CS11-711 Advanced NLP

Pretraining

Sean Welleck

Carnegie Mellon University
Language Technologies Institute

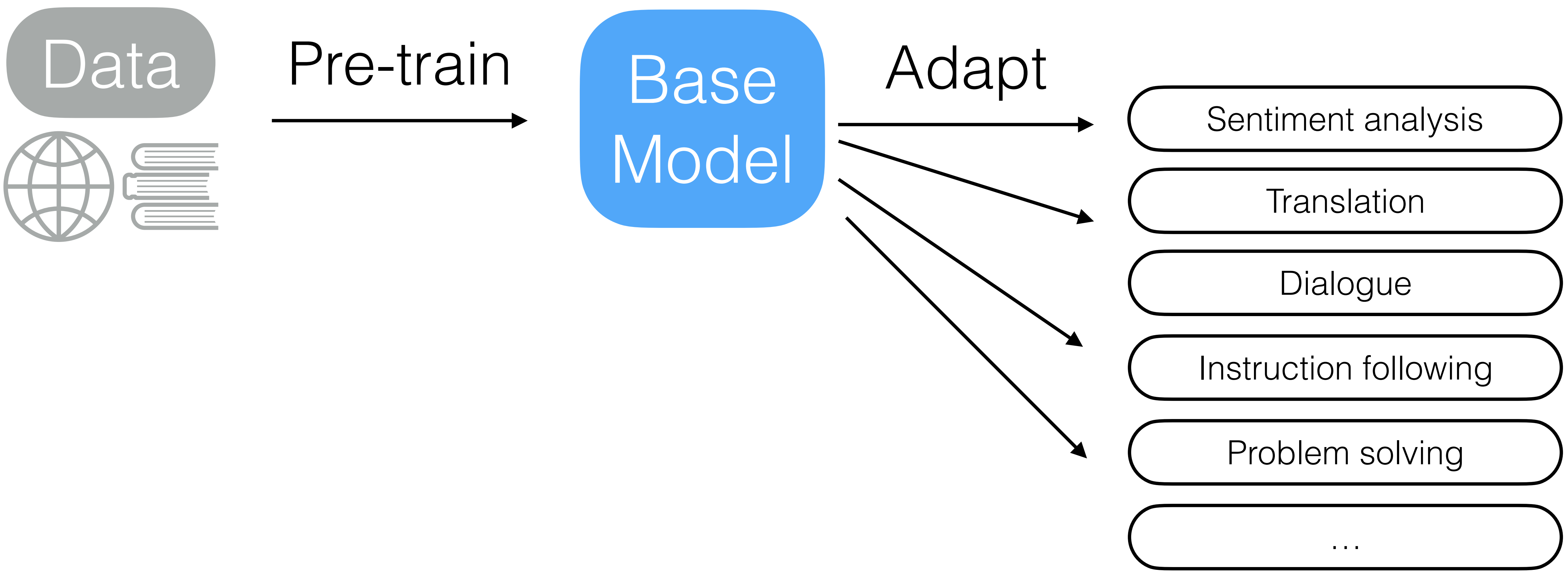
https://cmu-l3.github.io/anlp-spring2025/

https://github.com/cmu-l3/anlp-spring2025-code

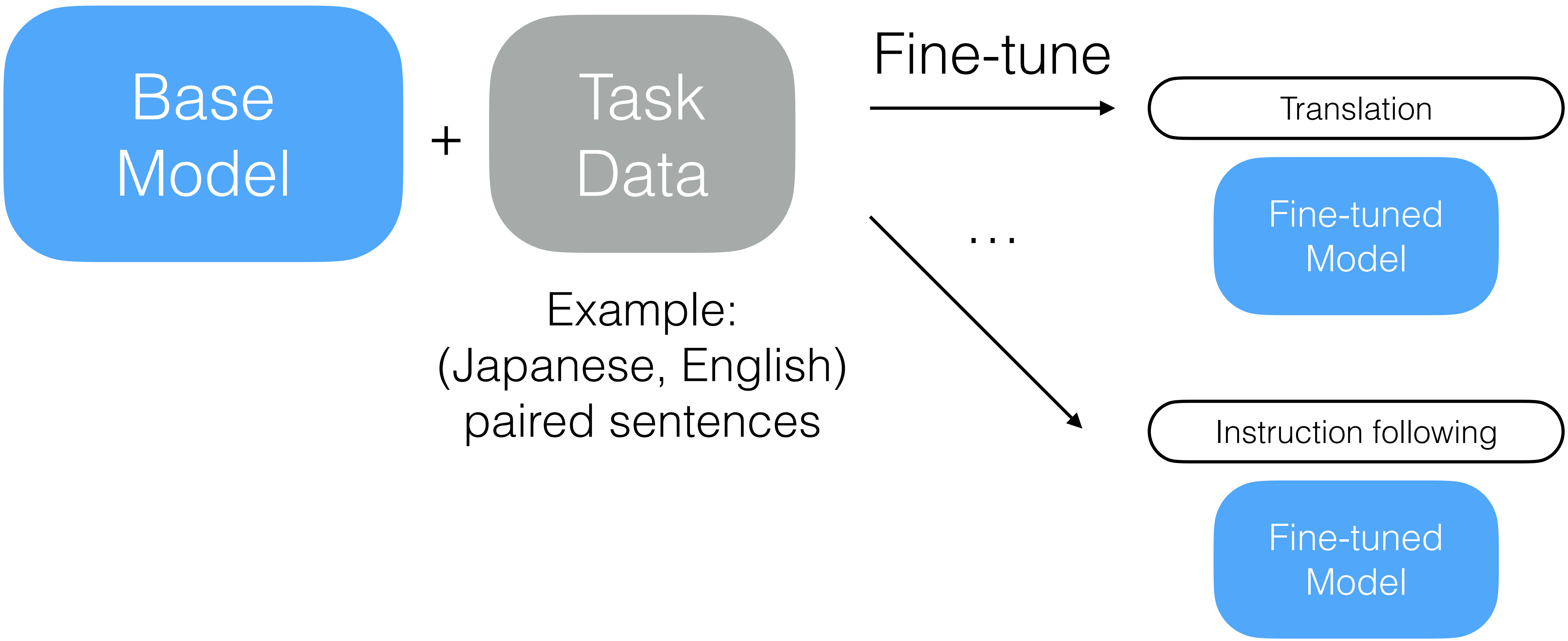
Recap

- Classification, sequence models, language models
- So far:
    - Train from scratch
    - 1 model, 1 task
- Today:
    - *Pretrain* a *single model*, adapt it to *many tasks*

Basic idea

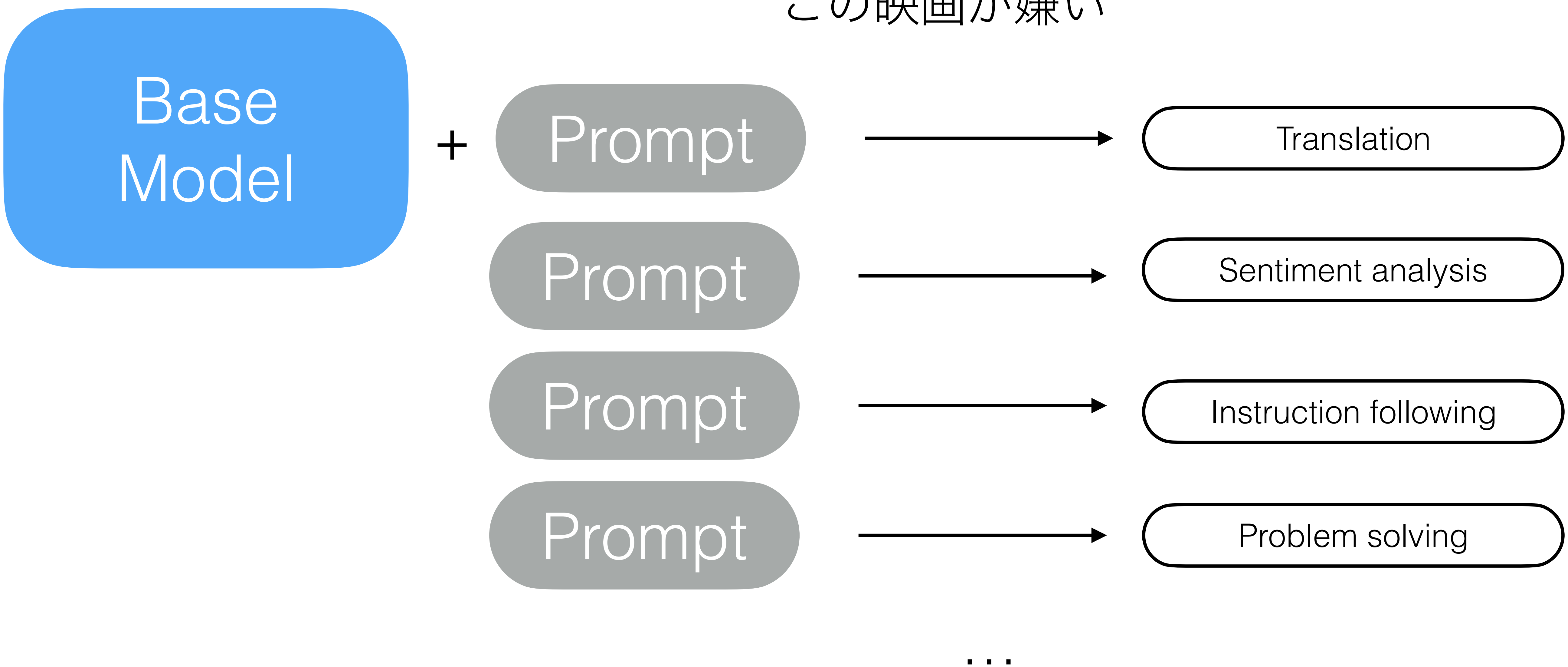

Data
Pre-train
Base Model
Adapt
Sentiment analysis
Translation
Dialogue
Instruction following
Problem solving
...

Adaptation: fine-tune



Adaptation: prompting

Example:
“Translate this sentence into English:
この映画が嫌い”



Why pre-train?

- Transfer learning: take “knowledge” from one task and apply it to another task
  - **Less task data**: use less data to reach a given level of performance
  - **Better task performance**: reach higher performance than training from scratch
  - **One model, multiple tasks**: convenient, amortizes cost, a starting point for many uses, …

Major factors

Major factors

- Pre-trained models have names like BERT, GPT-3, Llama, Deepseek-v3, …

- Each model is influenced by 4 major factors:
  - **Model:** The underlying neural network architecture
  - **Training objective:** The objective used to train
  - **Data:** the data used to train the model
  - **Hyperparameters**: e.g. learning rate, batch size

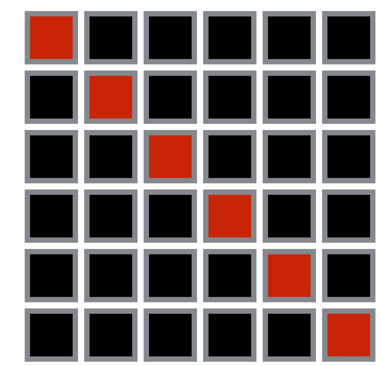
Which model?

- Usually **Transformer**, although the details vary
- **Size**: bigger usually has better task performance within a given model family
- **Model details** can vary (or are underspecified)

Which Objective?

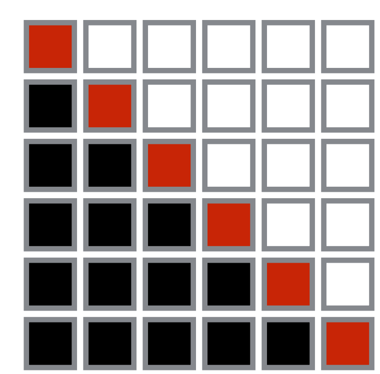
- **Masked language modeling:** used more for fine-tuning

$$P(X) \neq \prod_{i=1}^{|X|} P(x_i | x_{\neq i})$$



- **Auto-regressive language modeling:** used for fine-tuning, prompting. Extremely popular in 2025

$$P(X) = \prod_{i=1}^{|X|} P(x_i | x_1, \ldots, x_{i-1})$$



Which Data?

- Data is extremely important, common sources
  - [2018] **Books corpus** - a large corpus of books
  - [2018-] **Wikipedia**
  - **Common crawl** - data from the whole internet
  - …

Which hyper-parameters?

