Knowledge Distillation via Data Scaling: How Much Teacher-Generated Instruction Data Does a Student Need?

Meryem Nur Nalbant

What is Knowledge Distillation?

Definition: Transfer capability from a large teacher model to a smaller student by using the teacher to generate high-quality answers and training the student to imitate them.

- The student never sees the teacher's weights; they learns only from the teacher's text, like learning from a skilled tutor.
- The teacher's predictions provide richer supervision than labels since they encode probabilities/preferences, uncertainty, output structure (format + reasoning style).

Problem it solves: Large LLMs are powerful but expensive to deploy. Distillation makes models practical by enabling:
- lower latency and higher tokens/sec
- smaller memory footprint (VRAM/RAM)
- lower serving cost
- deployment on limited hardware (consumer GPUs / edge)

Goal: preserve instruction-following quality while making inference faster and cheaper.

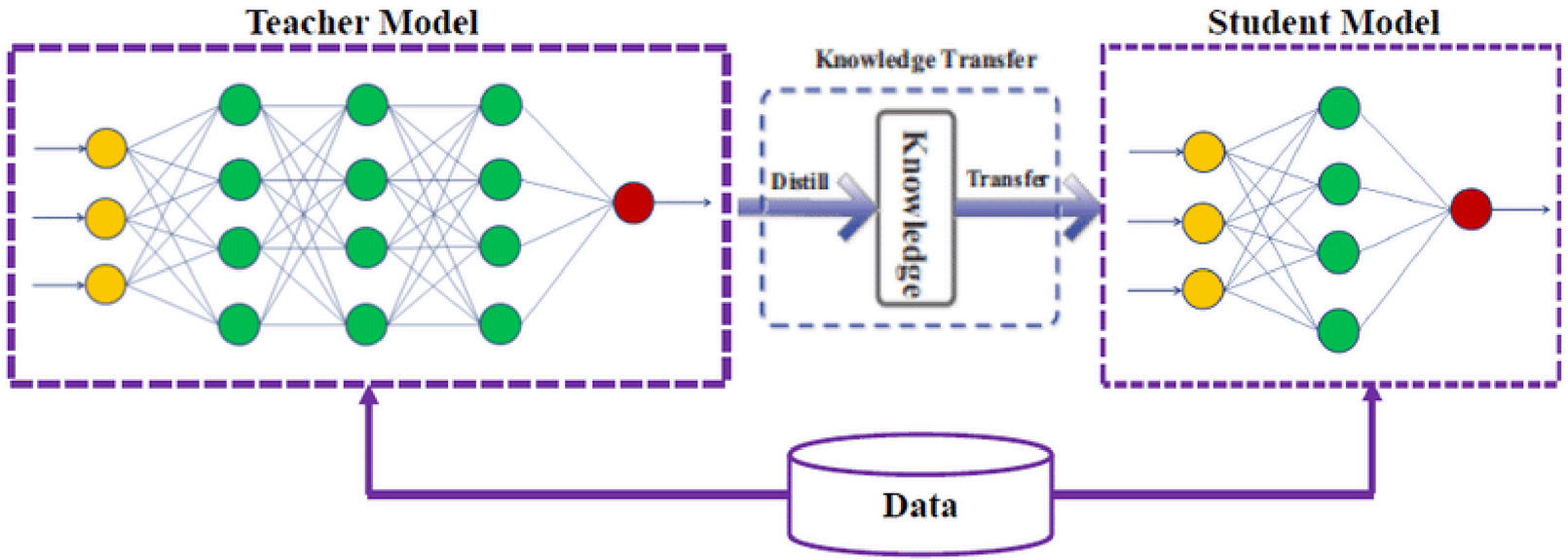
What is the Teacher–Student Model?

A **Teacher–Student model** is a training framework where a large, high-performing model (Teacher) guides the training of a smaller, efficient model (Student).

- **Teacher:** a high-capacity neural network trained for strong performance.
- **Student:** a compact network trained to reproduce the teacher's behavior using far fewer parameters.

