STAT 992: Foundation Models for Biomedical Data

Ben Lengerich

Lecture 09: Unsupervised training of LLMs

February 23, 2026

Today

- Let’s outline our review paper
- Sign up for section ownership
- For reference, our paper interest sign-up sheet

Ben Lengerich © University of Wisconsin-Madison 2026

Last Time: From GPT-1 to GPT-4

- **Architecture:**
  - **Scale:** Variety of options, with biggest (1.5B params → >1T params):
    - Block size (max context): 512 → 128k
    - Layers: 12 → >96
    - Attention Heads: 12 → >96
    - Embedding Dim: 768 → >12,288
    - Vocab: 40k → >50k tokens
  - Tokenizer: Includes image patches for multimodal
  - **Mixture-of-Experts**
- **Training:**
  - Dataset: BookCorpus (5GB) → Private 13T tokens (~50TB)
  - Reinforcement learning for alignment

Ben Lengerich © University of Wisconsin-Madison 2026

Today

- Unsupervised training of LLMs
  - Emergent Capabilities
  - Challenges of MLE-based unsupervised training

Ben Lengerich © University of Wisconsin-Madison 2026

Modern Training Pipeline

- Pipeline
  - ✓ **Pretraining**
  - ✓ Domain Adaptation
  - ✓ Alignment
  - ✓ Instruction Tuning
  - ✓ Preference Optimization
  - ✓ RLHF / DPO

Ben Lengerich © University of Wisconsin-Madison 2026

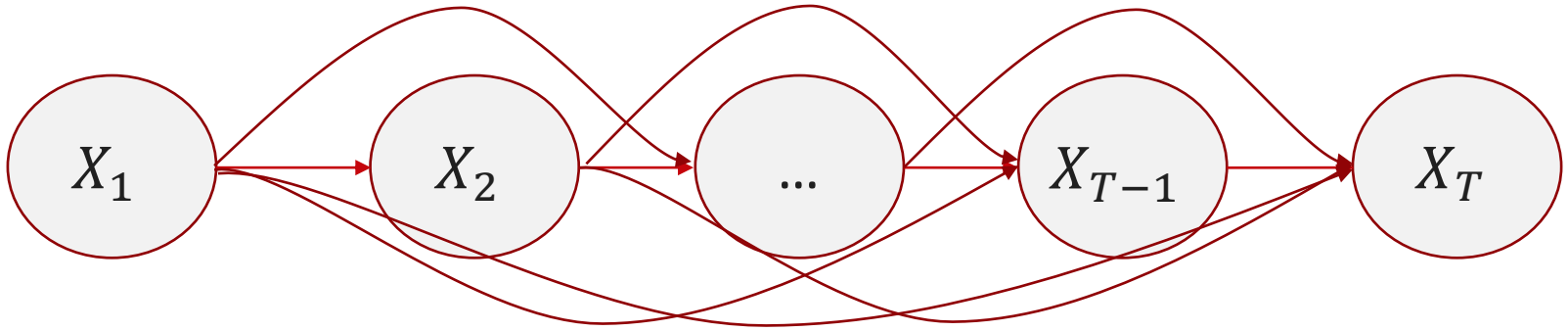
Unsupervised Training (“Pre-training”) of LLMs

The Pretraining Problem

- **Goal:** Learn a model of language from raw text
  - **Probabilistic objective:** Max log-likelihood of observed seqs

$$\max_{\theta} \sum_{i} \sum_{t} \log P_{\theta}\left(X_{i,t} \mid X_{i,<t} \right)$$

- **Auto-regressive PGM:**



Ben Lengerich © University of Wisconsin-Madison 2026

Why next-token prediction is powerful

- Language contains latent structure
- Next-token prediction forces models to infer that structure
- This produces general-purpose representations

Ben Lengerich © University of Wisconsin-Madison 2026

Why next-token prediction is powerful

- ”The simplest way to predict the next token is to understand what happened throughout the context.”
- To predict the word "*is*" in "*The capital of France ___ Paris.*", the model must:
  - Resolve subject-verb agreement
  - Recognize a factual structure
  - Know the topic is geography

Ben Lengerich © University of Wisconsin-Madison 2026

We've had MLE-based Language Models for a while…

Large Language Models in Machine Translation 2007

Thorsten Brants **Ashok C. Popat** **Peng Xu** **Franz J. Och** **Jeffrey Dean**

Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94303, USA
{brants,popat,xp,och,jeff}@google.com

Some fun:
https://github.com/LRitzdorf/TheJeffDeanFacts

We've had MLE-based Language Models for a while…

Large Language Models in Machine Translation 2007

ɔk C. Popat Peng Xu Franz J. Och Jeffrey Dean

Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94303, USA
popat,xp,och,jeff}@google.com

This paper reports on the benefits of large-scale statistical language modeling in machine translation. A distributed infrastructure is proposed which we use to train on up to 2 trillion tokens, resulting in language models having up to 300 billion n-grams. It is capable of providing smoothed probabilities for fast, single-pass decoding. We introduce a new smoothing method, dubbed *Stupid Backoff*, that is inexpensive to train on large data sets and approaches the quality of Kneser-Ney Smoothing as the amount of training data increases.

$$P(w_1^L) = \prod_{i=1}^{L} P(w_i|w_1^{i-1}) \approx \prod_{i=1}^{L} \hat{P}(w_i|w_{i-n+1}^{i-1})$$

We've had MLE-based Language Models for a while…

- Why wasn't this the LLM moment?

Key change: representation learning through shared functional parameterization

n-gram Models	LLMs
$P(w_t = v \mid w_{t-n+1:t-1}) = \theta_{c,v}, \quad c = w_{t-n+1:t-1}$	$P(w_t = v \mid w_{<t}) = softmax(Wh_t), \quad h_t = f_\theta(e(w_{<t}))$
Tokens used as indexes into tables: $\theta_{c,v} = \frac{count(c,v)}{count(c)}$	Tokens mapped to vectors via embedding matrix
Context length fixed as table dimensionality	Dynamic hierarchical structure via stacked transformer blocks
No parameter sharing across contexts	**Shared parameters** across all contexts
Generalization only via **smoothing / backoff**	Generalization through **representation similarity**

Representation Learning

Contextual Representations

- $x_i = e(w_i)$
- $h_t = f_\theta(x_1, \ldots, x_{t-1})$

- Representations encode:
  - syntax
  - entities
  - relations
  - task structure

Ben Lengerich © University of Wisconsin-Madison 2026

Scale & Emergent Capabilities

What happens as we scale training?