- Reported for some models
- We’ll go over strategies that are used to select some of them

Today’s lecture

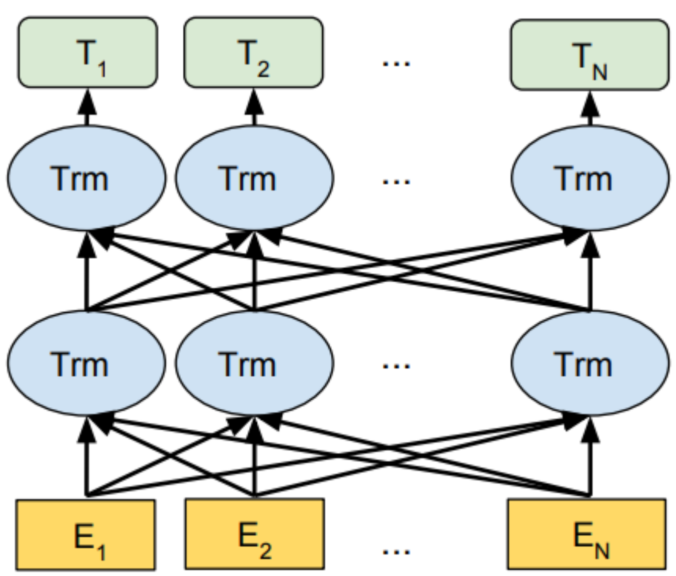
- Objectives
  - Masked language modeling
  - Autoregressive language modeling
- Data: sources, quality, and quantity
- Thinking about pretraining
  - Tokens, model size, compute
  - Scaling laws

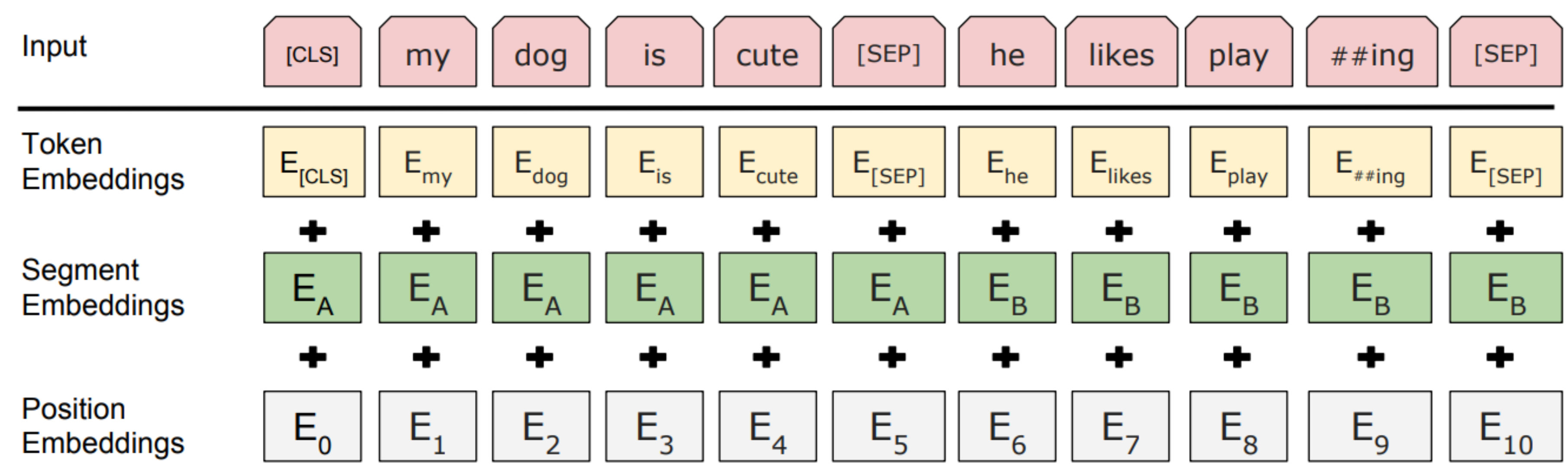
Today’s lecture

- Objectives
  - **Masked language modeling**
  - Autoregressive language modeling

Masked Language Modeling (BERT)
(Devlin et al. 2018)

- **Model:** Multi-layer self-attention. Input sentence or pair, w/ [CLS] token, subword representation





- **Objective:** Masked word prediction + next-sentence prediction
- **Data:** BooksCorpus + English Wikipedia

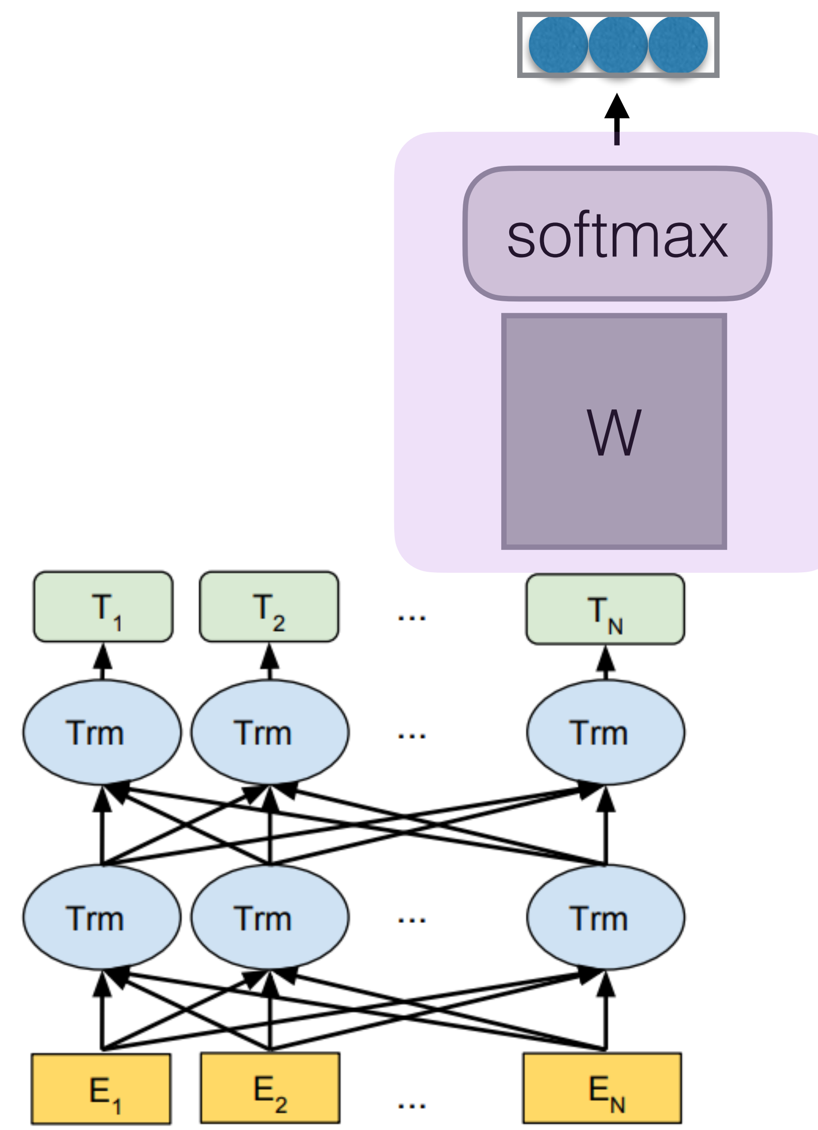
Masked Word Prediction
(Devlin et al. 2018)

Predict a masked word

- 80%: substitute input word with [MASK]
- 10%: substitute input word with random word
- 10%: no change

Adapting a masked language model

- Add an output layer that maps a hidden vector to scores
- Example:
  - Data: (movie review, {positive, neutral, negative})
  - Initialize the model with BERT
  - Train with cross-entropy loss



Today's lecture

- Objectives
  - Masked language modeling
  - **Autoregressive language modeling**

Autoregressive language modeling

$$\min_{\theta} \sum_{x \in D_{train}} \sum_{t} - \log p_{\theta}(x_t \mid x_{<t})$$

$$\text{minimize}_{\theta} \quad \text{CrossEntropyLoss}(\theta, D_{train}, \phi)$$

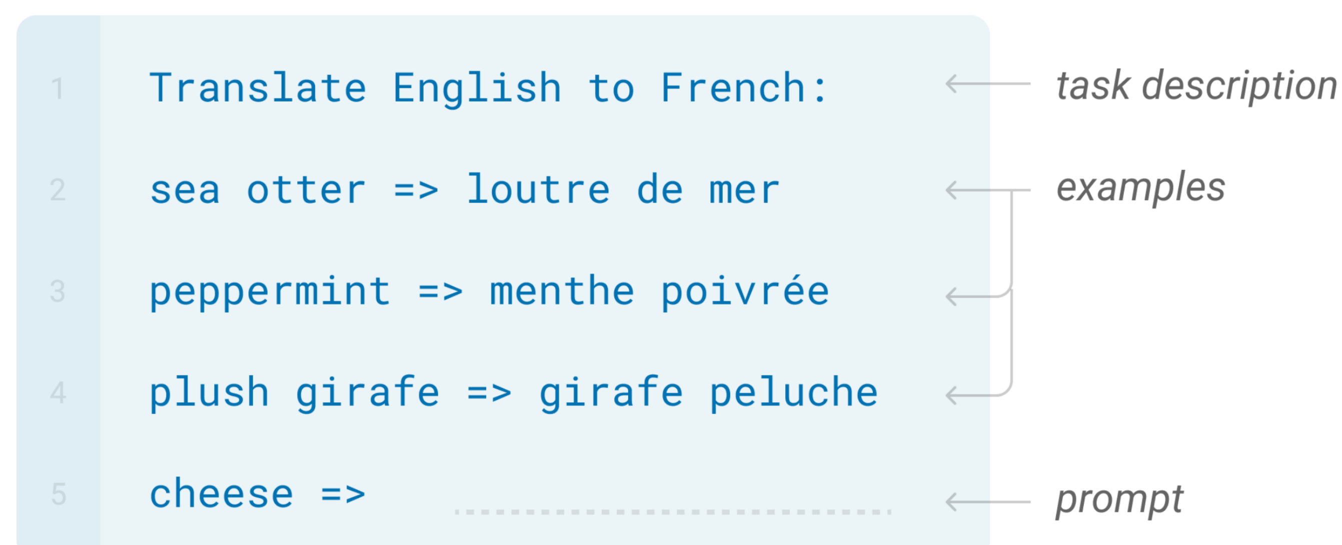
- We've seen this a few times now. Key inputs:
  - θ: model architecture and size
  - D_{train}: data (content, quantity)
  - ϕ: hyper-parameters, e.g. learning rate, batch size

Evaluating a model

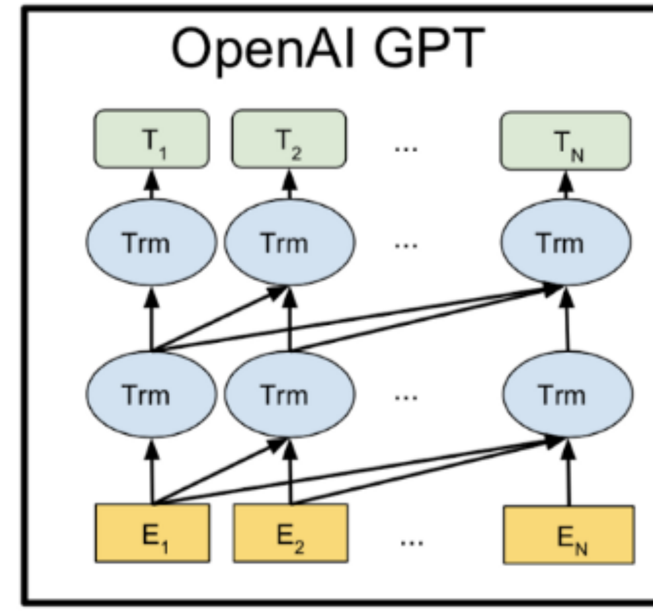
- Loss (training, validation, test)
    - Diagnose training trajectory, compare models in the same family
- Few-shot prompting
- Fine-tuning

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Example: GPT-2



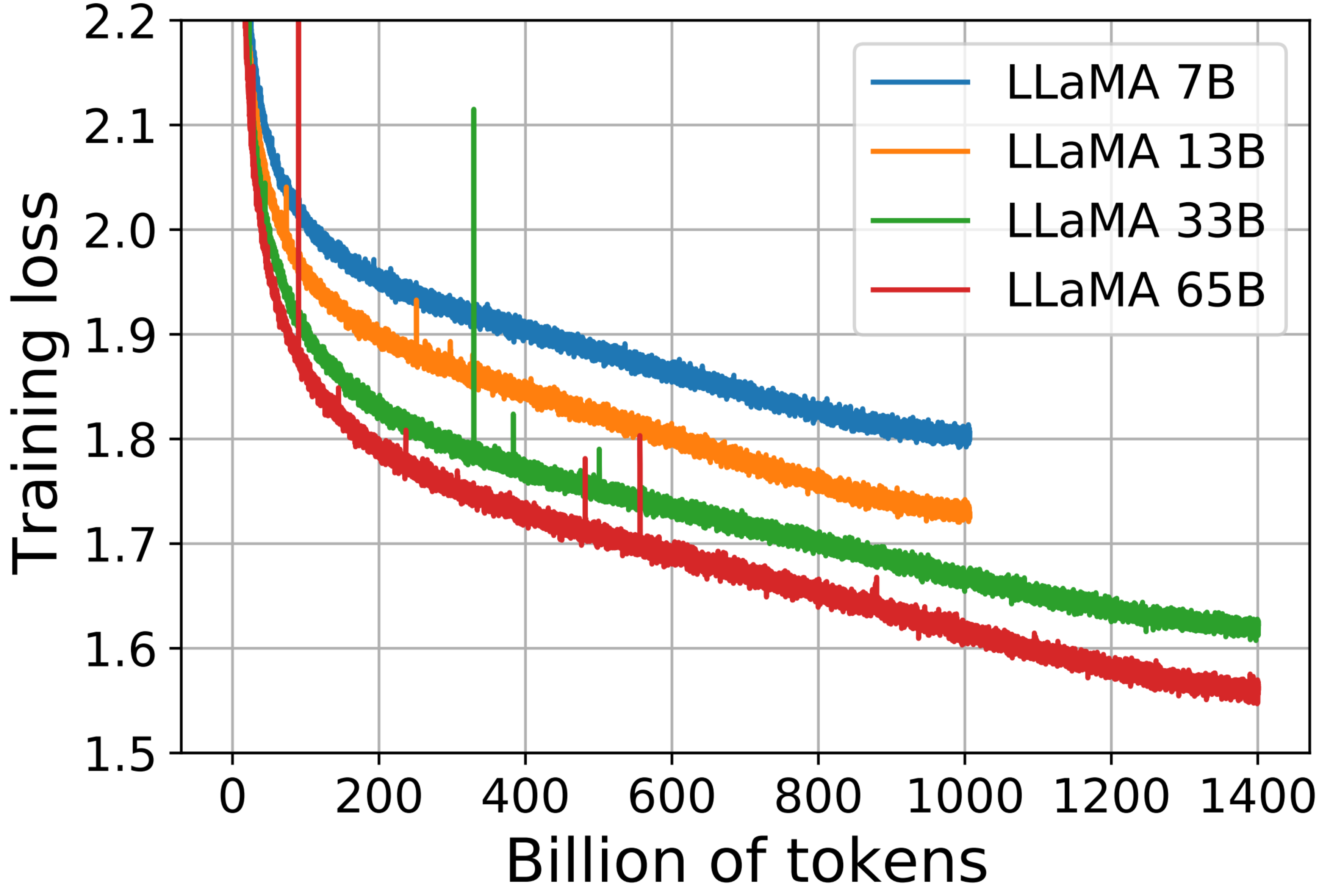
- **Model:** Transformer (1.5B)
- **Data:** WebText (millions of web pages)
- **Results:** Impressive results in generation of long-form text, and *zero shot* task completion

