35%	0.41%	58.39%	70.14%

Table 3: Single-task Test Accuracy: higher values are greener; lower are redder. The colors are shown relative to each evaluation task (column-wise)

A noteworthy result is that the pretrained model does not always perform the worst, as can be seen in the first column where it performed the second best. However, for all other tasks, the pertaining performance is surpassed by some experts. Therefore, the takeaway is that not all experts exceed the pretrained model in zero-shot mathematic reasoning, but there are some tasks that definitely exhibit better positive transfer.

Multi-task Transfer Learning We report the results again in a similar matrix form 4. However, the rows here do not represent experts but multi-task models fine-tuned on all tasks except the indicated one.

Model/Test Task	Percentage	Mean	Finance	Word Problem
All Merged	16.85%	98.93%	94.74%	88.41%
w/o Percentage	0.75%	99.31%	94.76%	83.97%
w/o Mean	40.89%	3.33%	92.39%	86.76%
w/o Finance	22.10%	99.60%	88.97%	91.81%
w/o Word Problem	13.79%	99.38%	93.67%	81.55%
Pretrained	20.35%	0.41%	58.39%	70.14%

Table 4: Multi-task Test Accuracy: Higher values are greener and lower are redder. The colors are shown relative to each evaluation task (column-wise)

From the above table, we derive two insightful findings. First, except for the percentage task in the first column, the multi-task model always outperforms the pretrained baseline in zero-shot fashion by a large margin. This suggests that mathematical tasks generally mutually transfer positively. Second, by comparing the two tables (3 4), we observe that for a given target task, properly including training data from other tasks can be beneficial. This is especially true for mean and finance tasks. The reason why positive transfer did not occur for all tasks might be attributed to overfitting on the synthetic data, given the limited formats and linguistic features of percentage and word problem examples. In the ideal scenario where the datasets are real and well-prepared, we expect to observe positive transfer to occur even more frequently, unless some tasks entail skills that conflict with one another and cause interference, which is unlikely in mathematics.

6 Conclusion

This project demonstrates that fine-tuning small language models (LLMs) on mathematical reasoning tasks can lead to positive transfer learning, even when computational resources are severely limited. By focusing on symbolic reasoning and formula generation rather than direct numerical computation, we were able to leverage the **Flan-T5-large** model' s ability to generalize across related mathematical tasks. Our experiments on four tasks (percentage calculation, mean computation, financial problem-solving, and arithmetic word problems) showed that the model could transfer learned skills across tasks in a zero-shot fashion. Moreover, multi-task transfer learning was shown to enhance performance, where fine-tuning on multiple tasks improved generalization across unseen tasks. Despite the small model size and synthetic data, the results indicate that positive transfer is achievable, especially with efficient fine-tuning methods like AdaLoRA.

While this study serves as a proof of concept, further research with larger models, real data, and more complex mathematical tasks could yield even more robust and grounded results. The findings of this project highlight the potential for small, resource-efficient models to handle specialized tasks, making them more accessible for applications in fields like education, finance, and scientific research.

The code and datasets of this project are accessible here.

ACKNOWLEDGMENT

This report is in partial fulfillment of the final project for the **CSC 6203** course (Selected Topics in Computer Science III by Professor Benyou Wang), see details in `https://llm-course.github.io/`.

REFERENCES

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

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

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. *Sort*, 2 (4):0–6, 2021.

Ziniu Hu, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed Chi. Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35:11450–11466, 2022.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*. Openreview, 2023.