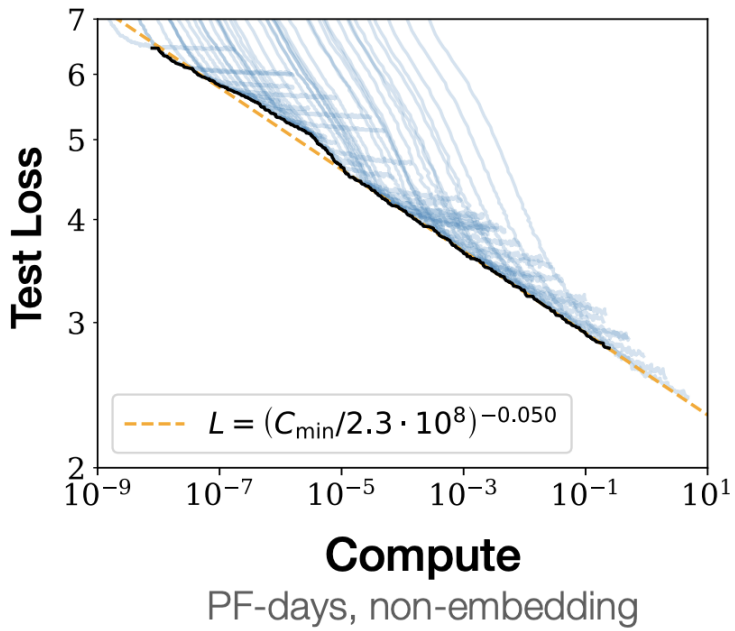


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

"Scaling Laws for Neural Language Models". Kaplan et al 2021

What happens as we scale training?

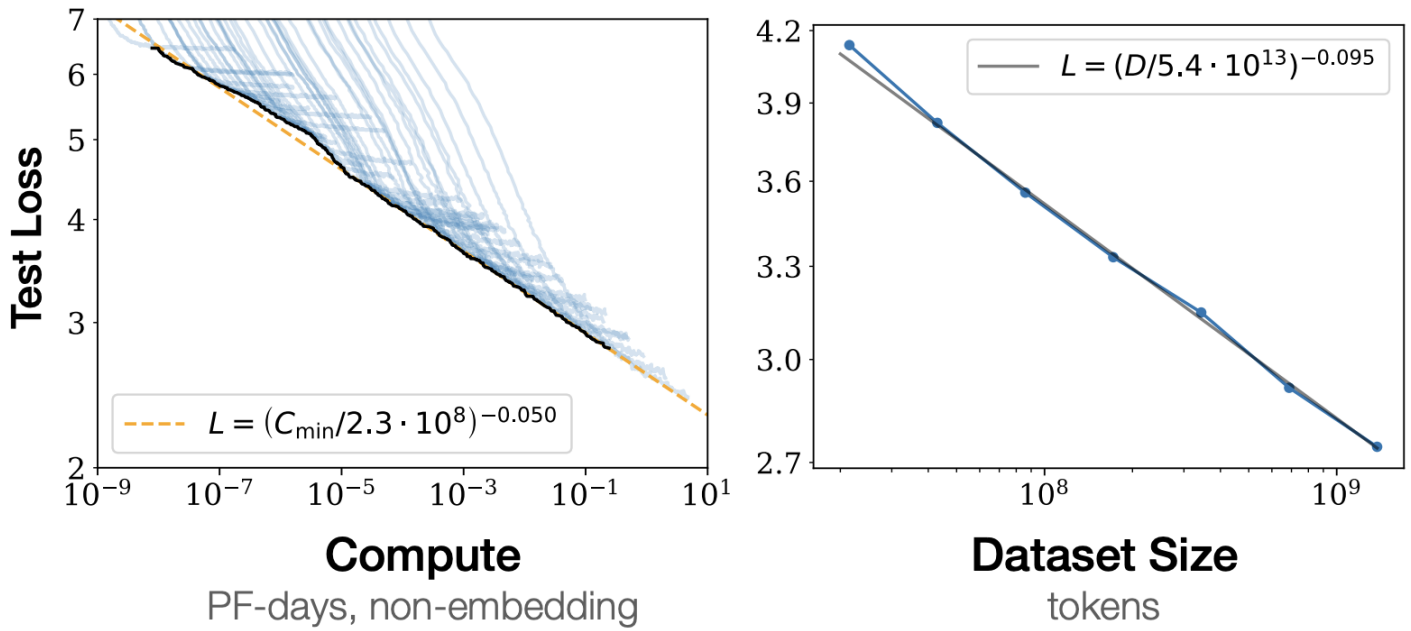


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

"Scaling Laws for Neural Language Models". Kaplan et al 2021

What happens as we scale training?

