A Comparative Analysis of Reasoning in LLMs

Alex Lindley
Supervisor: Oscar Benjamin

Motivation

The rise of Large Language Models (LLMs) in recent years has sparked interest in their ability to perform mathematical reasoning. This surge has driven the development of ever larger models, which generally yield improved accuracy (see Figure 1).

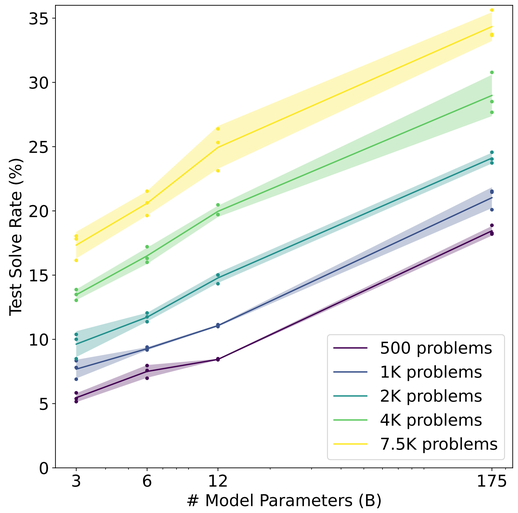

Figure 1. Final evaluation for various GPT-3 model sizes after finetuning on training sets of different sizes [1].

Research by Deepseek [2] has demonstrated that more optimised models can be trained at a fraction of the price while achieving performance comparable to, or even better than, cutting edge models such as GPT-4.

Problem Statement

This raises the question: to what extend does a model's reasoning capability depend on its parameter count versus the quality of its training methodology? Specifically, is it possible to optimise smaller LLMs, without sacrificing their ability to reason or are their limitations inherent compared to larger models?

Models

For a robust evaluation of reasoning accuracy, three model types are compared:

- **Small LLM**: Evaluates the reasoning efficiency of compact models (under 100M parameters). This serves to test the hypothesis that efficient architectures can deliver competitive reasoning performance despite smaller size. Model selected: DistilGPT-2 (82M parameters).
- **Large LLMs**: Benchmark performance with models exceeding 1B parameters (GPT-4, DeepSeek and InternLM) to set an upper performance bound.
- **Supervised Machine Learning (ML) algorithms**: Establish a baseline using traditional methods with limited textual understanding. Models used: Logistic Regression (LR), Support Vector Machines (SVM) and Random Forest (RF).

Dataset

The evaluation leverages the GSM8K dataset [1], which includes grammar school level maths problems requiring 2 to 8 steps to solve. Two variants are used:

- **GSM8K**: Standard version, includes question, step by step reasoning and final answer.
- **GSM8K-v2**: Enhanced version where each question is paired with a computation snippet (as if the LLM were using a calculator) to ensure robust reasoning, even where answer tokens are absent.

Example:

Question: *Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?*
Reasoning: *Natalia sold* $\frac{48}{2} = \mathbf{24}$ *clips in May.*
Natalia sold $48 + 24 = \mathbf{72}$ *clips altogether in April and May.*
Computation code (GSM8K-v2 only): *"*$48 + (48/2)$*"*

Structure

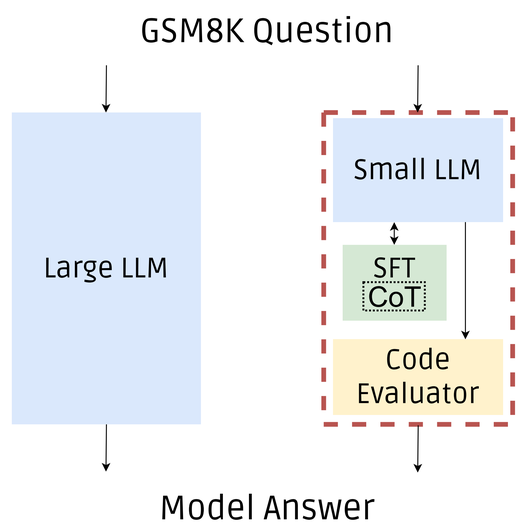

Figure 2. Large LLM versus enhanced small LLM.

This diagram contrasts the inherent reasoning of a large LLM with a small LLM enhanced by:

- **Supervised Fine Tuning (SFT)**: Refines the model using task specific data to boost the mathematical reasoning.
- **Chain of Thought (CoT)**: Provides intermediate reasoning steps to improve understanding.
- **Code Evaluator (CE)**: Offloads computations to a "calculator" to handle simple arithmetic. This allows the model to focus on logic.

Results

Results reveal a clear hierarchy. ML algorithms establish a baseline, and the standard small LLM performs around this level. However, when enhanced with SFT and particularly the CE, small LLM accuracy triples (with CE solutions can be computed). Both ML approaches underwent hyperparameter tuning, and the small LLM was tested for up to 25 epochs.

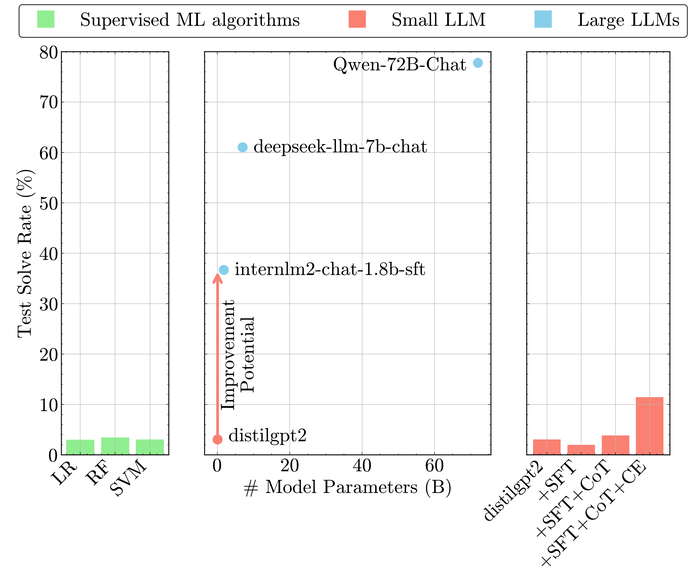

Figure 3. Accuracy for various model sizes and types tested on GSM8K dataset.

Data confirms that larger models generally achieve higher accuracy. Yet, with proper training and fine-tuning, enhanced small LLMs might eventually match the performance of the lowest-performing large LLMs.

Key Insights

- Enhanced small LLMs, when paired with SFT, CoT and a CE, achieve up to triple the baseline accuracy.
- Large LLMs clearly outperform in accuracy, however, their training cost (financially and computationally [2, 3]) can limit practical accessibility (82M vs. 72B).
- The Code Evaluator plays a crucial role in bridging the size gap, enabling small models to improve their mathematical reasoning.

References

[1] 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. *CoRR*, abs/2110.14168, 2021. URL `https://arxiv.org/abs/2110.14168`.

[2] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

[3] Zijian Wu, Suozhi Huang, Zhejian Zhou, Huaiyuan Ying, Jiayu Wang, Dahua Lin, and Kai Chen. Internlm2.5-stepprover: Advancing automated theorem proving via expert iteration on large-scale lean problems, 2024. URL `https://arxiv.org/abs/2410.15700`.