Example: Llama

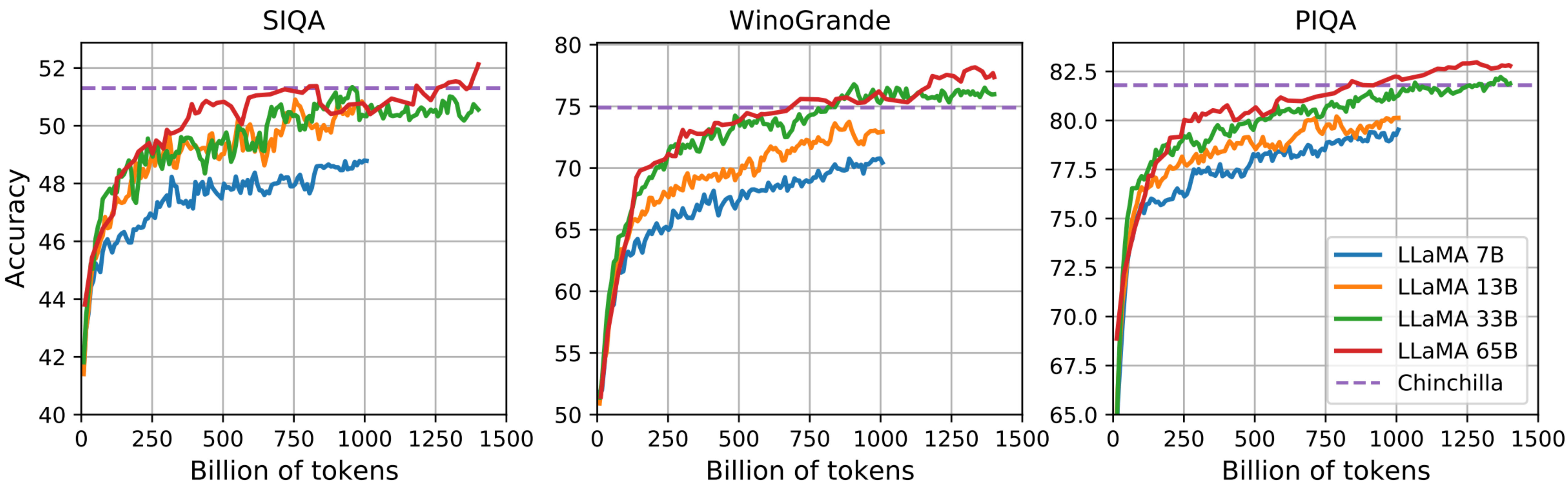
- **Model:** Transformer, {6.7B, 13B, 32B, 65B}
- **Data:** 1.4 trillion tokens, sources:

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Llama: training loss



Llama: few-shot performance trajectory

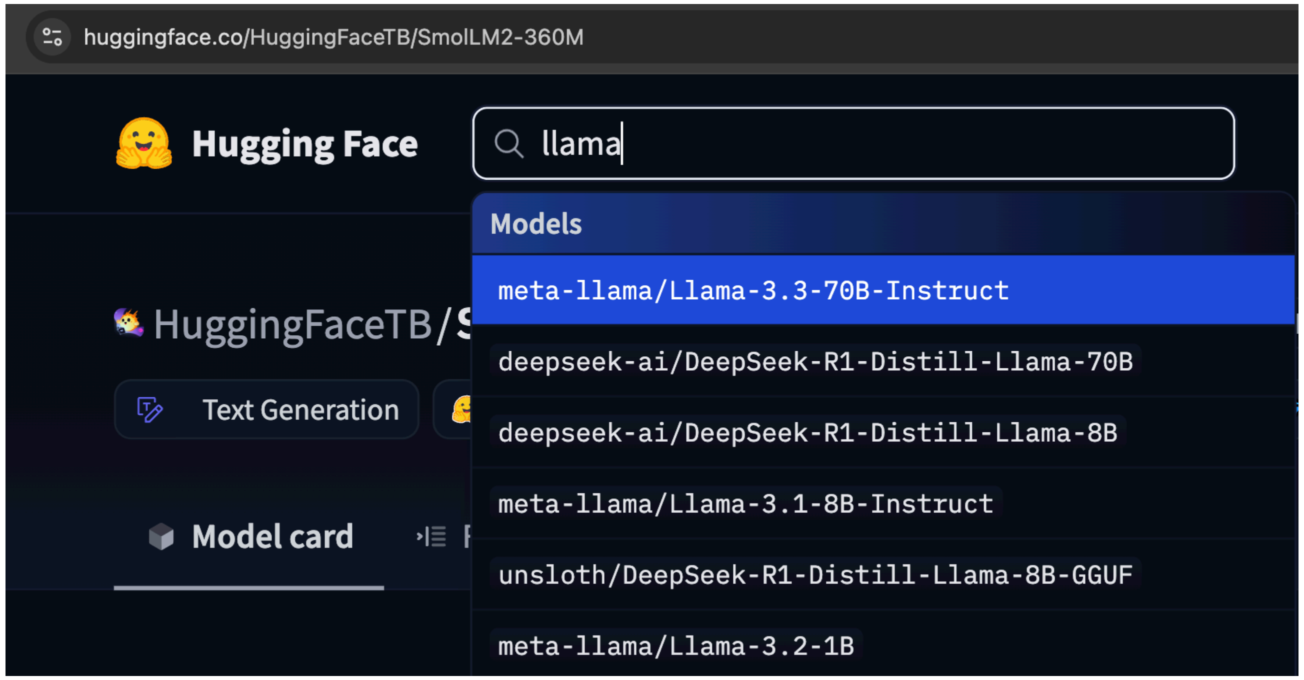


Practical tools: HuggingFace

Load tokenizer and model

- Find models at https://huggingface.co/

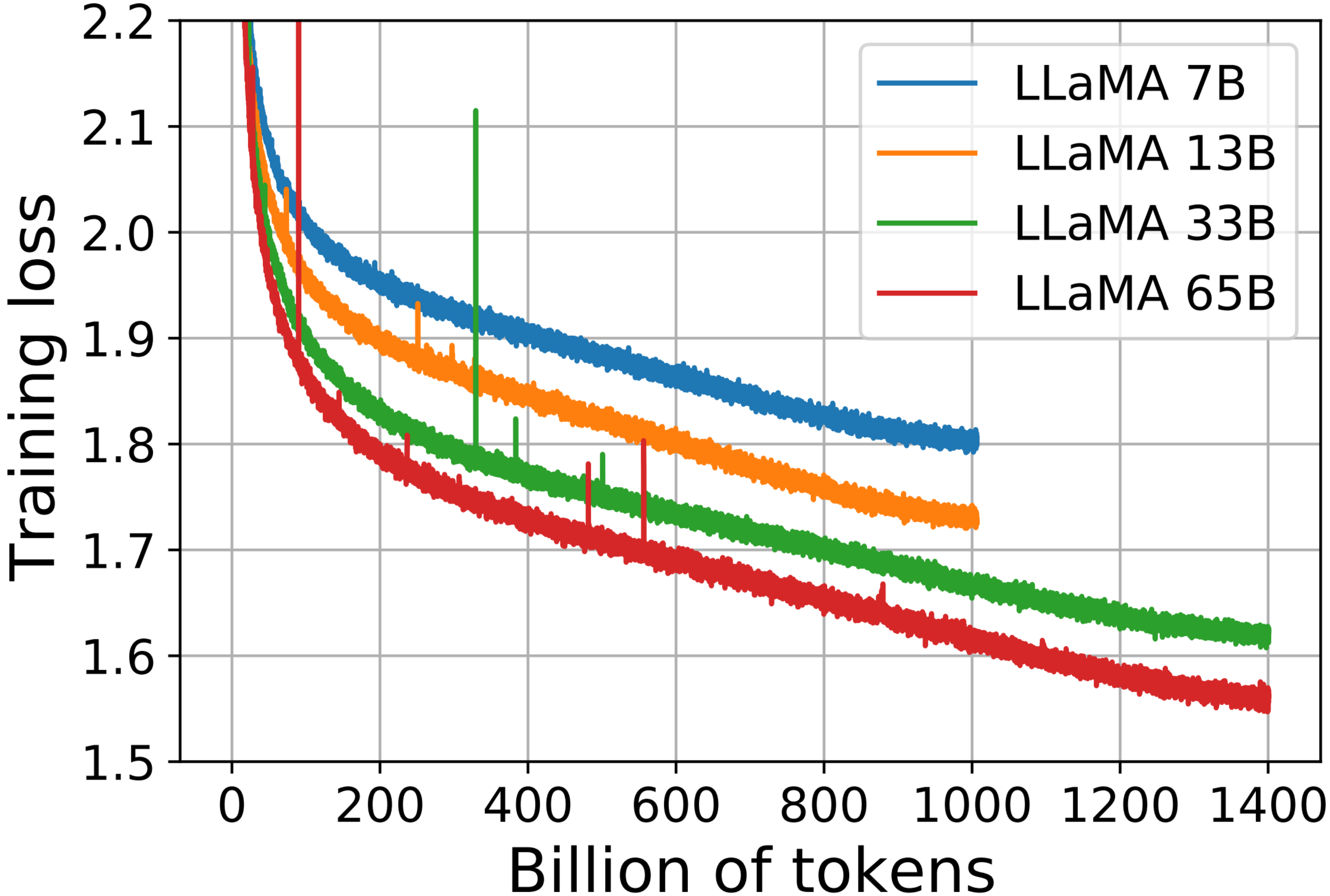
```
from transformers import AutoTokenizer, AutoModelForCausalLM

model = "HuggingFaceTB/SmolLM2-360M"

tokenizer = AutoTokenizer.from_pretrained(model)
model = AutoModelForCausalLM.from_pretrained(model)
```



https://github.com/cmu-l3/anlp-spring2025-code/blob/main/06_pretraining/pretraining.ipynb

Today's lecture

- Objectives
  - Masked language modeling
  - Autoregressive language modeling
- **Data**: sources, quality, and quantity



Better Loss

More data

Data factors

- **Quantity**: How much data do I have?
- **Quality**: Is it beneficial for training?
- **Coverage**: Does the data cover the domain(s) I care about, and in the right proportions?

