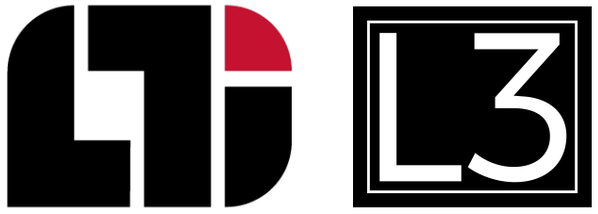
CS11-711 Advanced NLP

Pretraining

Sean Welleck



https://cmu-l3.github.io/anlp-spring2026/

https://github.com/cmu-l3/anlp-spring2026-code

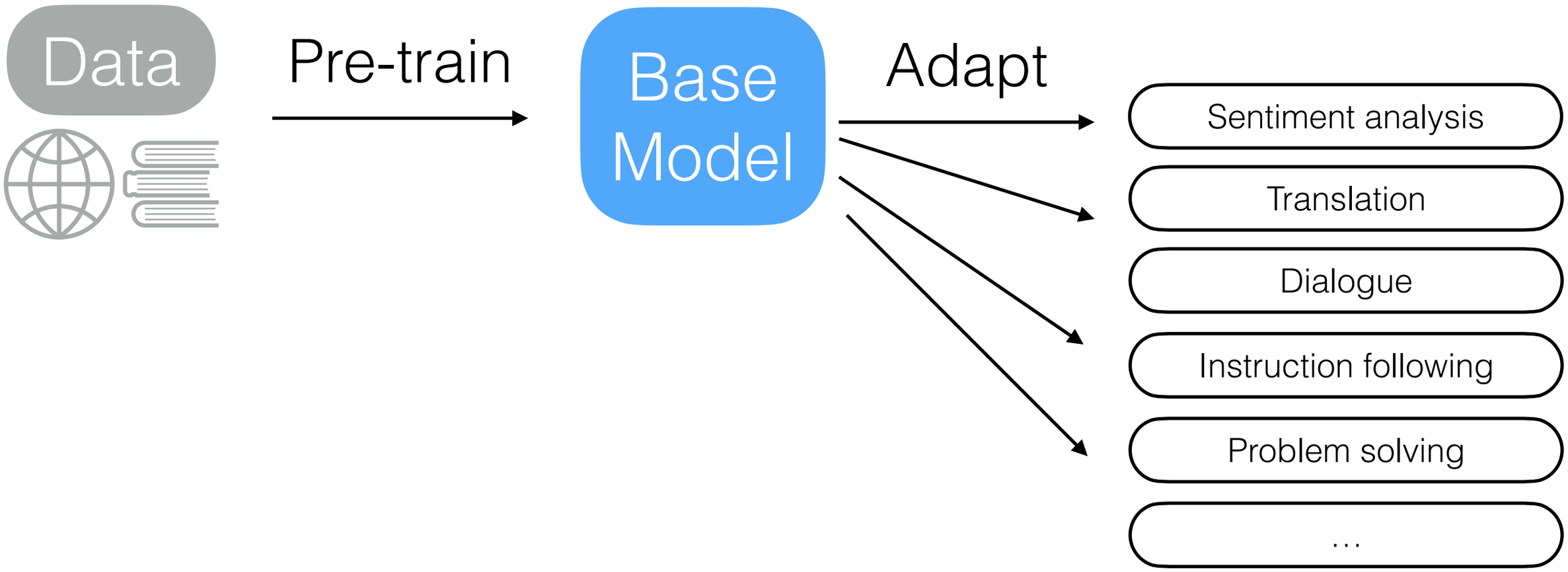
Some slides adapted from Graham Neubig's Fall 2024 course

