TOM: Teach Only What Matters for Selective Chain-of-Thought Supervision

ChanJoo Jung (2022121057)

1 Introduction

This is a simple project for CAS2105 @ Yonsei University in 2025.

2 Task Definition

- **Task description:** The goal of this project is to train a small language model to solve grade-school math problems by effectively borrowing reasoning capability from a larger model. Given a math problem from the GSM8K [1] dataset, the model generates a step-by-step reasoning process followed by a final numeric answer. Instead of naively using the teacher model's reasoning on all examples, I investigated a selective strategy in which teacher-generated chain-of-thought (CoT) [2] supervision is applied only to **difficult** problems. The key message I want to deliver is that "Only a small subset of data samples is needed. Using all samples is not necessary"
- **Motivation:** Recent developments in large language models have shown that smaller models can efficiently leverage the capabilities of larger models through diverse methods of guidance, including speculative decoding [3] and distillation [4]. This raises a natural question: *can a weaker model also borrow the reasoning ability of a stronger model, and if so, how should this be done?* A simple, direct approach is to just train the smaller model to adopt ALL the reasoning trace of the larger model. However, **are all reasoning traces equally useful?** Motivated by this concern, I explored whether reasoning guidance from a large model can just only be applied selectively rather than uniformly, and investigated how to more intelligently borrow reasoning chains only when they are truly beneficial for the student.
- **Input / Output:**
  - **Input:** A grade-school math problem.
  - **Output:** A step-by-step reasoning process and a final numeric answer.
- **Success criteria:** The system is considered successful if:
  - Despite using less teacher supervision, the selectively trained student model outperforms the self-CoT baseline,
  - AND attains accuracy comparable to the teacher-CoT baseline.

  To know more about the baselines, please refer to 3.1.

3 Methods

3.1 Baseline and Reference Models

To evaluate the effectiveness of selectively borrowing reasoning chains from a larger model, we compare our method against several baselines that represent different ways of providing (or not providing) reasoning supervision to a small language model.

1. Self-CoT Baseline

- The student model first generates its own chain-of-thought for each training example.
- The model is then trained to imitate these self-generated reasoning traces.
- This baseline examines the capacity of a small model to bootstrap its reasoning skills independently of a more powerful teacher.
- It acts as a benchmark for comprehending the advantages of external reasoning supervision.

2. Full Teacher-CoT Reference

- **This actually is not a baseline to win, rather an upperbound that the selective model should able to be comparable with.**
- For every training example, a larger teacher model produces chain-of-thought reasoning.
- For each case, the student model is trained to mimic the teacher's complete reasoning trace and final response.
- This baseline represents the standard approach to chain-of-thought distillation.
- It assumes that teacher reasoning is uniformly beneficial, regardless of problem difficulty.

Why these baseline and reference models are informative

- The self-CoT baseline evaluates whether reasoning can be learned through self-generated explanations/reasoning traces alone.
- The full teacher-CoT baseline is training the student model with all reasoning traces of the teacher model.
- Together, these baselines allow us to isolate the effect of selectively applying teacher-generated reasoning.

Common Failure Cases of the Self-CoT Baseline

- Self-generated chain-of-thought (CoT) can exhibit plausible logical structure while containing subtle arithmetic or modeling errors.
- Once an early mistake occurs, the model tends to remain consistent with its incorrect reasoning.

3.2 AI Pipeline

- **Models used:**
  - Student model: `Llama-3.2-3B-Instruct`[1]
  - Teacher model: `Llama-3.1-8B-Instruct`[2]

[1] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[2] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

- **Pipeline stages:**
  - **A. Preprocessing**
    1. Load the GSM8K dataset and extract a subset of 100 training examples and 10 test examples (This is due to the homework specification for using only a few dataset).
    2. Parse the gold final numeric answer from each GSM8K solution and store it as `gold_final`.
    3. Shuffle the training data (for reproducability, I have set seed=42) before selecting the subset to reduce order bias.
    4. Load the tokenizer of the student model.
    5. Construct chain-of-thought (CoT) style prompts used for both generation and evaluation:

```
You are an expert math tutor.
Solve the problem step by step, then clearly state the final numeric
    answer.
At the very end, write the answer in the form: Answer: <number>.

Question: {math problem}
Let's think step by step.
```