Instead of learning only from human-written data, the student also learns from the teacher's predictions (like generated responses or token probabilities), without access to the teacher's internal weights.

Teacher–Student & Knowledge Distillation



- The result is a compact model that approximates the performance of a much larger one.

Source: Adapted from Almadan & Rattani (2021), Compact CNN Models for On-device Ocular-based User Recognition in Mobile Devices

Core Research Question (Data Scaling Distillation)

Can a very small model approach the quality of a large model if it is trained on enough high-quality teacher-generated data?

More precisely:
How does the student's performance change as we increase the amount of teacher-generated training data? (0.5k → 2k → 5k → 10k → 15k samples)

What this tests:
Whether **more distilled data can compensate for smaller model capacity**, revealing the quality–data scaling trade-off in teacher–student distillation.

Experimental Design

We train the student model (GPT-2 Small) using Supervised Fine-Tuning (SFT) on data generated by the teacher model (GPT-2 XL).

Role	Model	Size
Teacher	GPT-2 XL	1.5B parameters
Student	GPT-2 Small	124M parameters

The teacher is about 12× larger than the student.

Supervised Fine-Tuning (SFT)

Supervised fine-tuning (SFT) trains a model on prompt–response pairs so it learns to produce the desired output for a given instruction.
Each training example looks like: Prompt → Target Response

Alpaca Dataset

Alpaca is a collection of prompts and answers designed to teach a model how to follow natural-language instructions.

instruction string · *lengths* 9 489	input string · *lengths* 0 2.47k	output string · *lengths* 0 4.18k	text string · *lengths* 154 4.5k
Give three tips for staying healthy.		1.Eat a balanced diet and make sure to include plenty of…	Below is an instruction that describes a task. Write a…
What are the three primary colors?		The three primary colors are red, blue, and yellow.	Below is an instruction that describes a task. Write a…

Experimental Design

The process is:
1- Prompts come from the Alpaca instruction dataset
2- GPT-2 XL generates responses for each prompt
3- GPT-2 Small is trained to imitate those responses

What is being scaled?

The amount of teacher-generated training data.
We train five separate student models.
This lets us observe the diminishing-returns curve:
How much does each extra batch of data improve the student?

Dataset Size (Samples)
500
2000
5000
10000
15000

Experimental Design

Training Setup (Fixed Across All Runs)

To ensure a fair comparison, all models use the same training configuration

So any performance difference comes only from more or less teacher data, not from different training settings.