Recap

- Classification, language modeling, sequence architectures

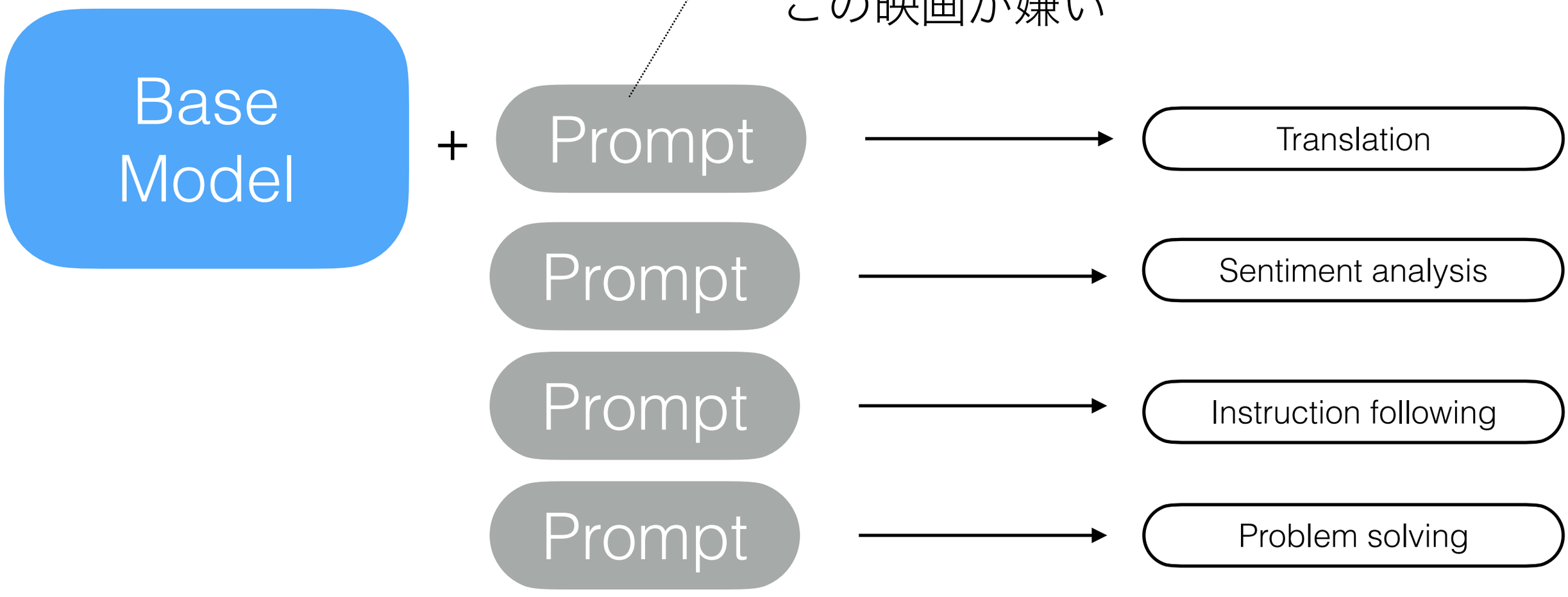
- So far:
    - Train from scratch
    - 1 model, 1 task
- Today:
    - *Pretrain* a *single model*, adapt it to *many tasks*