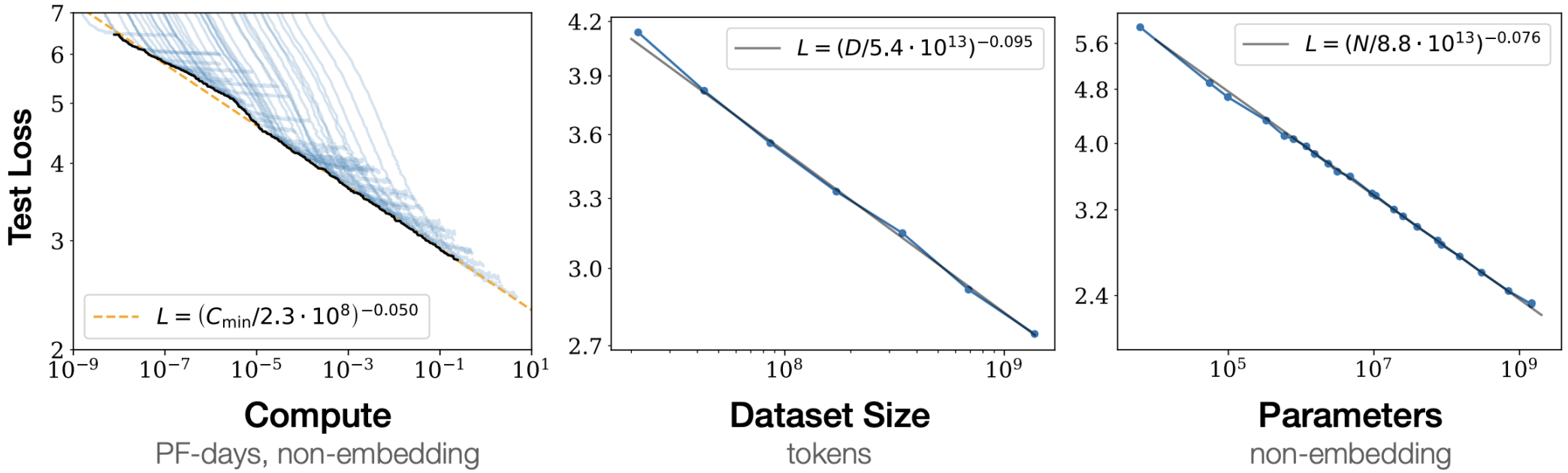
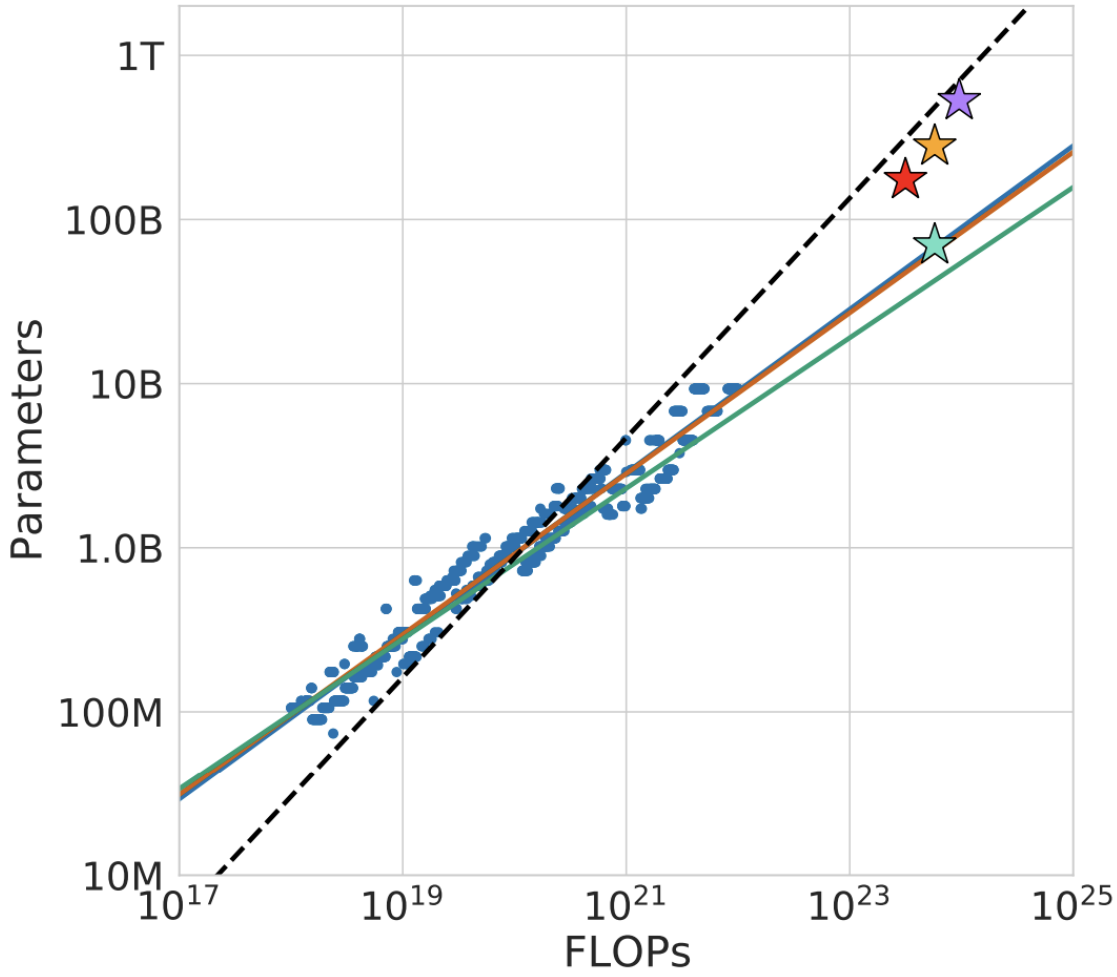


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

"Scaling Laws for Neural Language Models". Kaplan et al 2021

What happens as we scale training (another view)



Approach 1
Approach 2
Approach 3
Kaplan et al (2020)

Chinchilla (70B)
Gopher (280B)
GPT-3 (175B)
Megatron-Turing NLG (530B)

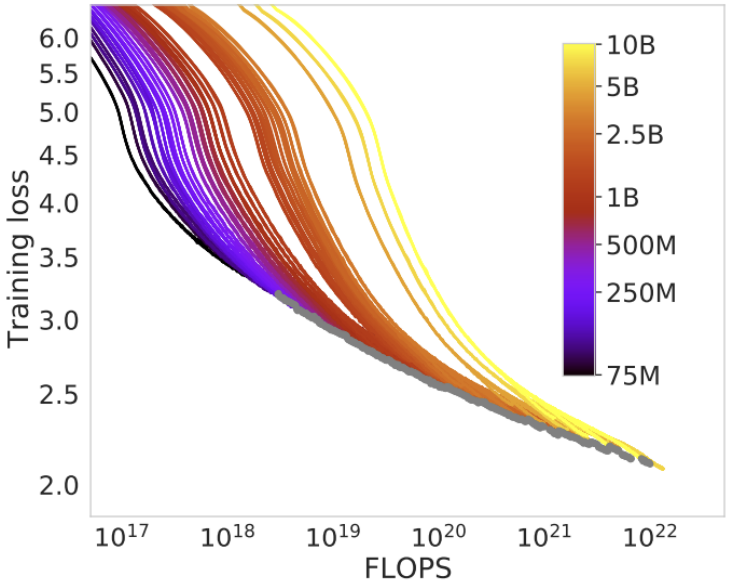
Turns out, most LLMs before 2022 were under**trained**.