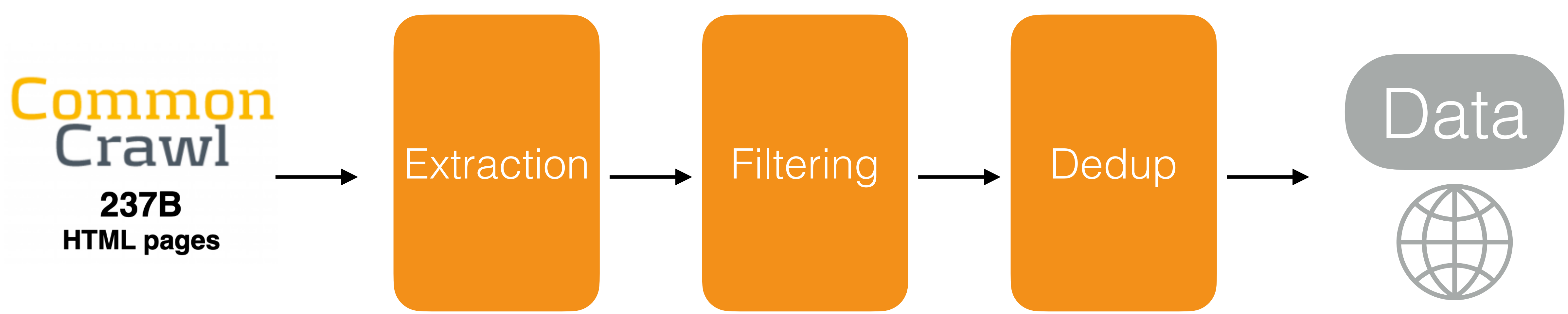
Data quantities

	Tokens of training data
Llama 1	1.4 trillion
Llama 2	1.8 trillion
Llama 3	15 trillion
Deepseek 3	15 trillion

Wikipedia: < 10 billion

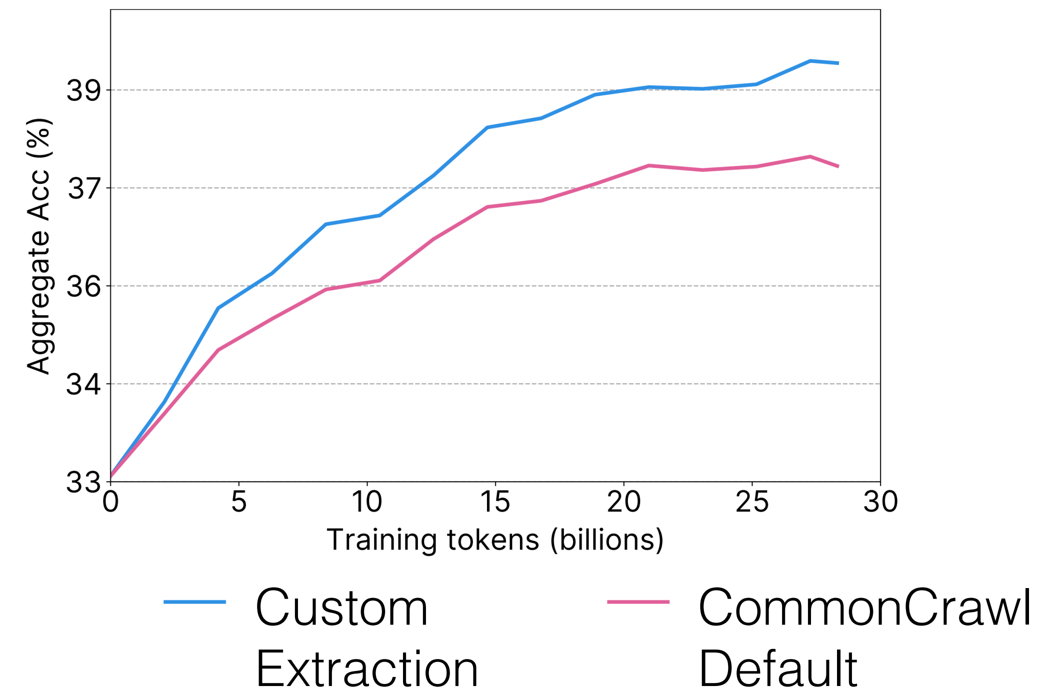
Web data: common crawl

- Large snapshots of web pages.
  - Extraction: HTML to text
  - Filtering: filter out unwanted pages
  - Deduplication: many duplicate web pages



Quality: Extraction

- Extraction: HTML to text
  - Remove boilerplate
  - Retain Latex, code, etc.



Penedo et al 2024

```
This paper concerns the quantity
<img src="https://s0.wp.com/
latex.php?latex=%7BM%28x%29..."
alt="{M(x)}" />, defined as the
length of the longest
subsequence of the numbers from
```

Image Equations

```
Suppose I have a smooth map
[tex]f\colon \mathbb{R}^3
\longrightarrow S^2[/tex]. If I
identify [tex]\mathbb{R}^3[/tex]
with [tex]U_S = S^3 - \
{(0,0,1)\}[/tex] via
stereographic projection
```

Delimited Math

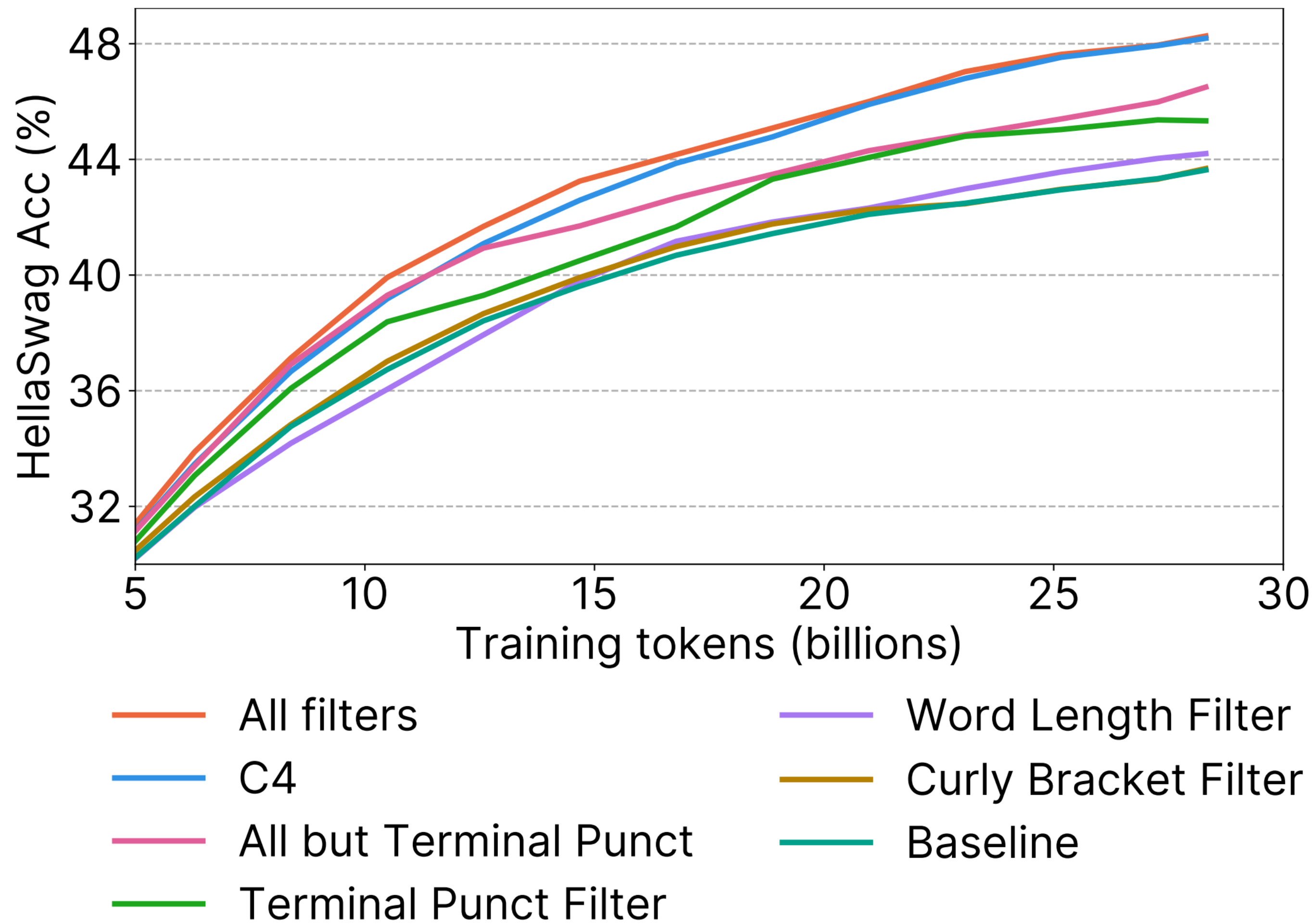
```
<math>
  <semantics>
    ...
    <annotation ...>
      {\displaystyle \mathrm {MA}
      ={\frac{f_{O}}{f_{E}}}}
    </annotation>
  </semantics>
</math>
```

Special Tags

Paster et al 2023

Quality: Filtering

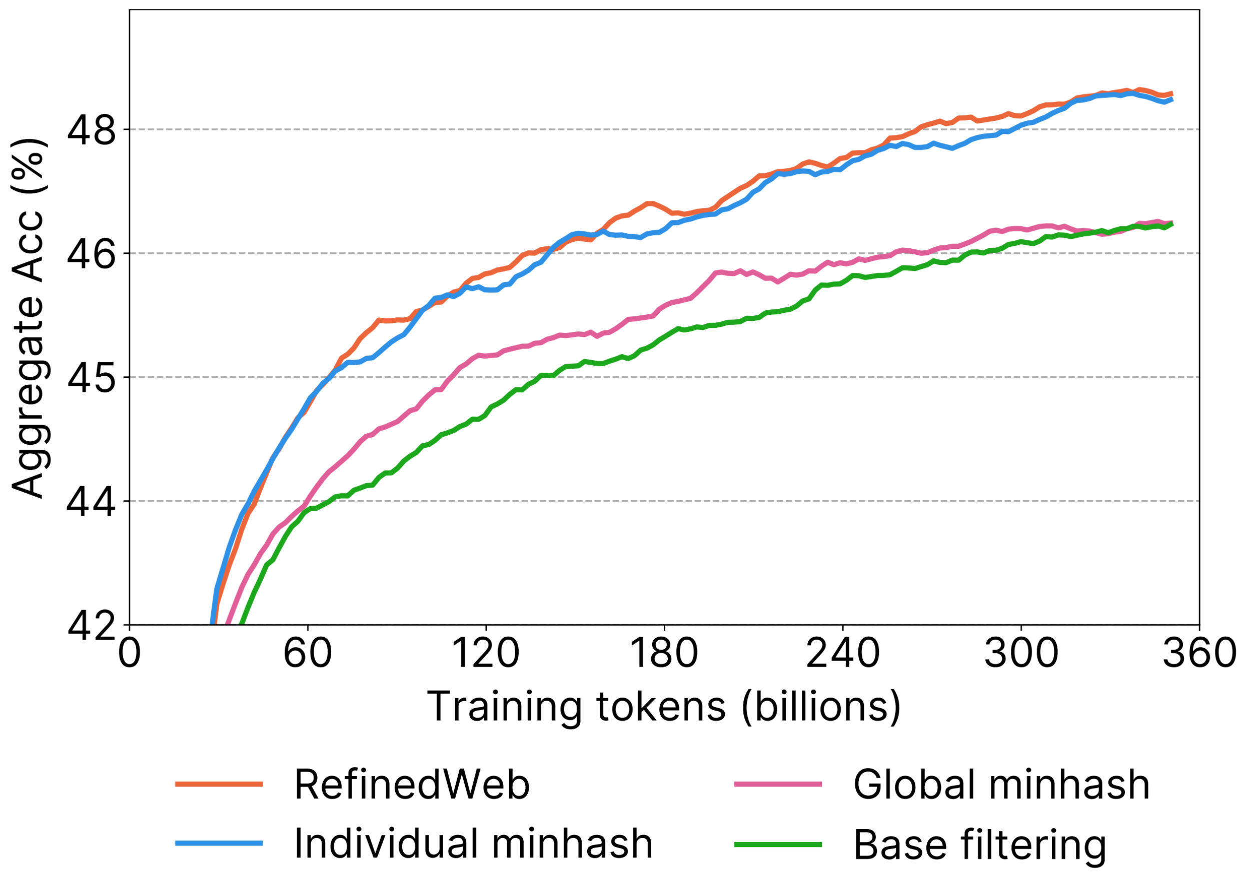
- Filter out unwanted text
  - Language filter
  - Repetitions
  - Too many short lines
  - …



Penedo et al 2024

Quality: Deduplication

- Remove duplicate content
  - Fuzzy strategy: *minhash*
  - Too much deduplication can be harmful
    - [Penedo et al 2024]: Deduplicate per-shard rather than globally



Penedo et al 2024

Example (Dolma)

```
added 2023-04-11T09:57:03.044571+00:00
attributes {'random_number_v1__random_number_v1__random': [[0, 9626, 0.11918]]}
created 2020-01-17T12:48:23Z
id http://250news.theexplorationplace.com/www.250news.com/65595.html
metadata {'bucket': 'head', 'cc_segment': 'crawl-data/CC-MAIN-2020-05/segments/1
source common-crawl
text Prince George, B.C.- Construction of the new RiverBend Seniors housing proj
The $33 million dollar project was first presented to Mayor and Council in 2013
Hall and key members of the City Staff, arranged to meet with Quinn in Kamloops
“This project comes at the perfect time for us” says Gwen Norheim. She and her h
Quinn says they did make an interesting discovery when they started constructior
That’s it, big smiles Shirley and Mike… there is an election coming.
This is an excellent and well needed project!
If the NDP was in power (god forbid) and it was NDP MLA’s in the picture, you wo
Go ahead and deny it if you want, but we know better!
What we do know with the liberals is they are always raising fees and medical co
grow up galt.
```

https://github.com/cmu-l3/anlp-spring2025-code/blob/main/06_pretraining/pretraining.ipynb