Basic idea

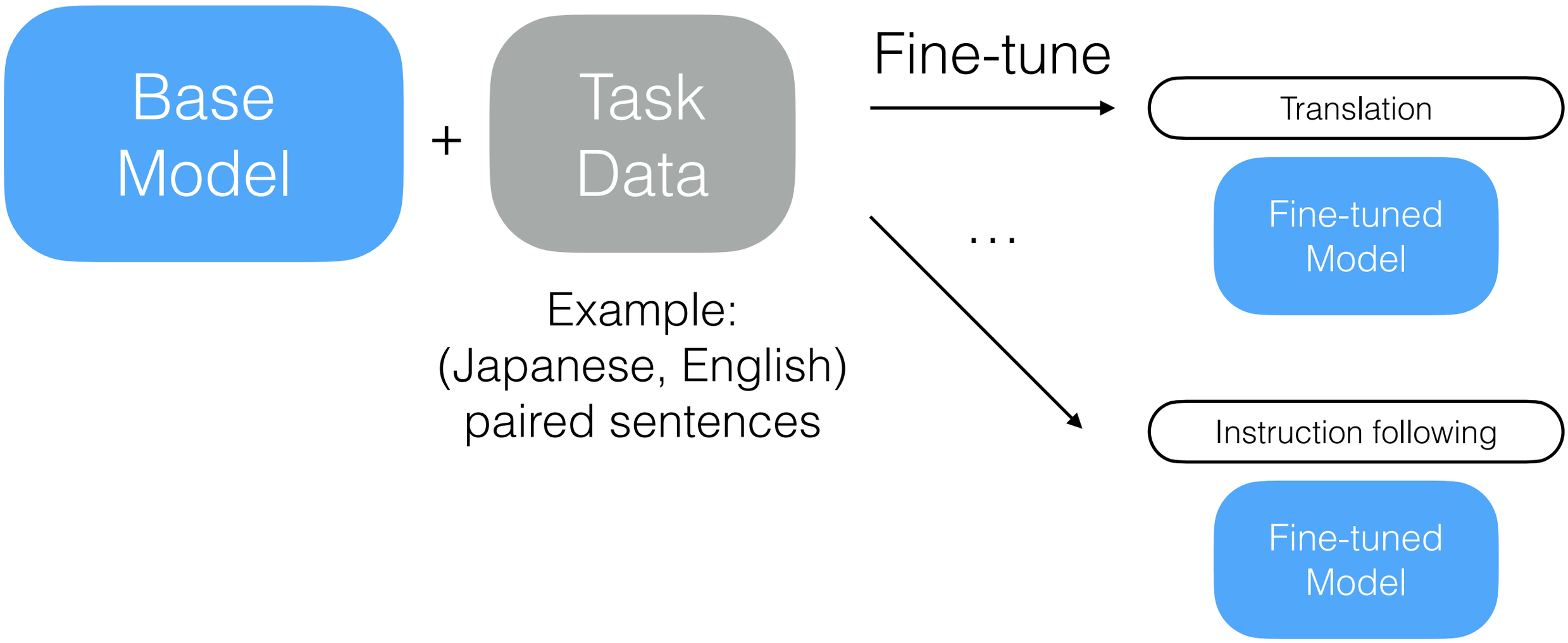

Data
Pre-train
Base Model
Adapt
Sentiment analysis
Translation
Dialogue
Instruction following
Problem solving
…