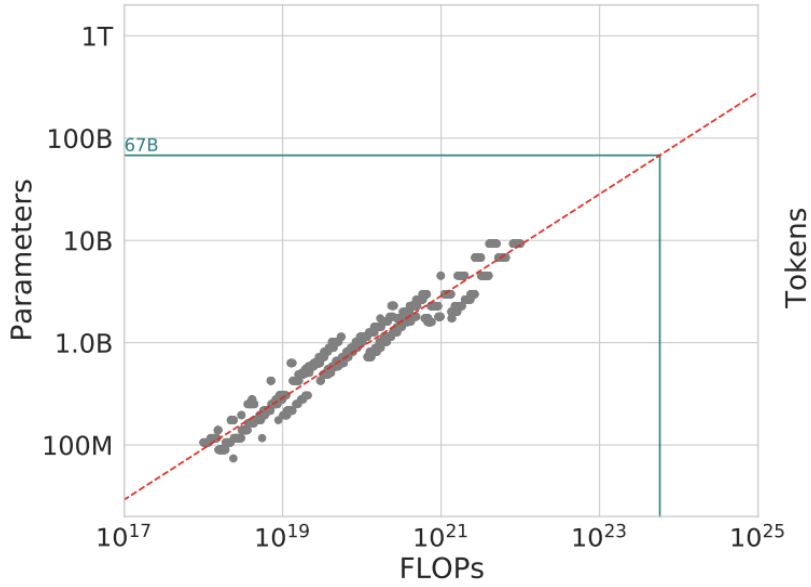
"Training Compute-Optimal Large Language Models". Hoffmann et al 2022

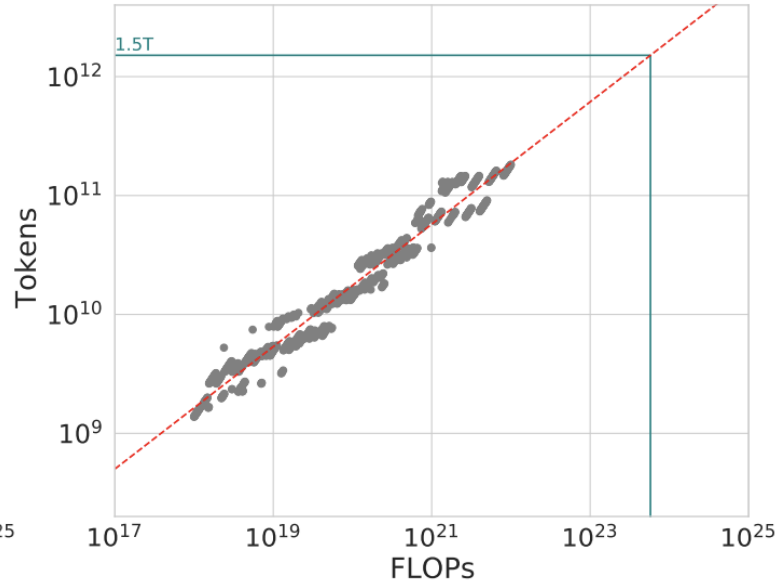
What happens as we scale training (another view)