    6. During training, mask prompt tokens by setting their labels to `-100` so that the loss is computed only on the target output tokens. This means that -100 tokens will be disregarded.
    7. Inputs are truncated to a maximum sequence length; for hardness estimation, sequences are padded to a fixed max length.
  - **B. Hardness Estimation**
    8. Train an initial answer-only student model using only the final numeric answers as supervision (i.e., targets are `gold_final`).
    9. Use this trained student to compute the log-probability of the correct answer for each training example.
    10. Define example hardness as the average negative log-likelihood (NLL) of the correct answer tokens, computed only over the answer portion (prompt tokens excluded).
    11. Split the training data into easy and hard subsets using a percentile threshold of the hardness scores (in the implementation, a 50/50 split is used).
  - **C. Reasoning Signal Generation**
    12. For all training examples, generate teacher chain-of-thought reasoning using the teacher model (sampling with `temperature=0.7`, `top_p=0.9`).
    13. Generate student chain-of-thought reasoning using the (answer-only) student model for baseline comparisons (also with `temperature=0.7`, `top_p=0.9`).
    14. For selective distillation, assign teacher-generated reasoning as the training target only for hard examples; for easy examples, use only the final numeric answer (`gold_final`) as the training target. Note that in both cases the same CoT-style prompt is used, but the supervision differs (teacher reasoning vs. answer-only target).
  - **D. Representation and Training**

15. Load the student base model with QLoRA (LoRA training with quantization). For justification of this training method, please refer to **Design choices and justification** 3.2.
16. Only LoRA parameters are updated during training while the base model remains frozen.
17. Train separate student models for each baseline (self-CoT, full teacher-CoT) and the proposed selective method using standard causal language modeling loss.
18. Since the selective model trained on fewer train data (almost half only), the selective model is trained for additional epochs (4 epochs vs. 2 epochs for the other baselines which has twice as more train data samples than the selective method). The rest of the hyperparameters are all same.

– **E. Inference and Evaluation**

19. At inference time, all trained models are prompted using the same CoT prompt and asked to generate step-by-step reasoning with a final answer at the end.
20. The final numeric answer is extracted from the generated output using pattern-based parsing (searching for a number, prioritizing lines containing "Answer").
21. Model performance is evaluated using exact-match accuracy on the GSM8K test subset.

- **Design choices and justification:** The pipeline is designed to reflect realistic constraints of small-model training. PEFT training using quantization (QLoRA) enables training in 1 gpu. Hardness-based selection allows the student model to borrow teacher reasoning only for difficult data samples,to provide stronger supervision where it is most needed.

4 Experiments

4.1 Datasets

- **Dataset source**: GSM8K (`main`) subset.
- **Size**: 110 samples total (100 training examples, 10 test examples).
- **Splits**: 100 for training and 10 for testing (no validation split due to the small subset size).
- **Preprocessing**: See Section 3.2. We shuffle the dataset (seed=42), extract the gold final numeric answer, and build chain-of-thought prompts for training and evaluation.

4.2 Metrics

- **Accuracy (GSM8K)**: We extract the final numeric answer from the model output and compute exact-match accuracy against the gold final answer on the test subset. For robustness, we use a *graceful* comparison that treats equivalent numeric strings (e.g., "32" and "32.") as the same value.

4.3 Results

The results are presented in Table 1.

4.4 Qualitative Examples

Below are qualitative examples comparing the **Self-CoT**, **Teacher full-CoT**, and **Selective** models on the GSM8K test subset. The examples highlight common failure modes: the self-CoT model

Model	Accuracy (Exact Match)
Self-CoT student (QLoRA)	0.50
Teacher full-CoT student (QLoRA)	0.60
Selective teacher-CoT student (QLoRA)	**0.60**

Table 1: Exact-match accuracy on the GSM8K test subset (10 examples).

can produce plausible reasoning but still output an incorrect final number, while the selective model can match teacher-level performance on some questions.

Example 1 (Self-CoT fails; Teacher/Selective succeed).

QUESTION:

> Darrell and Allen's ages are in the ratio of 7:11. If their total age now is 162, calculate Allen's age 10 years from now.