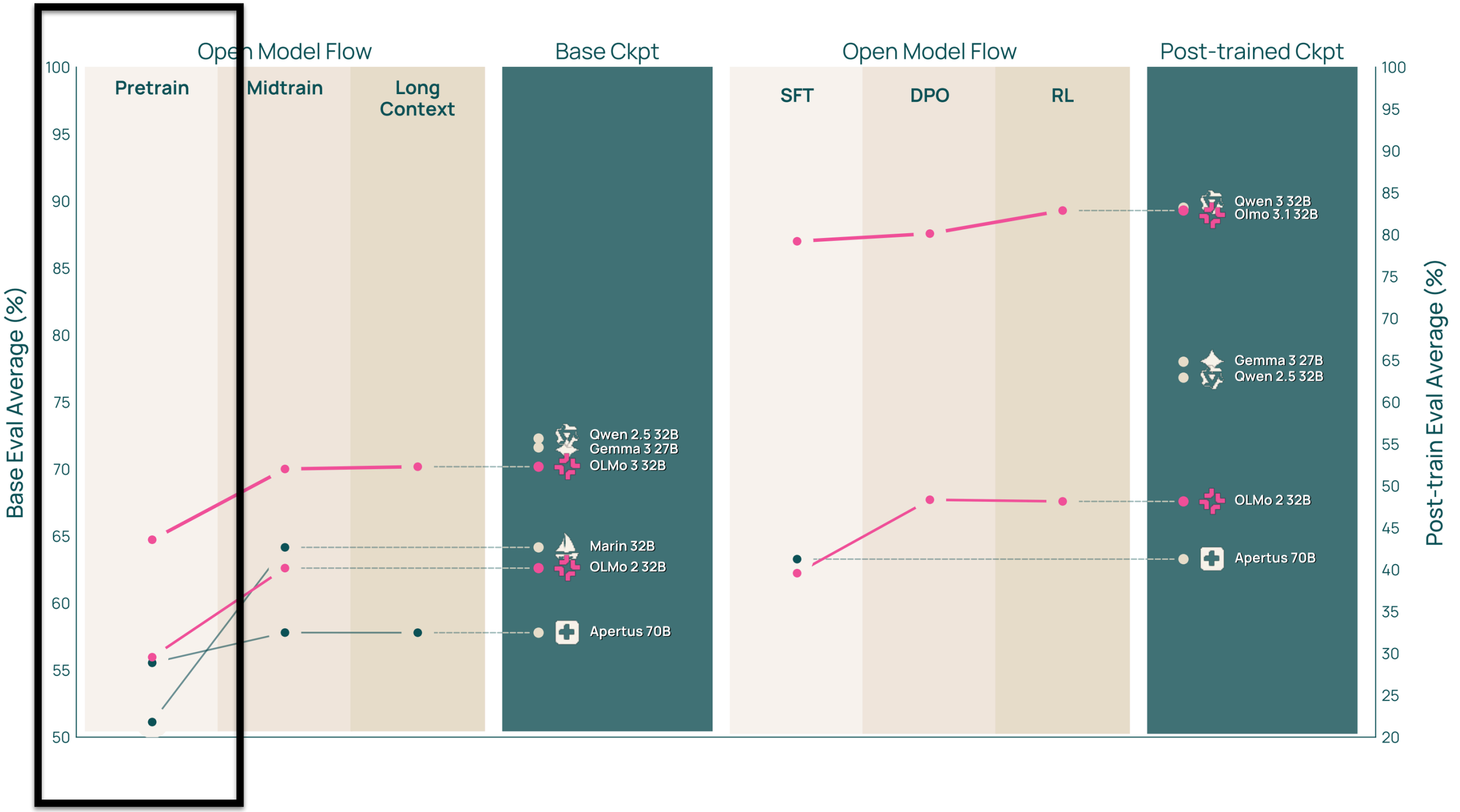
Adaptation: prompting [Lecture 7]

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



Adaptation: fine-tune [Lecture 8]





OLMO 3 [AI2 Dec 2025]

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

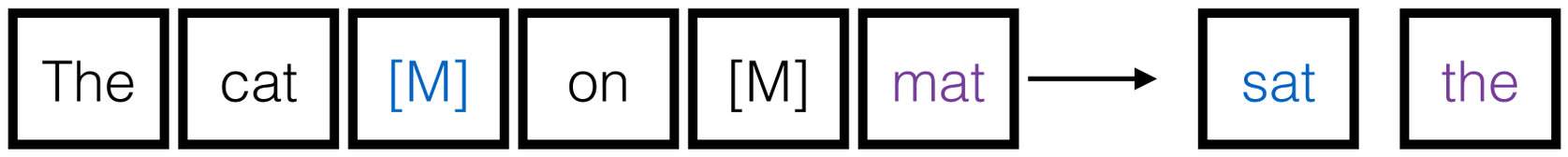
- Pre-trained models have names like BERT, GPT-3, Llama, Deepseek-v3, …
- Each model is influenced by 4 major factors:
  - **Architecture:** neural network architecture
  - **Task:** what the model predicts (e.g. next-token)
  - **Data:** the data used to train the model
  - **Hyper-parameters**: e.g. learning rate, batch size