Coverage

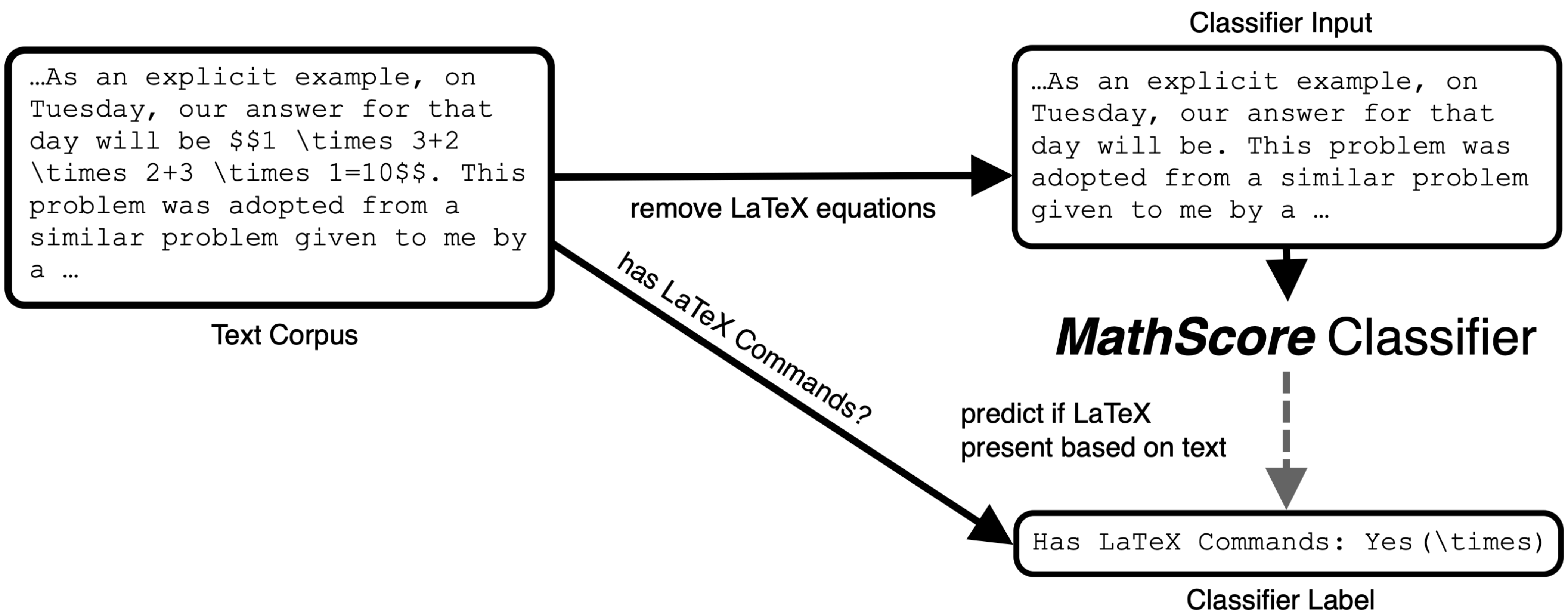
- The data determines the data distribution
  - And hence the model, $p_\theta \approx p_{data}$
- Web data $\neq$ educational data $\neq$ math data

Approach: classifier filtering

- Train a classifier to detect desired data
- Use it to filter out undesired data

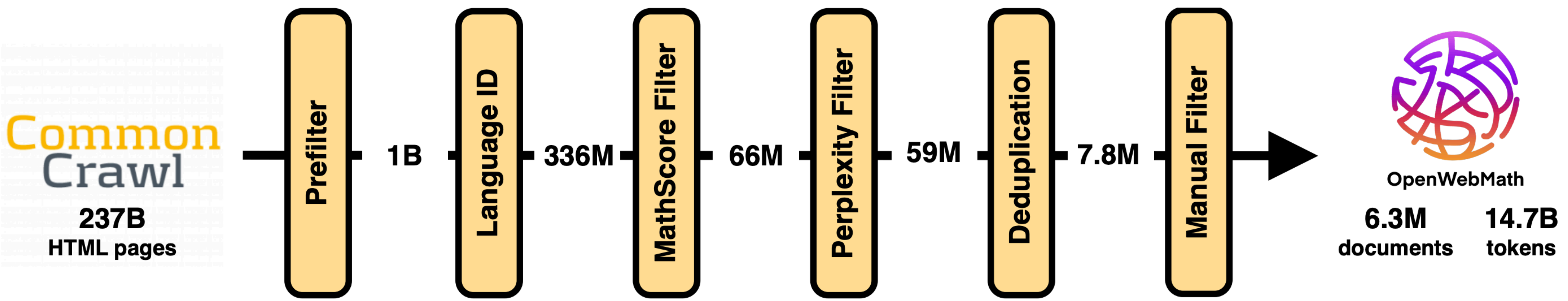
Approach: classifier filtering

- Example: **OpenWebMath** [Paster et al 2023]
    - MathScore classifier detects math content



Approach: classifier filtering

- Example: **OpenWebMath** [Paster et al 2023]
    - MathScore classifier detects math content



https://huggingface.co/datasets/open-web-math/open-web-math

Approach: classifier filtering

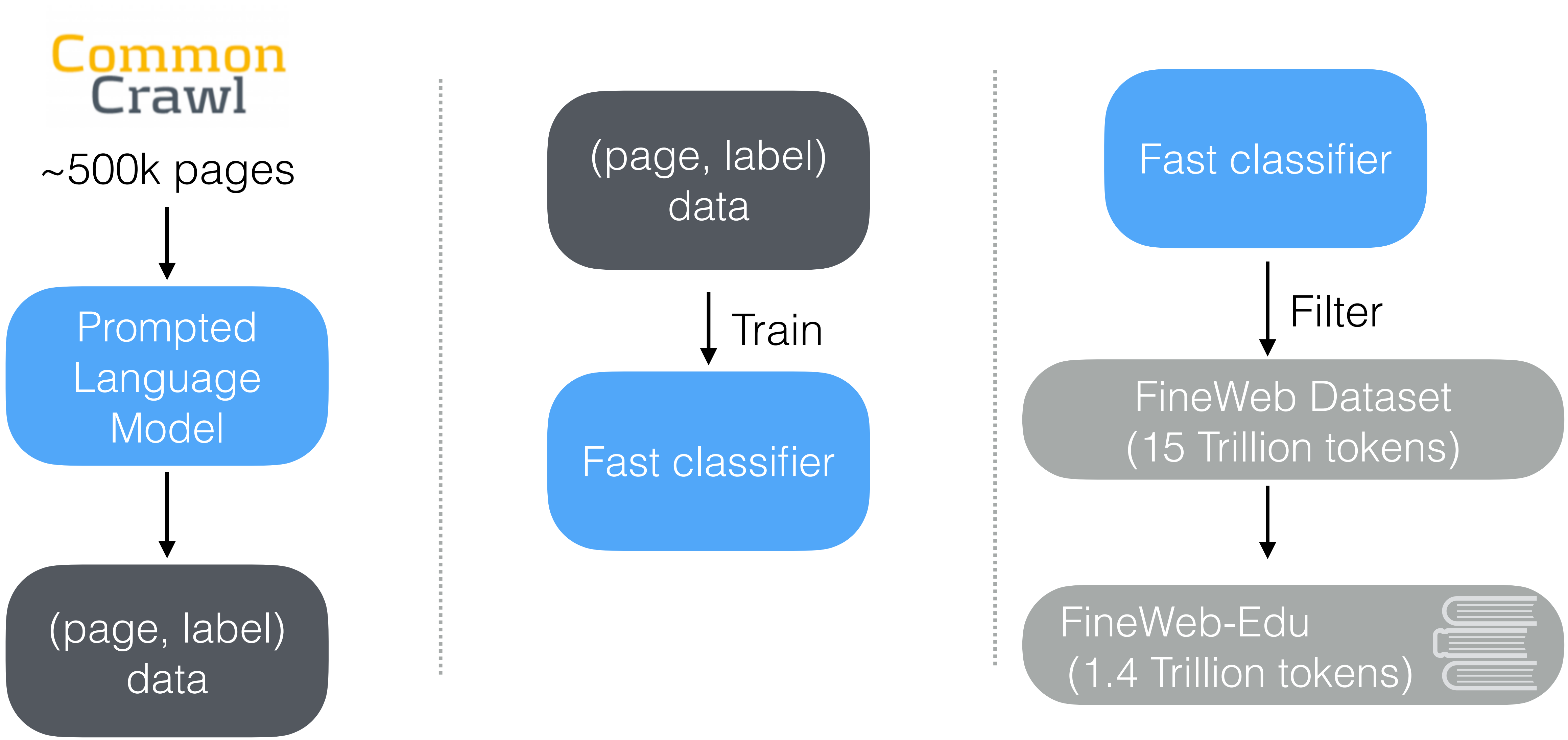
- Example: **OpenWebMath** [Paster et al 2023]

Domain	# Characters	% Characters
stackexchange.com	4,655,132,784	9.55%
nature.com	1,529,935,838	3.14%
wordpress.com	1,294,166,938	2.66%
physicsforums.com	1,160,137,919	2.38%
github.io	725,689,722	1.49%
zbmath.org	620,019,503	1.27%
wikipedia.org	618,024,754	1.27%
groundai.com	545,214,990	1.12%
blogspot.com	520,392,333	1.07%
mathoverflow.net	499,102,560	1.02%

https://huggingface.co/datasets/open-web-math/open-web-math

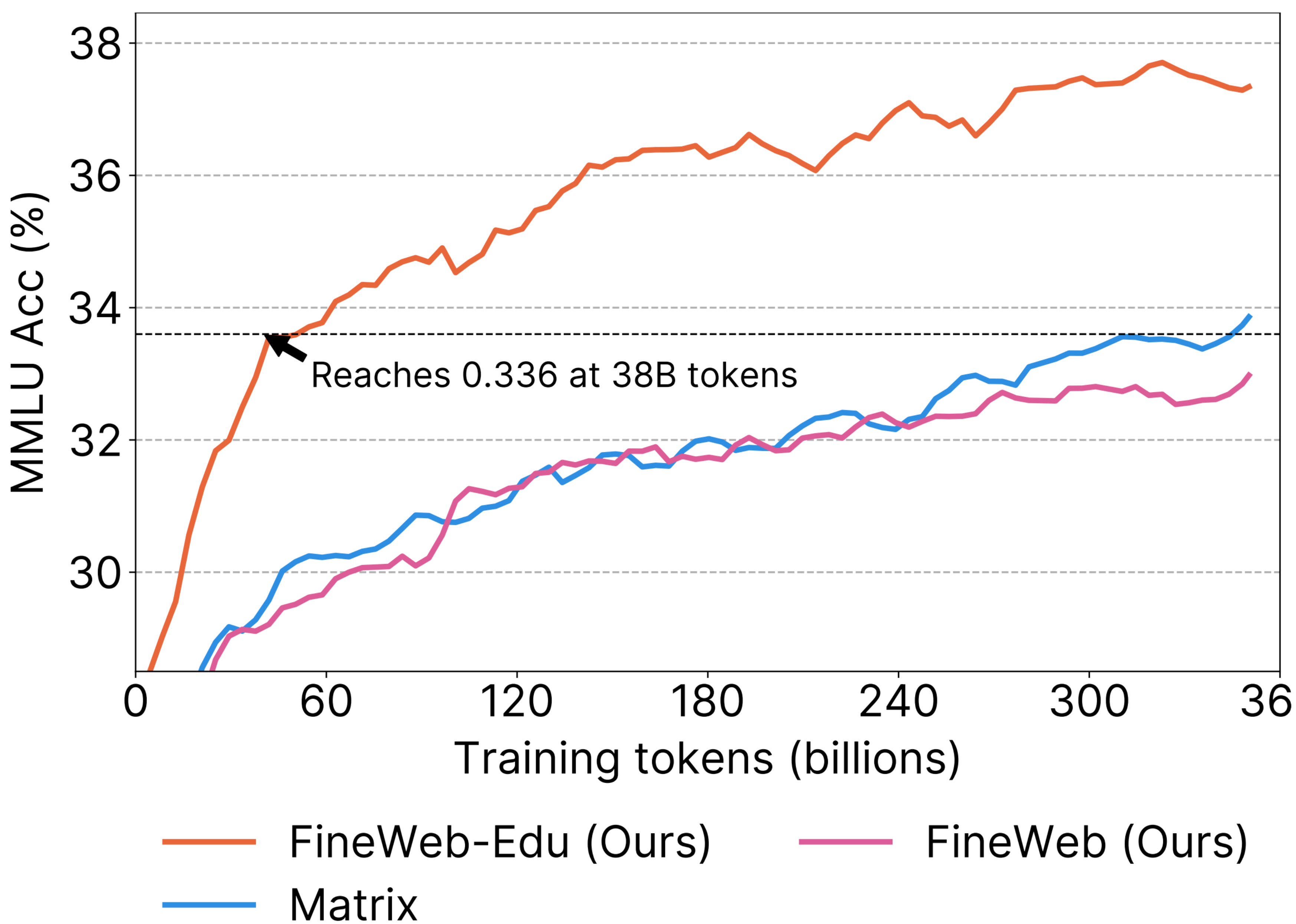
Approach: classifier filtering

- Example: **FineWeb-Edu** [Penedo et al 2024]
  - Classifier to classify pages as "educational"