Training compute = parameters * tokens

Kaplan (2020): Increase parameters
Chinchilla (2022): Parameters ∝ Tokens



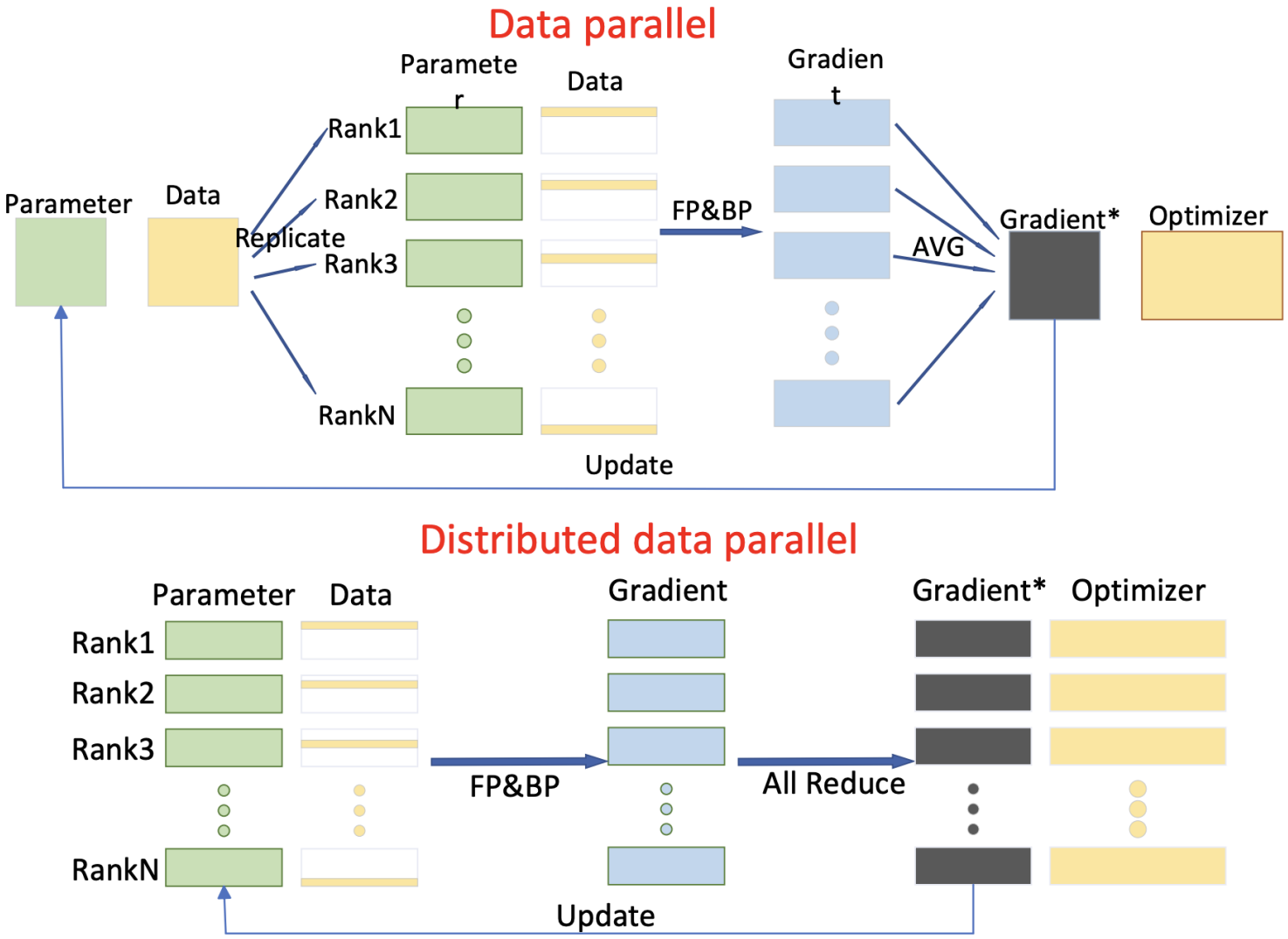




"Training Compute-Optimal Large Language Models". Hoffmann et al 2022

Training at Scale

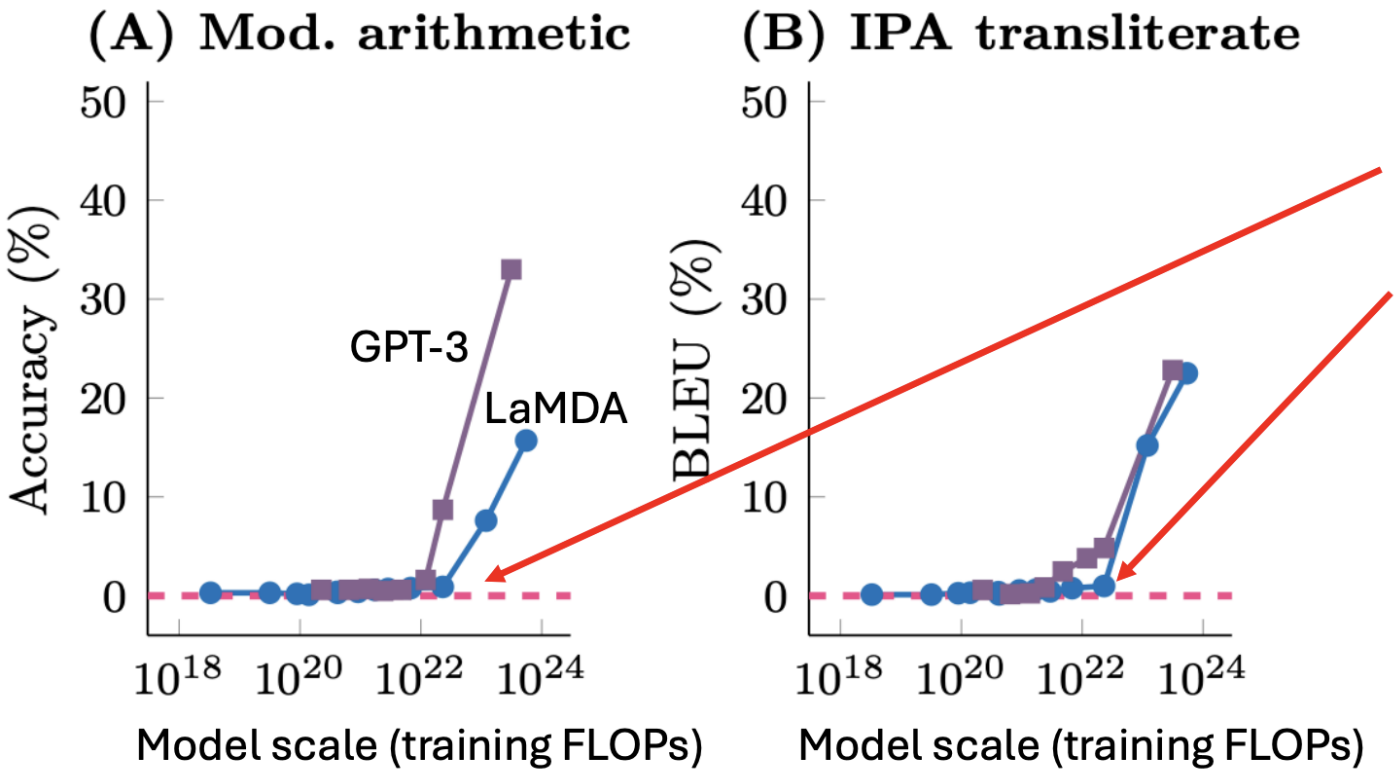
Parallel training



[Understanding LLMs: A Comprehensive Overview from Training to Inference]

Ben Lengerich © University of Wisconsin-Madison 2026

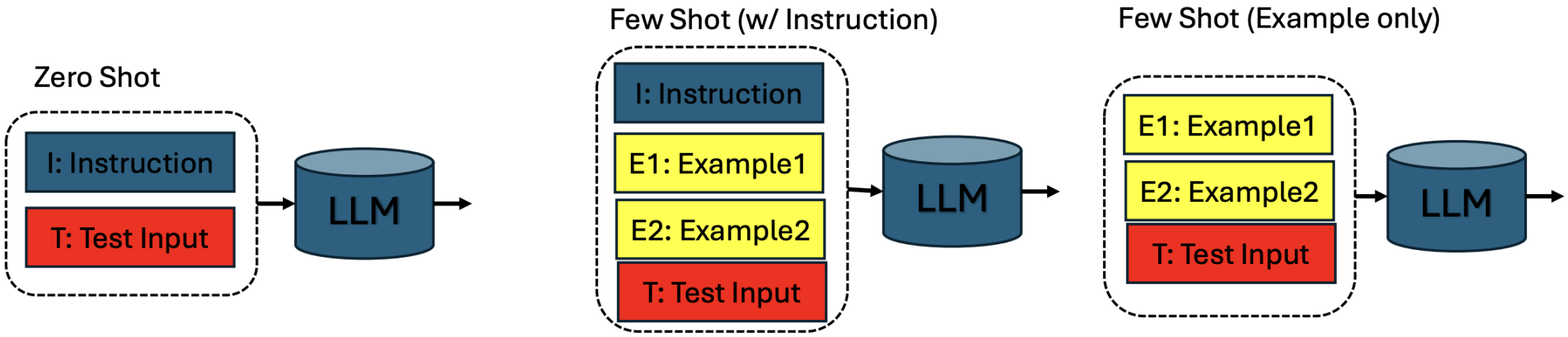
Smooth improvements → sharp emergent ability?