Epochs	2
Learning Rate	5×10^{-5}
Batch Size	16

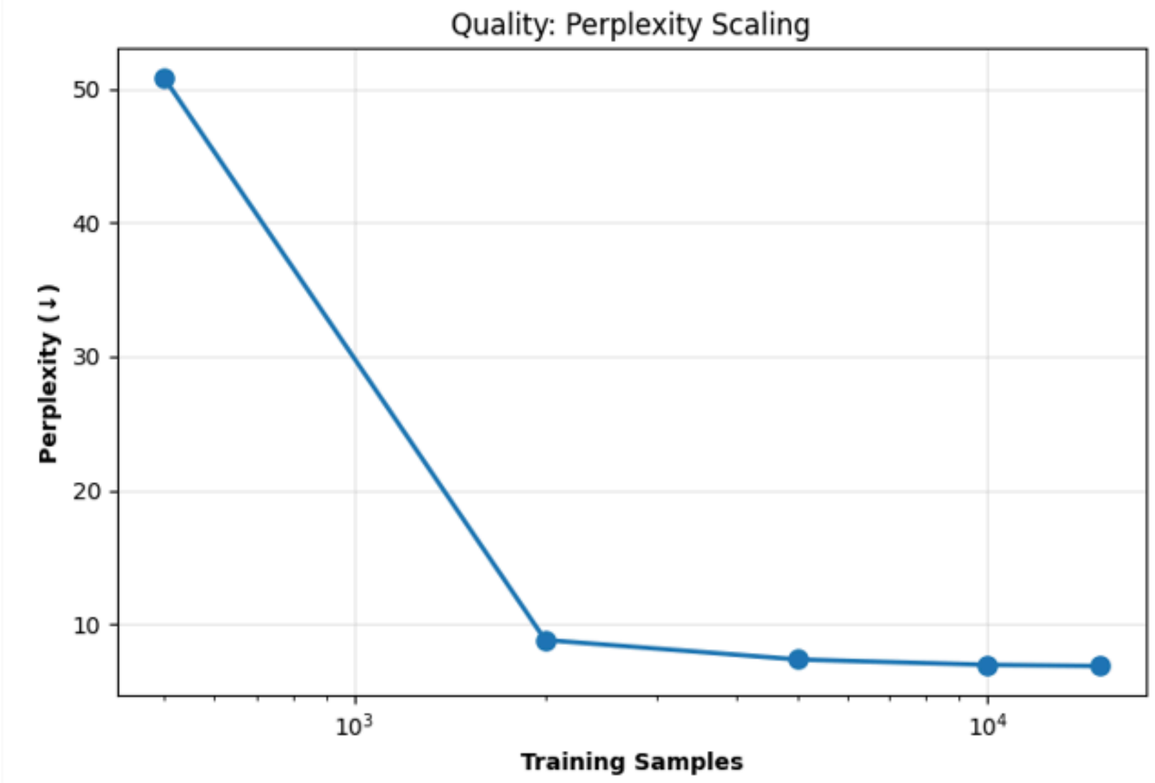
Evaluation Metrics

1. Perplexity (Language Quality): measures how well a model predicts the next token in a sequence. This reflects fluency and internal language modeling ability. "Does the student model sound like a good language model?" (ppl)

2. Instruction Adherence (Task Following): how well the model follows explicit instructions using a 50-prompt test set. This measures practical usefulness, not just fluency. (pass rate)

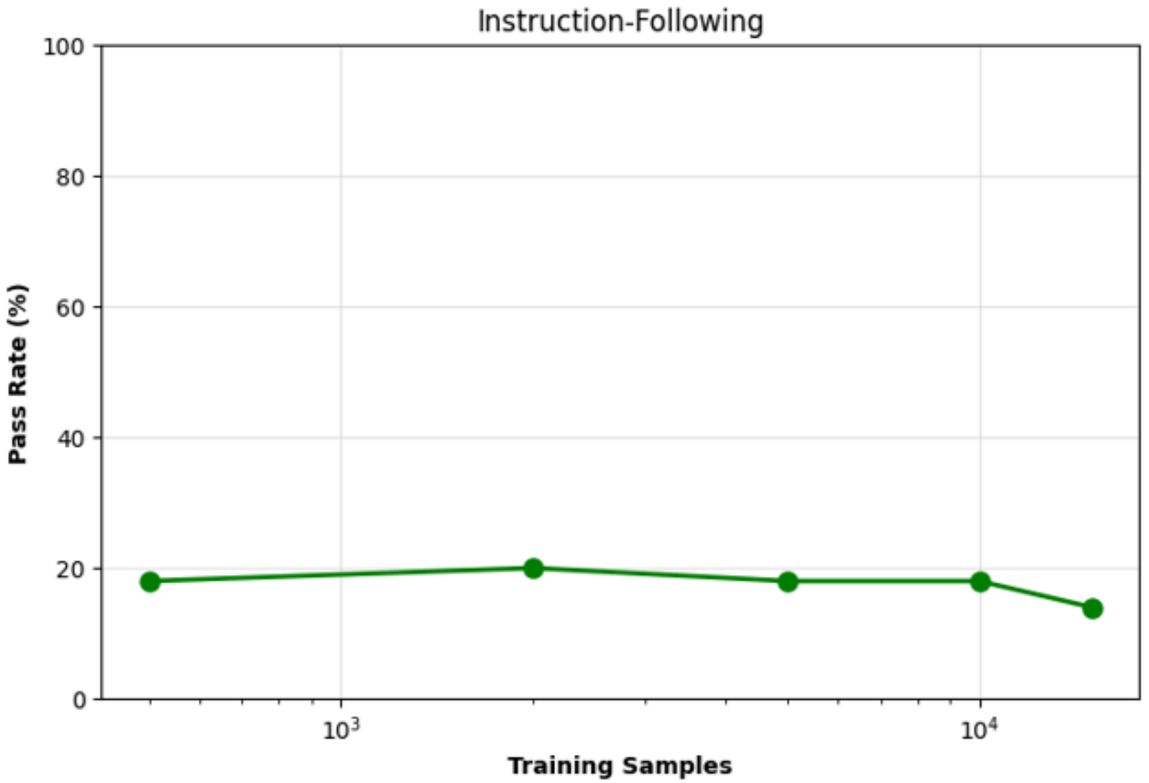
3. Efficiency (Why Use a Small Model?): (i) Tokens per second (generation speed) & (ii) Memory usage. This captures the deployment trade-off: how much performance we get per unit of compute. A small drop in quality may be acceptable if the model is much faster and cheaper. (tokens/sec & size)