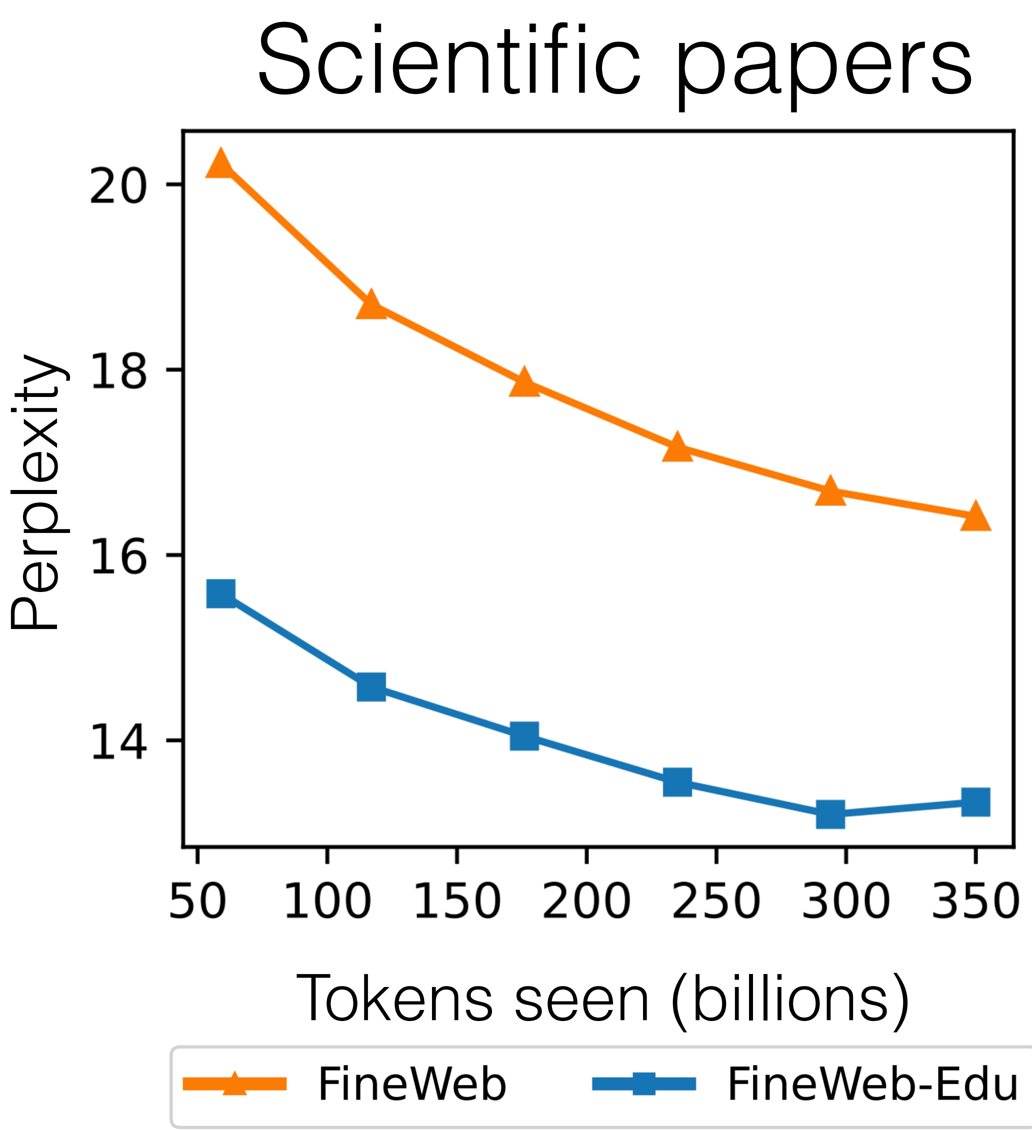
Approach: classifier filtering

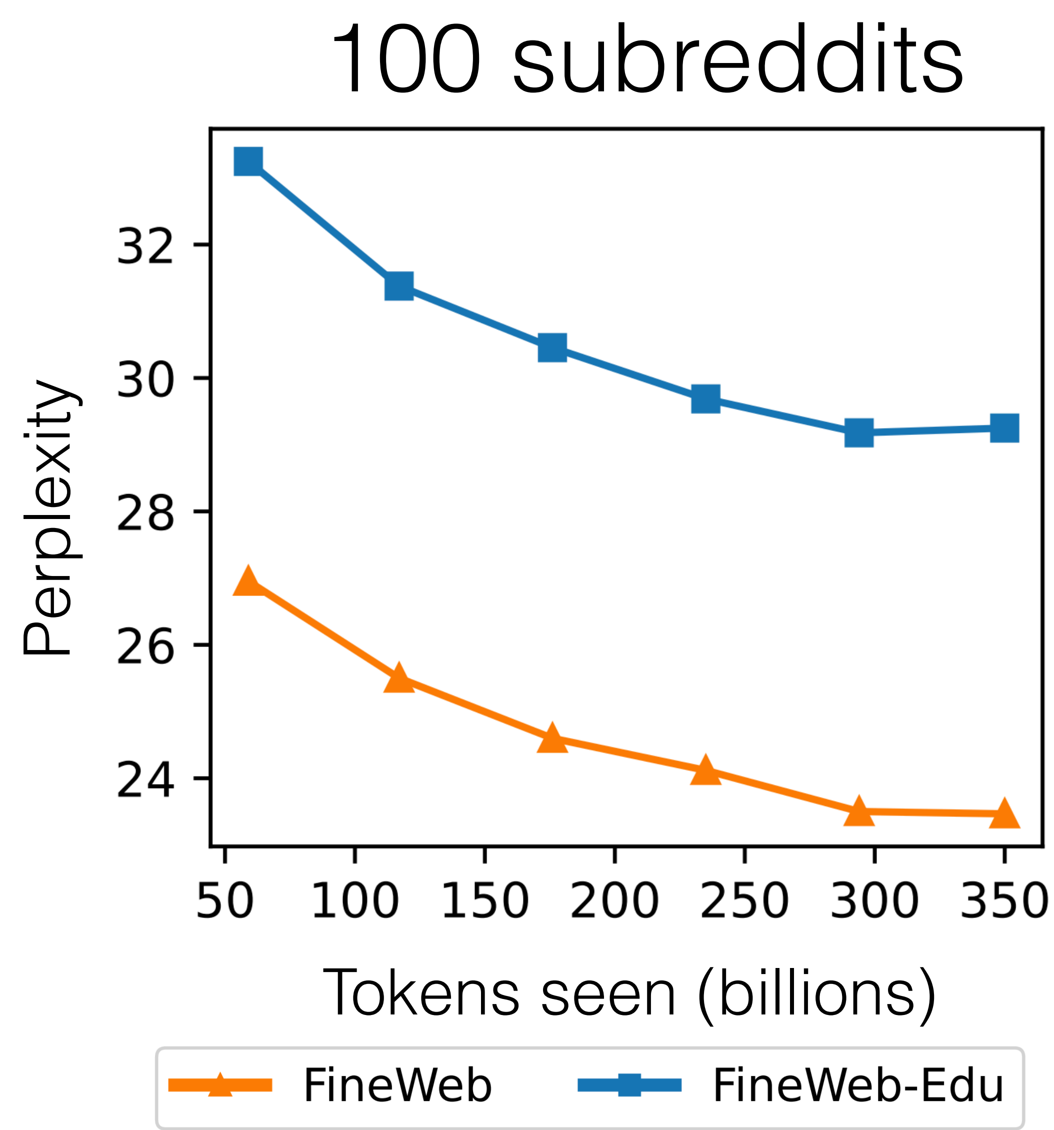
- Example: **FineWeb-Edu** [Penedo et al 2024]



Approach: classifier filtering

- Example: **FineWeb-Edu** [Penedo et al 2024]





Mixtures

- In practice, training data is a mixture of different sources

Source	Type	Tokens
	Pretraining ✦ OLMo 2 1124 Mix	
DCLM-Baseline	Web pages	3.71T
StarCoder filtered version from OLMoE Mix	Code	83.0B
peS2o from Dolma 1.7	Academic papers	58.6B
arXiv	STEM papers	20.8B
OpenWebMath	Math web pages	12.2B
Algebraic Stack	Math proofs code	11.8B
Wikipedia & Wikibooks from Dolma 1.7	Encyclopedic	3.7B
Total		**3.90T**

Recap

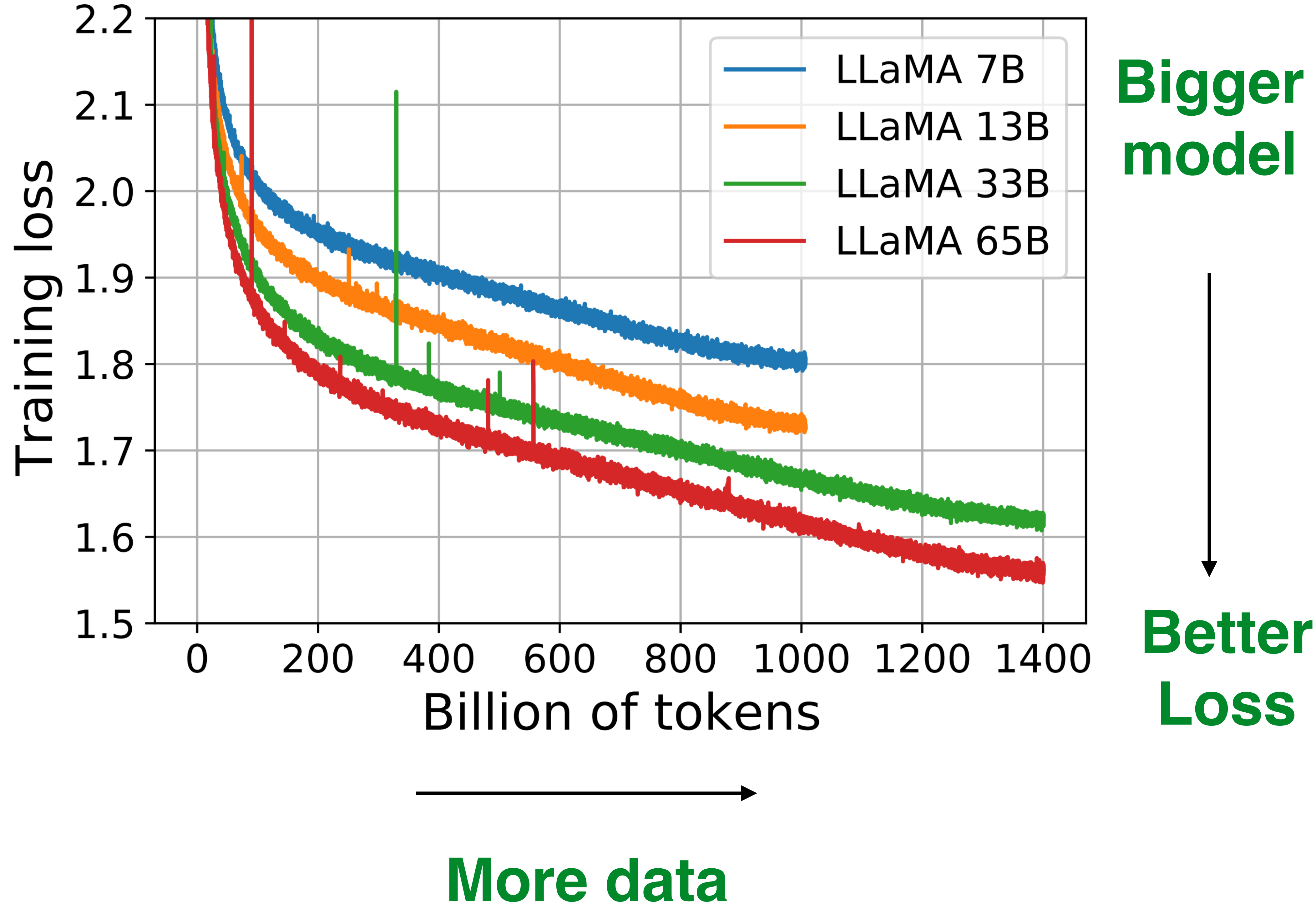
- Web data: large quantities of data
  - Extract, filter, deduplicate to improve quality
  - Filter to cover desired domain(s)
- Mix together web data and other sources to make a pre-training dataset

Recent examples

	Year	Domain	Tokens
FineWeb	2024	Web	15 trillion
RedPajama v2	2024	Web	30 trillion
Dolma	2024	Mix	3 trillion
OLMO2 Mix	2025	Mix	4 trillion
OpenWebMath	2023	Math web pages	15 billion
AlgebraicStack	2023	Math code	11 billion
FineWeb-Edu	2024	Educational (middle-school)	1.4 trillion

Today's lecture

- Objectives
- Data
- **Thinking about pretraining**
  - Tokens, model size, compute
  - Scaling laws



2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
Training loss
0
200
400
600
800
1000
1200
1400
Billion of tokens
LLaMA 7B
LLaMA 13B
LLaMA 33B
LLaMA 65B
Bigger model
Better Loss
More data

Pretraining and compute

- Goal: get a better pretrained model by “adding more compute”

    - *“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.”*

      \- *The Bitter Lesson*, Richard Sutton 2019

What is compute?