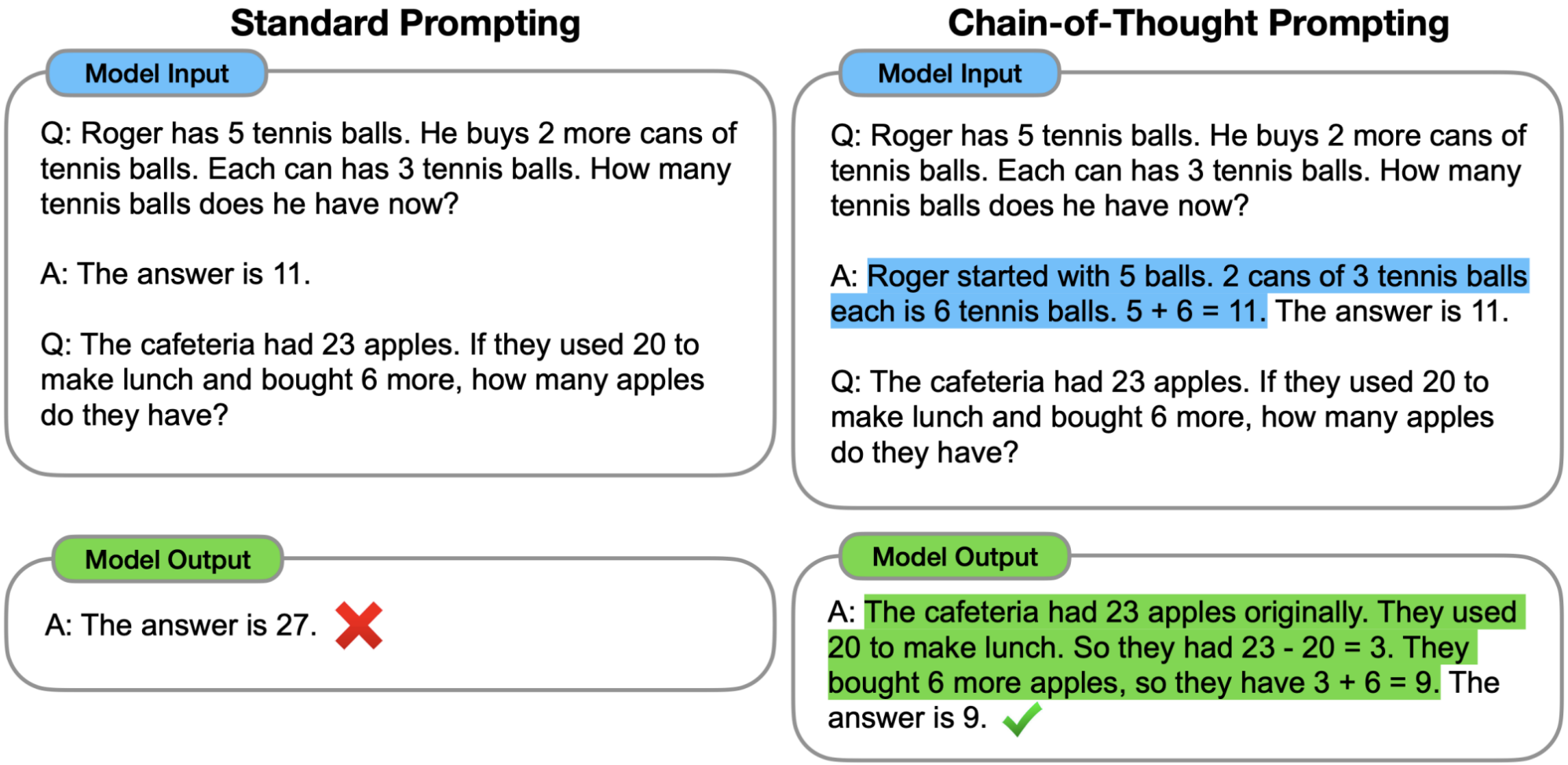
An ability is emergent if it is not present in smaller models but is present in larger models [Wei, et al (2022). Emergent Abilities of Large Language Models

Example of emergence: In-Context Learning

I: Instruction	Translate English to French
E1: Example1	[en]: A discomfort which lasts. [fr]: Un malaise qui dure
E2: Example2	[en]: HTML is a language for formatting formatage [fr]: HTML est un langage de
T: Test Input	[en]: After you become comfortable with formatting [fr]:



Example of emergence: Chain-of-Thought



Wei, et al. (2023) Chain-of-Though Prompting Elicits Reasoning in LLMs

Ben Lengerich © University of Wisconsin-Madison 2026

Why does this work? Some hypotheses

- Task identification?
  - Xie et al. (2021). An explanation of in-context learning as implicit Bayesian inference
  - Raventos, et al. (2023). Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression
- Some kind of ”learning” without model updates?
  - Akyurek, et al. (2024). In-context language learning: architectures and algorithms
  - von Oswald, et al. (2023). Transformers learn in-context by gradient descent
- Both?
  - Pan, et al. (2023). What in-context learning ”learns” in-context: disentangling task recognition and task learning

Why does this work? Some debate

Measurement artifact?

[Schaeffer et al 2023]

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo

Computer Science, Stanford University

Abstract

Recent work claims that large language models display *emergent abilities*, abilities not present in smaller-scale models that are present in larger-scale models. What makes emergent abilities intriguing is two-fold: their *sharpness*, transitioning seemingly instantaneously from not present to present, and their *unpredictability*, appearing at seemingly unforeseeable model scales. Here, we present an alternative explanation for emergent abilities: that for a particular task and model family, when analyzing fixed model outputs, emergent abilities appear due the researcher's choice of metric rather than due to fundamental changes in model behavior with scale. Specifically, nonlinear or discontinuous metrics produce apparent emergent abilities, whereas linear or continuous metrics produce smooth, continuous, predictable changes in model performance. We present our alternative explanation in a simple mathematical model, then test it in three complementary ways: we (1) make, test and confirm three predictions on the effect of metric choice using the InstructGPT/GPT-3 family on tasks with claimed emergent abilities, (2) make, test and confirm two predictions about metric choices in a meta-analysis of emergent abilities on BIG-Bench; and (3) show how to choose metrics to produce never-before-seen seemingly emergent abilities in multiple vision tasks across diverse deep networks. Via all three analyses, we provide evidence that alleged emergent abilities evaporate with different metrics or with better statistics, and may not be a fundamental property of scaling AI models.