GOLD FINAL ANSWER:

> 109

SELF-CoT (prediction = 99; incorrect):

> Step 1: Let's represent Darrell's age as 7x and Allen's age as 11x.
> Step 2: The total age is 7x + 11x = 18x.
> Step 3: 18x = 162 ⇒ x = 9.
> Step 4: Allen's current age is 11x = 99.
> Step 5: Therefore, Allen's age 10 years from now is 99.
> Answer: 99.

TEACHER full-CoT (prediction = 109; correct):

> Step 1: Let Darrell = 7x, Allen = 11x.
> Step 2: 7x + 11x = 18x = 162 ⇒ x = 9.
> Step 3: Allen now = 11x = 99, so in 10 years = 99 + 10 = 109.
> Answer: 109.

SELECTIVE (prediction = 109; correct):

> Step 1: Let Darrell = 7x, Allen = 11x.
> Step 2: 18x = 162 ⇒ x = 9.
> Step 3: Allen now = 99, so in 10 years = 109.
> Answer: 109.

Example 2 (All models fail).

QUESTION:

> Lorraine and Colleen are trading stickers for buttons. Each large sticker is worth a large button or three small buttons. A small sticker is worth one small button. A large button is worth three small stickers. Lorraine starts with 30 small stickers and 40 large stickers. She trades 90% of her small stickers for large buttons. She trades 50% of her large stickers for large buttons and trades the rest of them for small buttons. How many buttons does she have by the end?

GOLD FINAL ANSWER:

89

SELF-CoT (prediction = 3; incorrect):

Step 1: Lorraine starts with 30 small stickers and 40 large stickers.
Step 2: 90% of 30 is 27.
Step 3: 50% of 40 is 20.
... (truncated) ...
Answer: 3.

TEACHER full-CoT (prediction = 7; incorrect):

Step 1: Lorraine starts with 30 small stickers and 40 large stickers.
Step 2: 90% of 30 is 27 small stickers.
Step 3: 50% of 40 is 20 large stickers.
... (truncated) ...
Answer: 7.

SELECTIVE (prediction = 60; incorrect):

Step 1: Lorraine starts with 30 small stickers and 40 large stickers.
Step 2: 90% of 30 is 27.
Step 3: 50% of 40 is 20.
... (truncated) ...
Answer: 60.

Example 3 (All models succeed).

QUESTION:

Indras has 6 letters in her name. Her sister's name has 4 more letters than half of the letters in Indras' name. How many letters are in Indras and her sister's names?

GOLD FINAL ANSWER:

13

SELF-CoT (prediction = 13; correct):

Step 1: Indras has 6 letters.
Step 2: Half of 6 is 3.
Step 3: Sister has 3 + 4 = 7 letters.
Step 4: Total letters = 6 + 7 = 13.
Answer: 13.

TEACHER full-CoT (prediction = 13; correct):

Step 1: Half of 6 is 3; 3 + 4 = 7.
Step 2: 6 + 7 = 13.
Answer: 13.

SELECTIVE (prediction = 13; correct):

Step 1: Half of 6 is 3; 3 + 4 = 7.
Step 2: 6 + 7 = 13.
Answer: 13.

4.5 Conclusion

Our results indicate that selectively applying teacher chain-of-thought supervision to hard examples is sufficient. The selectively trained model outperforms the self-CoT baseline and achieves accuracy comparable to the teacher full-CoT baseline, while using substantially less supervision.

5 Reflection and Limitations

I believe the use of a very small subset of the GSM8K dataset limits the statistical significance of the findings in this small-scale project. In fact, I am not sure whether this performance trend would persist when scaling to the full dataset. Example hardness may not apply to other models or datasets since it is based on a single answer-only student model.Furthermore, because just 10 test samples are used for the evaluation, even slight variations in predictions might have a significant impact on the stated accuracy.Lastly, while though selective supervision uses less teacher reasoning, it still necessitates having access to a teacher model throughout training, which might be expensive in some situations.Future research should investigate different hardness metrics and confirm these results on a bigger scale.

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, 2021. URL https://arxiv.org/abs/2110.14168.

[2] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

[3] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.