- We spend **compute** by performing forward and backward passes on training sequences

- An approximation for transformer language models:

$$C \approx 6ND$$

N: number of model parameters
D: number of tokens
C: compute; floating point operations (FLOPs)

What is compute?

- We spend **compute** by performing forward and backward passes on training sequences

- For example, Llama 2:

$$C \approx 6 \times 7 \text{ billion} \times 2 \text{ trillion}$$

$$= 8.4 \times 10^{22} \text{FLOPs}$$

N: number of model parameters
D: number of tokens
C: compute; floating point operations (FLOPs)

What is compute?

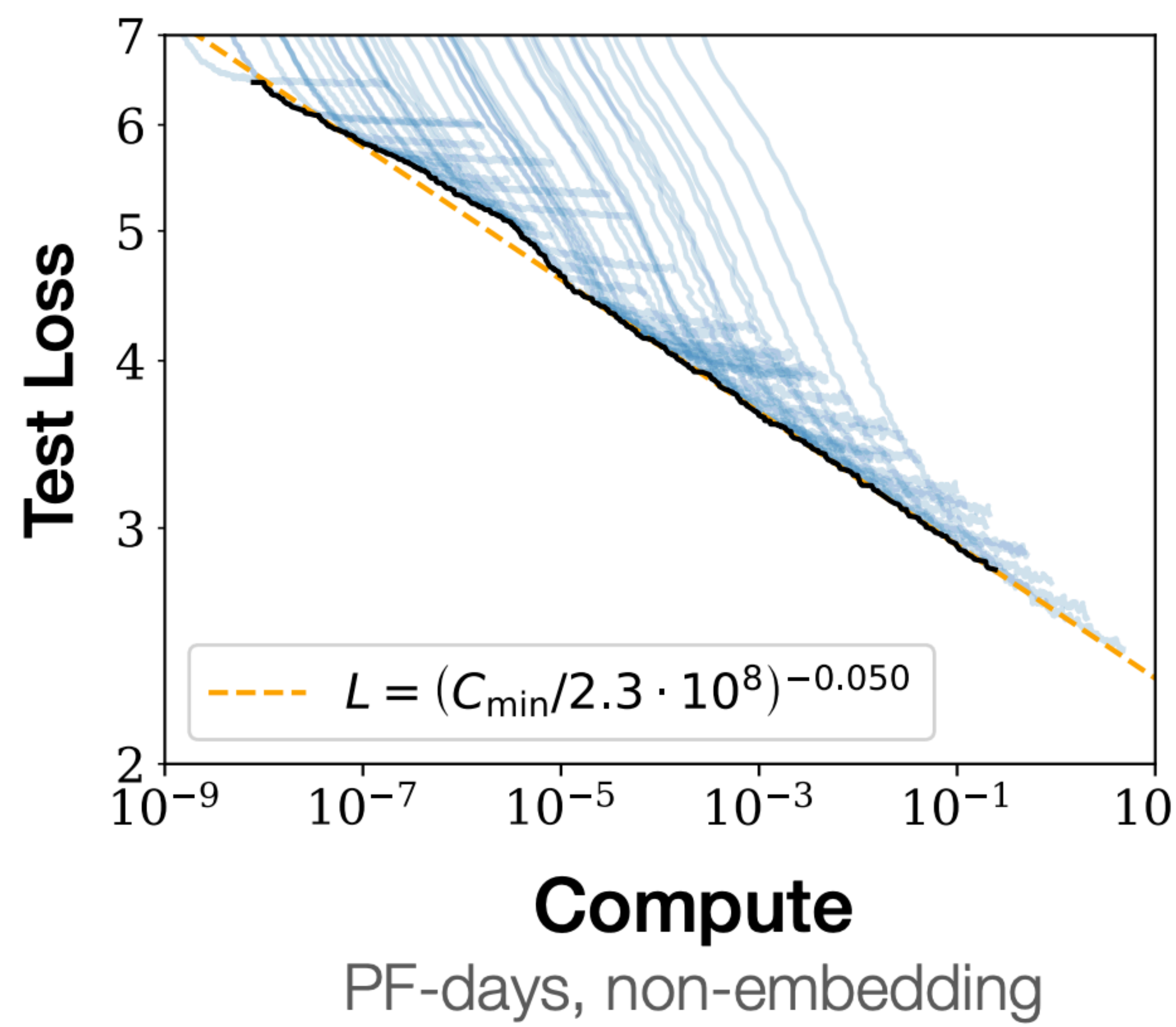
- We spend **compute** by performing forward and backward passes on training sequences

- We can **increase compute** by increasing the **number of parameters** ($\uparrow N$), training on **more tokens** **(** $\uparrow D$**)**, or a combination of both

N: number of model parameters
D: number of tokens
C: compute; floating point operations (FLOPs)

Scaling laws

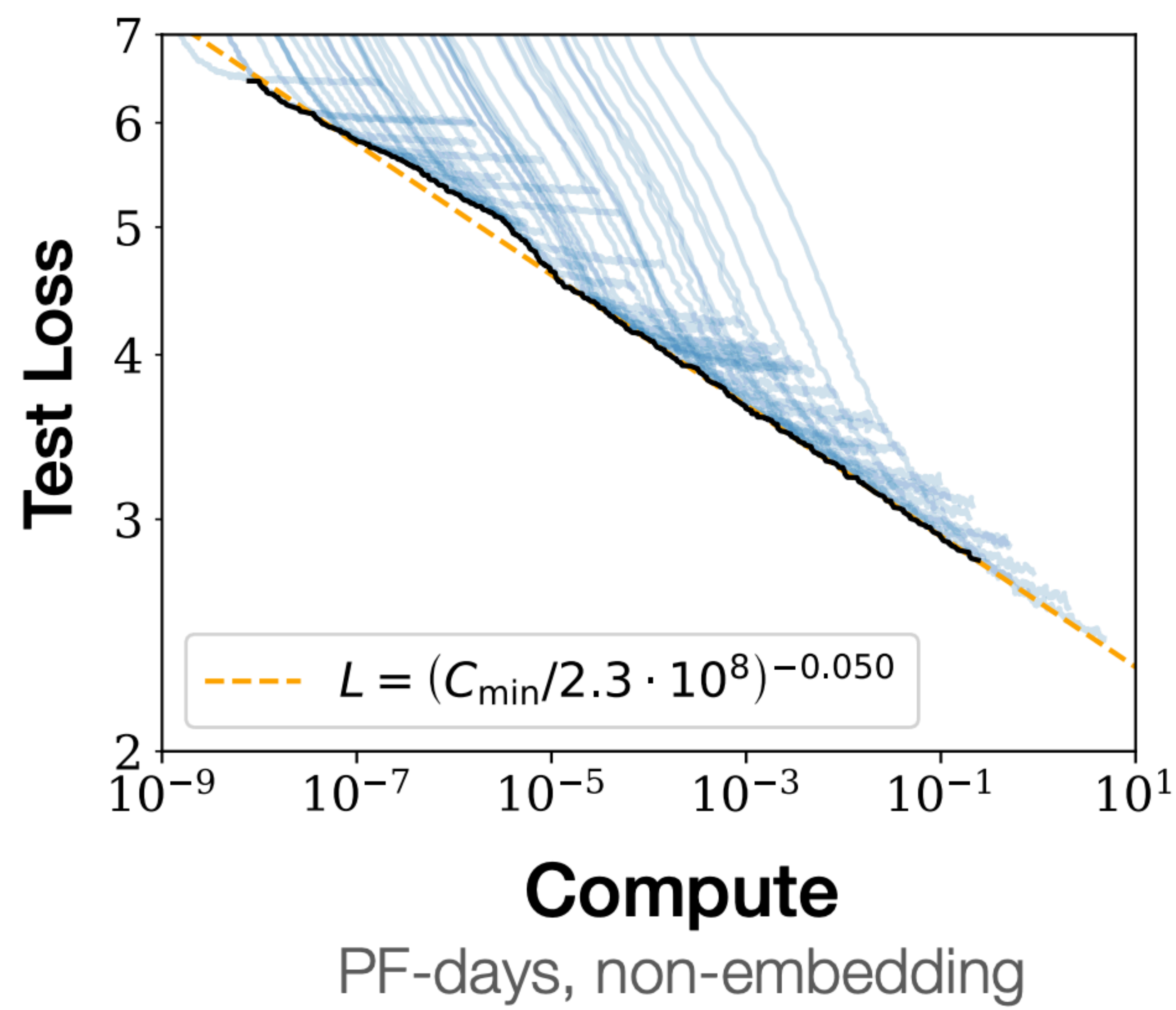
- **Key finding**: language modeling loss predictably improves with more compute



[Scaling Laws for Neural Language Models]()

Scaling laws

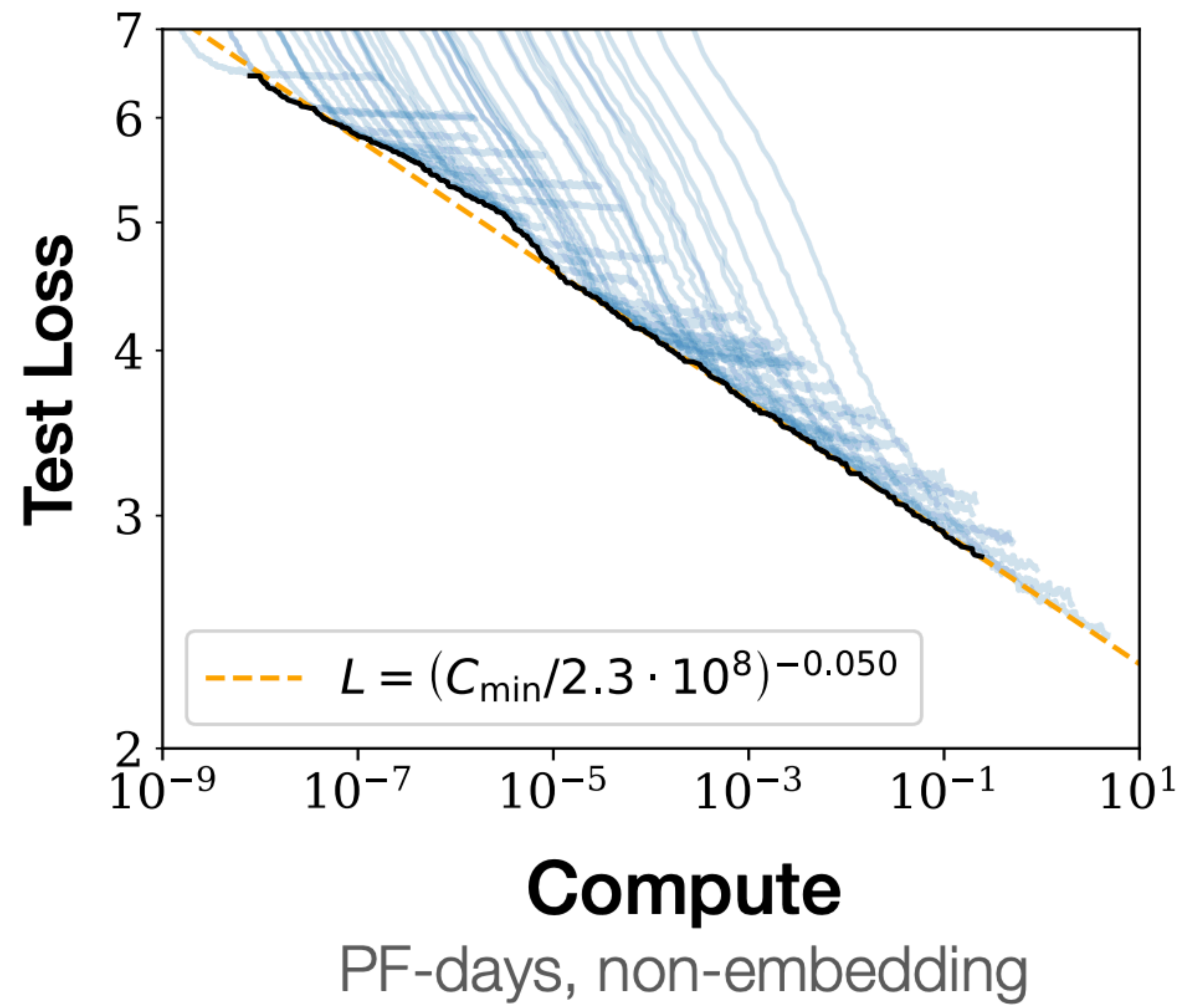
- **Basic idea**:
  - Train models of different sizes and numbers of tokens
  - Plot loss at each step of training [light blue]
  - Pick minimum loss at each amount of compute [black]
  - Run linear regression on the resulting (loss, compute) pairs [orange]



Scaling laws

Terminology:

- **Compute optimal**: black
- **Scaling law**: orange
    - E.g. $L(C) \propto 1/C^{0.05}$



Recap

- We can think of pre-training in terms of *compute*, which is determined by *model size* and *number of tokens*