Results: Perplexity



- Perplexity drops sharply from 50.81 (500 samples) to 8.84 (2,000 samples), showing a major gain in language quality.
- Performance continues to improve but saturates at scale, reaching 6.89 at 15,000 samples.
- Most of the benefit comes early: the jump from 500 → 2,000 is transformative, while 10k → 15k (6.98 → 6.89) shows a clear plateau in linguistic confidence.

Results – Instruction Following



- Pass rates stayed between 14% and 20%, peaking at 2,000 samples (20%) and dropping to 14% at 15,000 samples.
- Because GPT-2 is not instruction-tuned, distillation improves fluency but not rule-following.
- More data makes the model sound better without becoming more obedient, explaining the plateau and decline.

Results – Efficiency

- **The Scaling Reality:** As we move from 500 to 15,000 samples, training time increases linearly; however, instruction-following "intelligence" often exhibits diminishing returns.

- **The Student (500) Model:**
  - **Fastest training, minimal data cost**
  - **Speed:** ~126.9 tokens per second.
  - **Memory Size:** Only ~475.2 MB.

- **The Trade-off:** We trade a significant amount of "intelligence" (instruction following) for a model that can run locally on almost any hardware with near-instant response times.

Results – Qualitative Comparison

Please see Google Colab for the prompt-answer pairs for each student model.