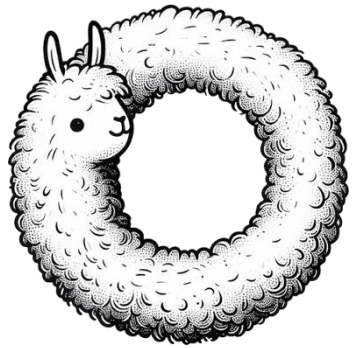
Data: The Real Bottleneck

Modern pre-training datasets

- FineWeb
- Dolma
- RedPajama
- SlimPajama
- Cosmopedia (synthetic)

Dataset composition strongly affects model behavior

TxT360



LLM360

Synthetic Data

Modern LLMs train on large amounts of synthetic data.

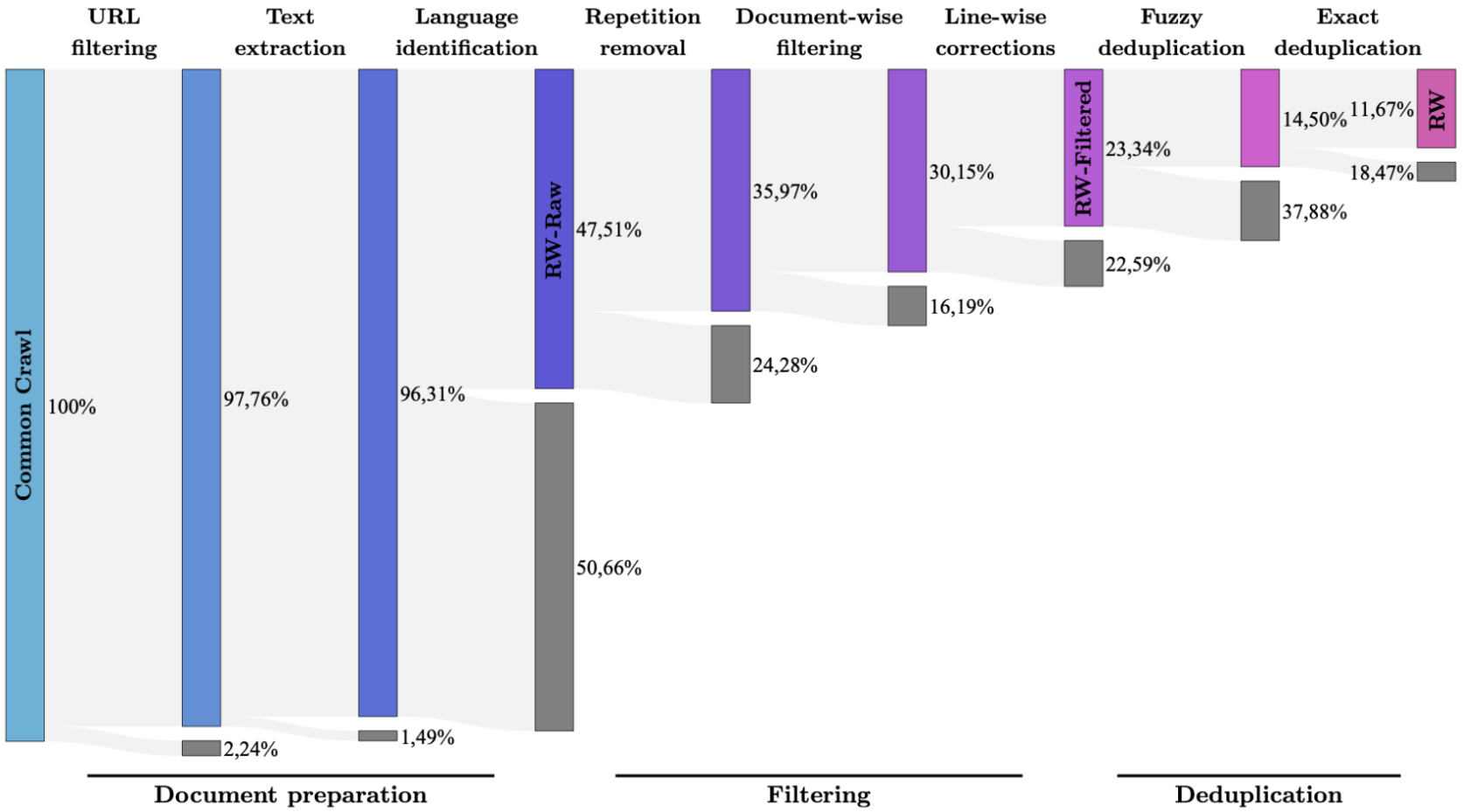
Examples:

- reasoning datasets
- synthetic textbooks
- distillation datasets

Synthetic data expands the training distribution.

Ben Lengerich © University of Wisconsin-Madison 2026

Data filtering

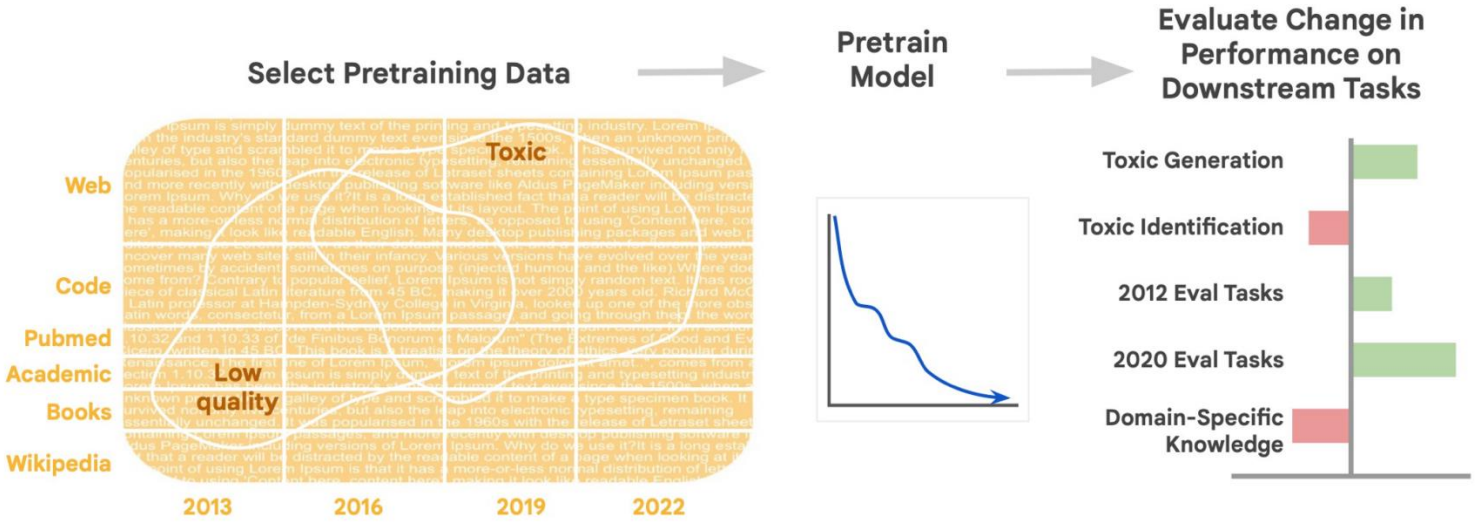


Penedo, et al. (2023) The Refined Web dataset for Falcon LLM

Ben Lengerich © University of Wisconsin-Madison 2026

Data filtering

A Pretrainer's Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity [Longpre et al, NAACL 2024]



Some findings: strongly encourage to read the paper!