- Key finding: *increasing compute* leads to a *better model*

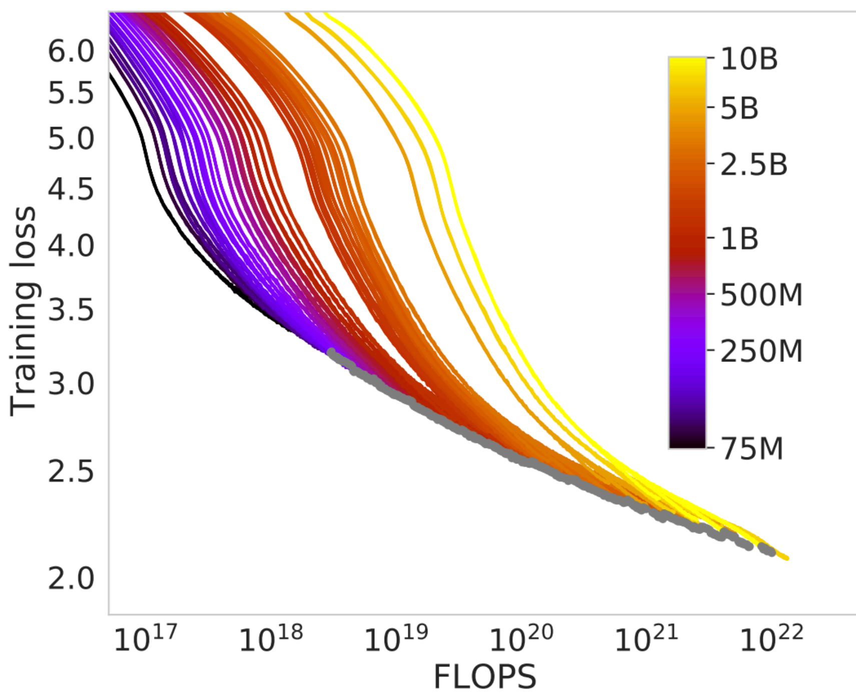
- *Scaling laws* give a formula for the relationship between compute and loss

    - (Or related quantities, such as model size and loss)

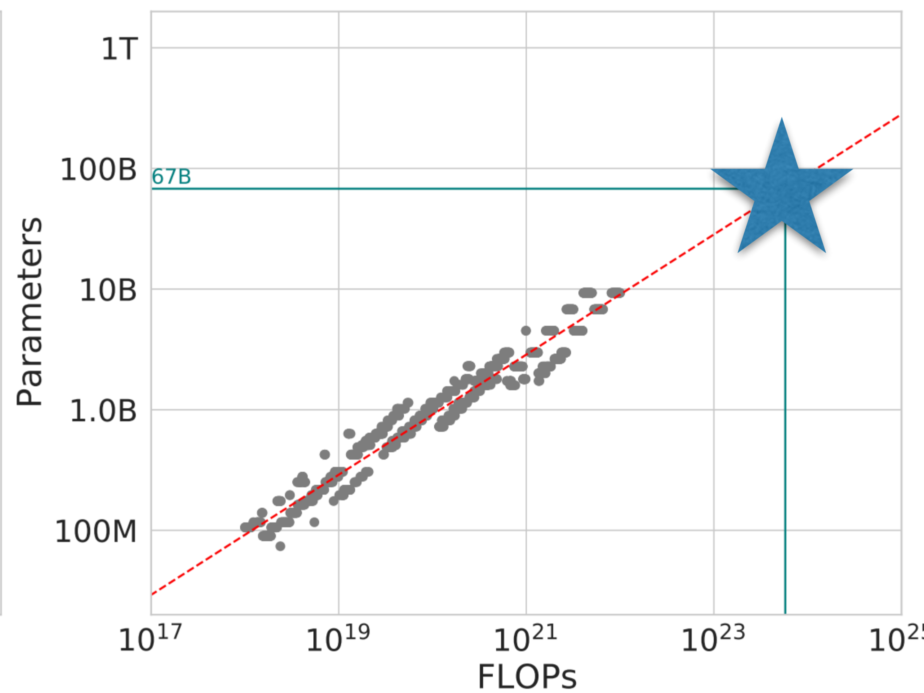
Using scaling laws

- Scaling laws are also used to choose hyper parameters
- Basic idea:
  - Run many experiments at a small scale
  - Use a scaling law to estimate the best hyper parameter for a large-scale model / training run

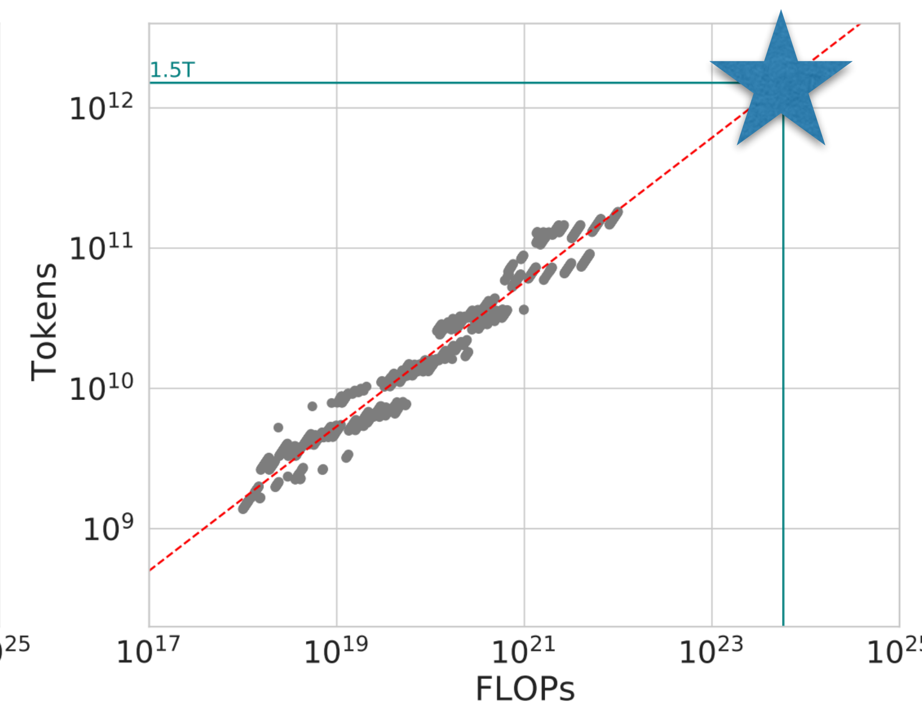
Example: choose model size and # of tokens



Run experiments



Fit a line and predict optimal model size

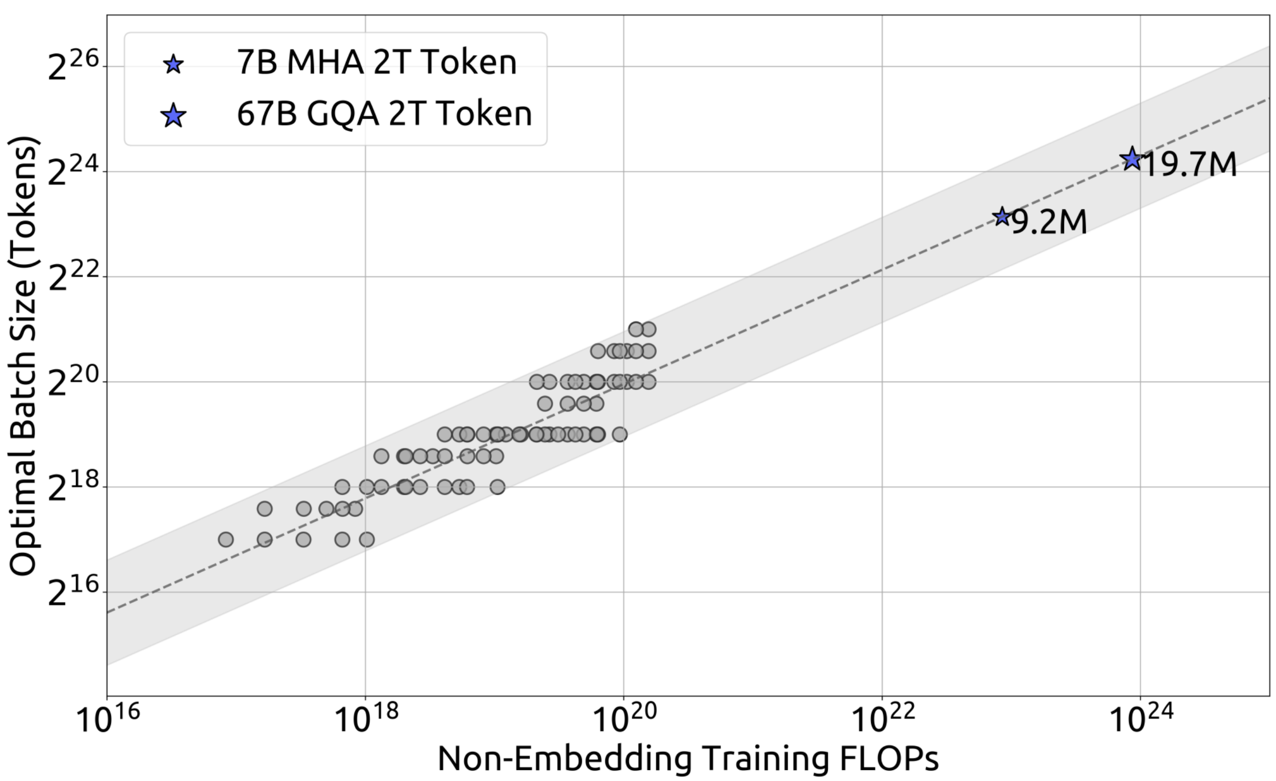


Fit a line and predict optimal # of tokens

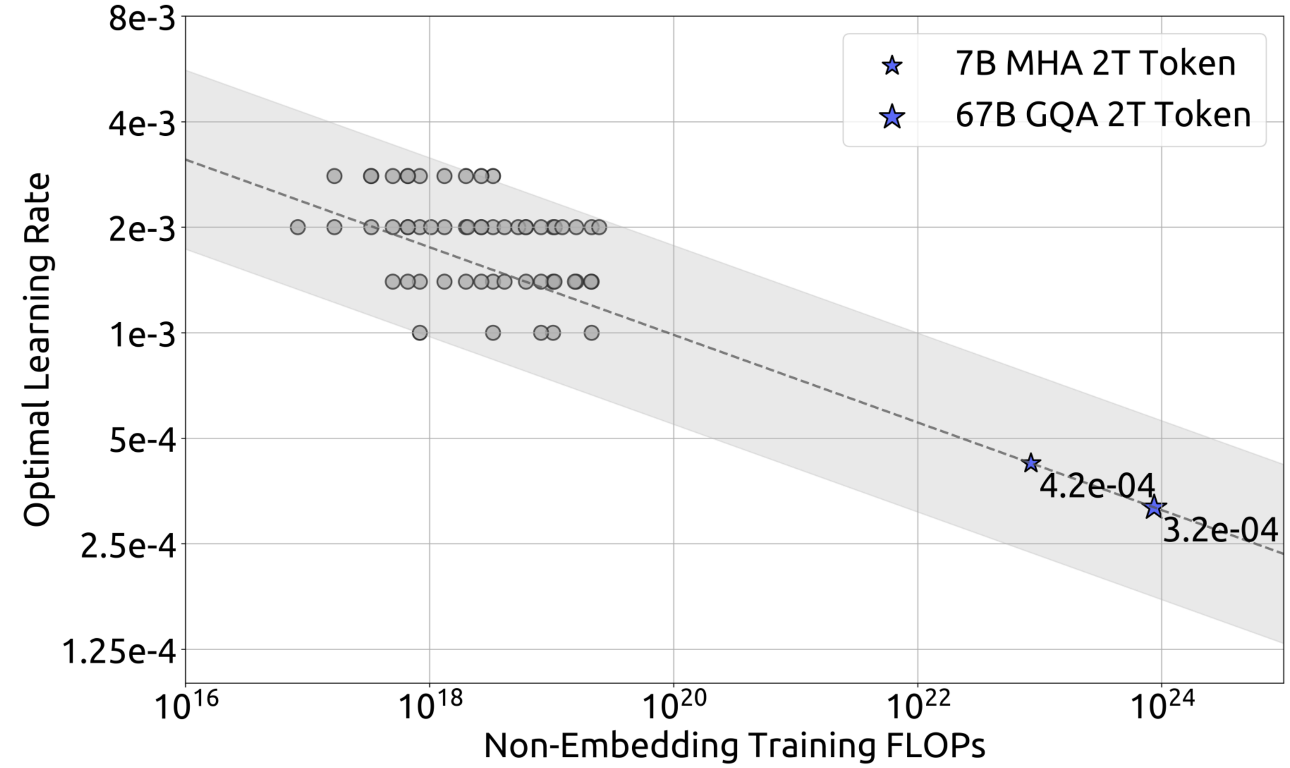
“Optimal”: best loss for a given compute budget (FLOPs)

Training Compute-Optimal Large Language Models

Example: choose batch size, learning rate



Optimal batch size



Optimal learning rate

DeepSeek LLM: Scaling Open-Source Language Models with Longtermism

Today's lecture

- Pretraining objectives
  - Masked language modeling
  - Autoregressive language modeling
- Pretraining data: sources, quality, and quantity
- Thinking about pretraining
  - Tokens, model size, compute
  - Scaling laws

Roadmap

2nd half of the course: ***advanced pretraining***

Next lectures:

- Lecture 7: generating with a model
- Lecture 8: prompting
- Lecture 9: fine-tuning

Questions?