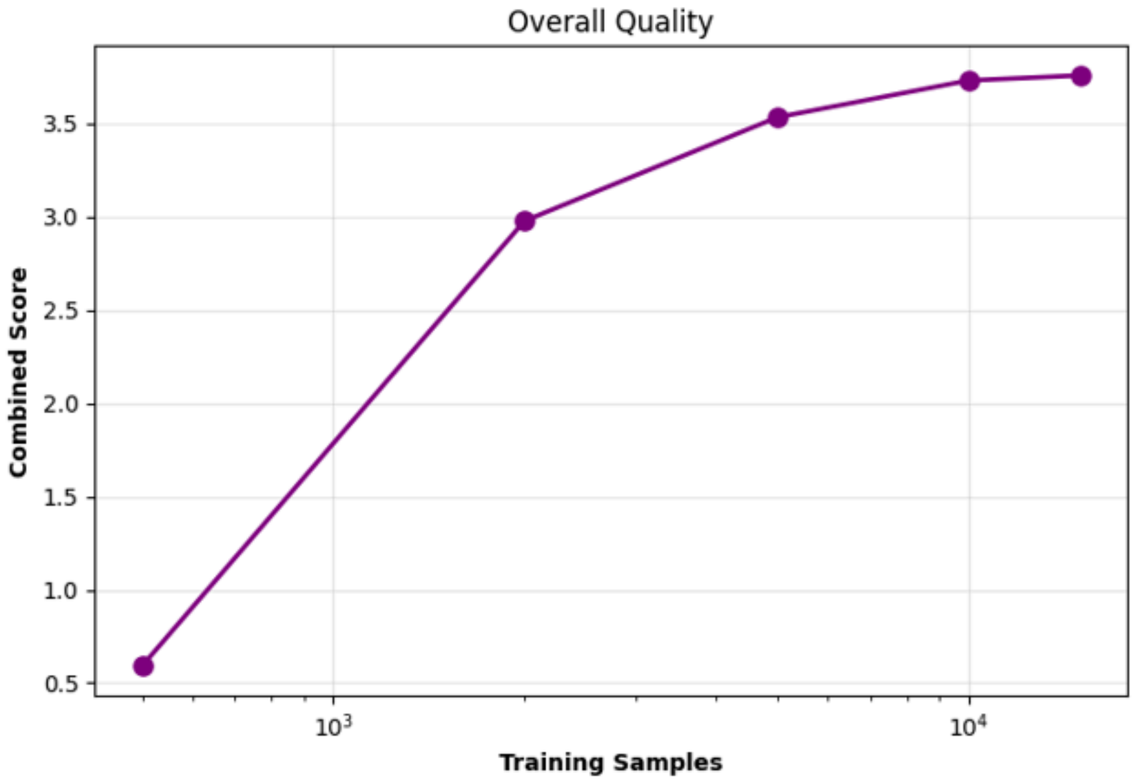
Interpretations:

1) As teacher data increases, responses become more coherent, better structured, and more aligned with teacher-like instruction style.

2) More data improved the format (the list structure), but didn't necessarily improve the factual depth, highlighting the physical limits of the 124M parameter size.

Qualitative results support the scaling trend: more distilled data → more stable instruction-following.

Discussion



- **Diminishing returns:** For a 124M model, the performance sweet spot ≈ 5,000 samples; beyond this, the model's limited capacity prevents learning much more.
- **Style vs. substance:** Distillation transfers how the teacher talks (fluency and phrasing) far better than how it reasons (complex rules and constraints).
- **Hardware efficiency:** The 1.5B teacher needed hours to generate data, while the 124M student trained on it in minutes and runs much faster and cheaper at inference.

takes the Perplexity (normalized so higher is better) and the Adherence Rate (0 to 1 scale) and averages them

Conclusion

In summary, data scaling significantly improves linguistic quality, but this improvement plateaus quickly for small models.

In short, focus on high-quality, diverse data rather than sheer volume when distilling into compact models.

Thank you for listening.

Extra: Evaluation Metrics

$$\mathrm{PPL} = \exp\left(\frac{\sum_i \ell_i \cdot N_i}{\sum_i N_i}\right) = \exp\left(\frac{\sum \text{loss} \cdot \#\text{valid tokens}}{\sum \#\text{valid tokens}}\right)$$

$$\text{tokens/sec} = \frac{128}{\text{avg latency}}$$

$$\mathrm{TokPerSec} = \frac{N_{\text{gen}}}{\overline{T}}$$

$$N_i = \sum \mathbf{1}\{y \neq -100\}$$

$$\overline{T} = \frac{1}{K}\sum_{k=1}^{K} T_k$$

$$\mathrm{PassRate} = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\{c_j(\hat{r}_j) = 1\}$$

$$\text{size (MB)} = \sum (\#\text{params} \times \text{bytes per param})/1024^2$$

$$\mathrm{ModelSize}_{\mathrm{MB}} = \frac{\sum_{p\in\theta} \left(\mathrm{numel}(p) \cdot \mathrm{bytes}(p)\right)}{1024^2}$$

$$\mathrm{Combined} = w_1 \cdot \frac{1}{\mathrm{PPL}} + w_2 \cdot \mathrm{PassRate} + w_3 \cdot \mathrm{TokPerSec}$$

$$w_1 + w_2 + w_3 = 1, \quad w_i \geq 0$$