- "temporal shift between evaluation data and pretraining data leads to performance degradation, which is not overcome by finetuning"
- "a trade-off between performance on standard benchmarks and risk of toxic generations… there does not exist a one-size-fits-all solution to filtering."

Challenges of MLE-based unsupervised training

MLE: A KL-Divergence view

$$KL(P_{data} \,||\, P_\theta) = E_{x \sim P_{data}}[-\log P_\theta(x)] + const.$$

$$\text{argmax}_\theta E_{x \sim P_{data}}[\log P_\theta(x)] = \text{argmin}_\theta KL(P_{data} \,||\, P_\theta)$$

- Must spread probability mass to cover observed sequences, even if incoherent
  - Repetition and genericity ("The man said the man said...")
  - Poor calibration on out-of-distribution prompts
  - Memorization of rare patterns
- Sampling strategies matter
- Entropy / confidence

Entropy and Confidence

- Token-level entropy:

$$H_t = -\sum_v P(x_t = v \mid x_{<t}) \log P(x_t = v \mid x_{<t})$$
$$= E_v[-\log P(x_t = v \mid x_{<t})]$$

- Low entropy = high confidence.

Do we want our model to be low or high entropy?

Ben Lengerich © University of Wisconsin-Madison 2026

Recall from Variational Inference

$$\log p(x \mid \theta) = E_{z \sim q}[\log p(x, z \mid \theta)] + H(q) + KL(q(z \mid x) \,||\, p(z \mid x, \theta))$$

$$\log p(x \mid \theta) \geq \underbrace{E_{z \sim q}[\log p(x, z \mid \theta)] + H(q)}$$

"ELBO": Evidence Lower Bound

Maximizing ELBO → Increased entropy of $q(z)$

Does Maximizing Likelihood → Increased entropy of $p(x \mid \theta)$?

Ben Lengerich © University of Wisconsin-Madison 2026

MLE and Entropy

$$\widehat{\theta_{MLE}} = \text{argmax}_\theta E_{x \sim P_{data}}[\log P_\theta(x)]$$

- Directly optimize the data distribution $P_\theta(x)$ without explicitly introducing auxiliary variables or explicitly controlling the entropy of x
- Often implicitly pushes $P_\theta(x)$ toward low-entropy distributions
- Mode collapse / degeneration / “memorization”

Ben Lengerich © University of Wisconsin-Madison 2026

MLE and Entropy

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

THE CURIOUS CASE OF NEURAL TEXT DeGENERATION [Holtzman 2020]

MLE and Entropy

$$\widehat{\theta_{MLE}} = \text{argmax}_\theta E_{x \sim P_{data}}[\log P_\theta(x)]$$

- Solutions:
  - Explicitly add a penalty for low entropy:

$$\hat{\theta} = \text{argmax}_\theta E_{x \sim P_{data}}[\log P_\theta(x)] + \lambda \sum_t H[P_\theta(x_t \mid x_{<t})]$$

  - Smooth labels:

$$\tilde{y_\epsilon} = \begin{cases} 1-\epsilon, & y = 1 \\ \dfrac{\epsilon}{|V|-1}, & y = 0 \end{cases}$$

  - Contrastive losses:
  - Scheduled sampling / noise contrastive objectives
  - Risk minimization, utility, preference-based losses

Ben Lengerich © University of Wisconsin-Madison 2026

What does MLE not do?

- No **semantics**
- No **task goals**
- No **explicit reward**

Ben Lengerich © University of Wisconsin-Madison 2026

Why alignment is needed

- MLE optimizes likelihood of text.
- But we want models that:
  - follow instructions
  - avoid harmful outputs
  - reason step-by-step
- Alignment adds new objectives on top of pretraining.

Ben Lengerich © University of Wisconsin-Madison 2026

Modern Training Pipeline

- Pipeline
  - ✓ **Pretraining**
  - ✓ Domain Adaptation
  - ✓ Alignment
  - ✓ Instruction Tuning
  - ✓ Preference Optimization
  - ✓ RLHF / DPO

Ben Lengerich © University of Wisconsin-Madison 2026

A good resource

https://huggingface.co/spaces/HuggingFaceTB/smol-training-playbook

Another good resource

frontier model training methodologies

Jan 31, 2026 • Alex Wa

Share on:

How do labs train a frontier, multi-billion parameter model? We look towards seven open-weight frontier models: Hugging Face's SmolLM3, Prime Intellect's Intellect 3, Nous Research's Hermes 4, OpenAI's gpt-oss-120b, Moonshot's Kimi K2, DeepSeek's DeepSeek-R1, and Arcee's Trinity series. This blog is an attempt at distilling the techniques, motivations, and considerations used to train their models with an emphasis on training methodology over infrastructure.

These notes are largely structured based on Hugging Face's SmolLM3 report due to its extensiveness, and it is currently supplemented with notes from other reports including Intellect-3, gpt-oss-120b, Hermes 4, DeepSeek, and Kimi. While this blog explores some infrastructure-related ideas like in-flight weight updates and multi-client orchestrators, there are many other ideas mentioned throughout those posts/blogs like expert parallelism and quantization. Hugging Face writes more about gpt-oss-120b's infrastructure here.

https://djdumpling.github.io/2026/01/31/frontier_training.html

Key Takeaways

- Pretraining uses **next-token prediction**
- Scaling enables **representation learning**
- Data quality and mixture matter enormously
- MLE has limitations
- Modern systems combine **pretraining + alignment**

Ben Lengerich © University of Wisconsin-Madison 2026

Questions?