Toolformers on Minimal Data
Language Models Can Teach Themselves to Use Tools

Zachary Chosed (zac27), Daniel Chuang (dc863), Eric Marchetti (emm347), Kevin Hu (kh785), Tony Matchev (akm99)
Cornell University Bowers CIS Computer Science - daniel-chuang/mathtoolformer-llm

Cornell Bowers CIS Computer Science

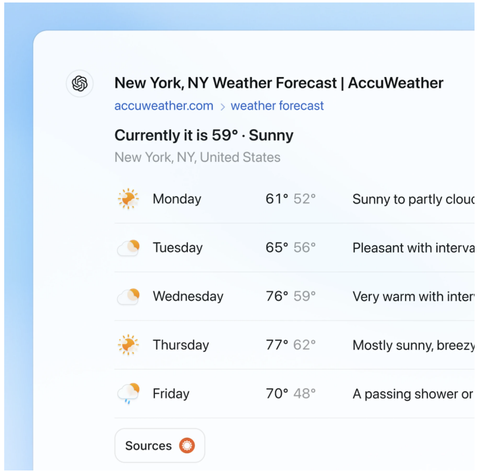
Introduction

- LLMs such as GPT struggled with complex arithmetic applications, such as multi-digit multiplication
- LLMs are trained on historical data, which prevents them from getting current information such as the date, weather, etc
- Our goal is to teach LLMs to use APIs in the following format:

<tool:calculator>5+3+2*5<tool>

Background

- *Toolformer: Language Models Can Teach Themselves to Use* Tools saw improvements in performance are substantial for many different APIs use cases (Math, QA, Machine Translation).[1]



ChatGPT calling the Accuweather API

Motivation

We aim to demonstrate the use of *Toolformer* for the average user by highlighting improvements in small LLMs with public access and low training costs

Methodology

- We elected to use the QWEN2.5-Math-1.5B[4] model offered by Hugging Face
- We tested our model again the EleutherAI arithmetic dataset, which is a HuggingFace set of 10 elementary test configs (4 addition, 4 subtraction, 1 multiplication, and notably 1 chain of calculation).
- We finetuned the foundational model on this dataset as a performance baseline

Table 1: Arithmetic QA: SmolLM2-135M (Base) Correctness

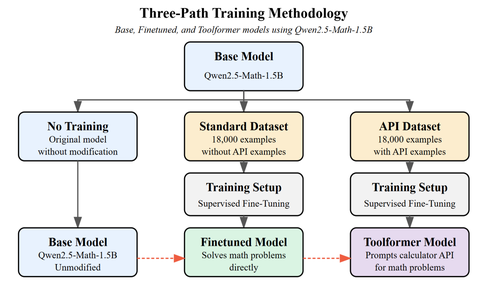
Config	Question	Expected	Model	Correct
3-digit addition	What is 355 plus 967?	1322	1322	✓
2-digit subtraction	What is 2 minus 40?	-38	38	✗
2-digit multiplication	What is 77 times 62?	4774	5124	✗
1-digit chain	What is (3 * 5) - 8?	7	8	✗

- The *Toolformer* dataset was modified into the format shown below

Table 2: Arithmetic QA: Toolformer Dataset

Question	Expected Tool Call
Question: What is 31 plus 72? Answer:	<tool:calculator>31+72</tool>
Question: What is (4 - 2) + 7? Answer:	<tool:calculator>(4-2)+7</tool>

- Hypothesis: fine-tuning will substantially boost small model performance by increasing latent space simplicity
- Implementation: fine-tuned with calls to a calculator API (selected due to its myriad uses and ease of prompt auto-generation without human involvement)

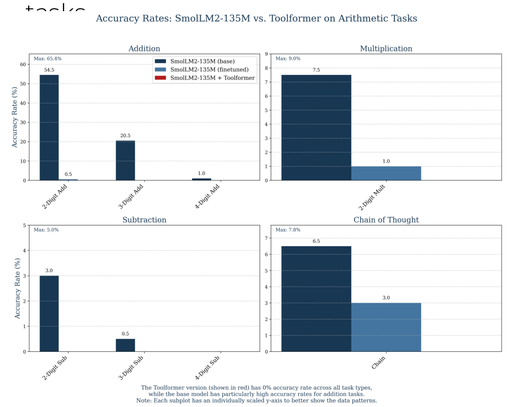


These elementary level questions were ideal for the small LM to demonstrate a proof of API usage.

We trained the model using SFTTraining with a learning rate of 0.0004, weight decay of 0.01 with the AdamW optimizer using Hugging Face's standard training program

Results

- QWEN2.5-Math-1.5B (base model), with its limited size and train time, achieved poor performance on mathematical tasks



- We found that the power of *Toolformer's* reduction in the complexity of the latent space had a significant impact on mode performance

Conclusion

- We did not see the improvements suggested by the paper in our testing
- We suspect that the barrier to learning an appropriate latent representation to utilize tools (API calls) is the size of the model, and have not reached the required size to do so

Future Work

- Determine what size the *Toolformer* implementation becomes useful
- Test its efficacy at that size

References

[1] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). *Toolformer: Language Models Can Teach Themselves to Use Tools*. arXiv. https://arxiv.org/abs/2302.04761

[2] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). *Scaling laws for neural language models*. https://arxiv.org/abs/2001.08361.

[3] Bi, J., Wu, Y., Xing, W., & Wei, Z. (2024). Enhancing the Reasoning Capabilities of Small Language Models via Solution Guidance Fine-Tuning. https://arxiv.org/abs/2412.09906.

[4] Qwen Team. (2024). *Qwen/Qwen2.5-Math-1.5B* [Computer software]. Hugging Face. https://huggingface.co/Qwen/Qwen2.5-Math-1.5B