Today's lecture

- Tasks
  - Masked language modeling objective
  - Autoregressive language modeling objective
- Data: sources, quality, and quantity
- Thinking about pretraining [next lecture]
  - Tokens, model size, compute
  - Scaling laws
- [Extra topic]: Adam optimizer

Masked Language Modeling

- Predict masked tokens x_M given visible tokens $x_{\neg M}$



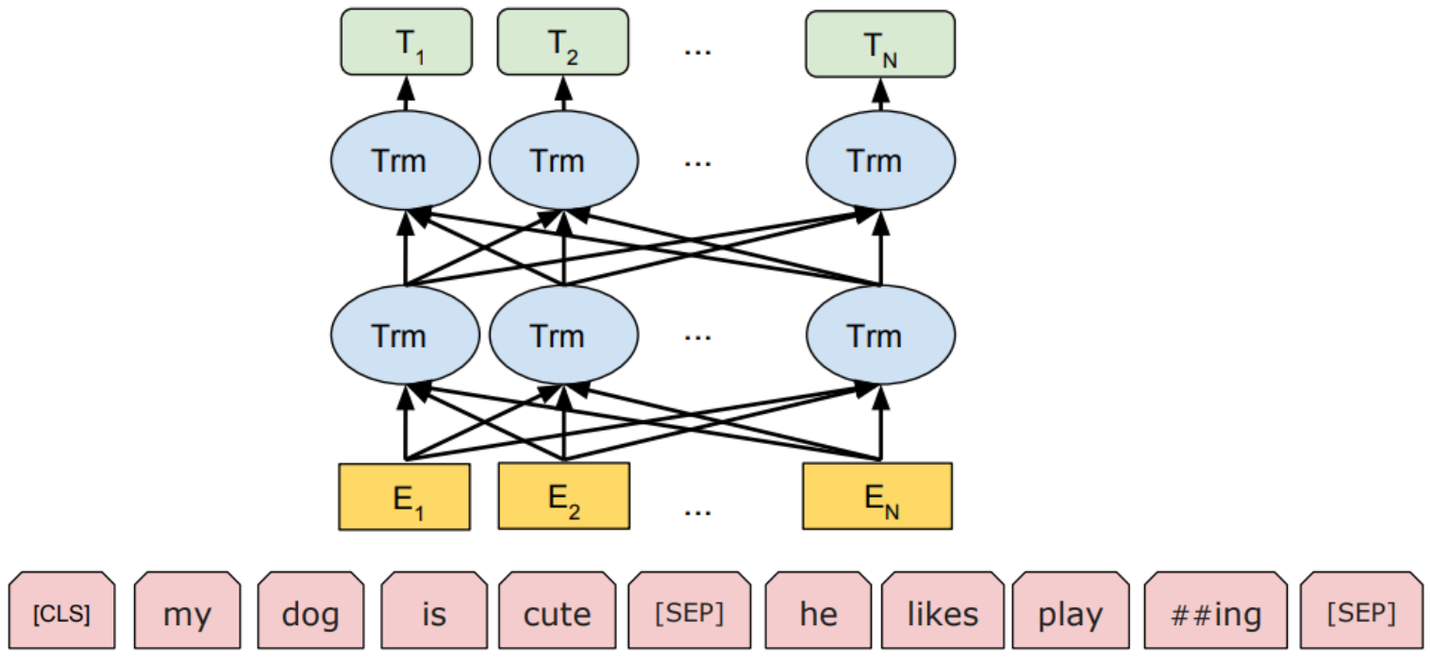
$$\mathscr{L}_{\text{MLM}}(\theta; D) = -\frac{1}{|D|} \sum_{x \in D} \mathbb{E}_{M \sim \text{corrupt}(x)} \sum_{t \in M} \log p_\theta(x_t \mid x_{\neg M})$$

- View as *denoising*: corrupt $x \rightarrow$ reconstruct x
- Maximizes *pseudo-likelihood*

Example: BERT

(Devlin et al. 2018)

- **Architecture:** Transformer



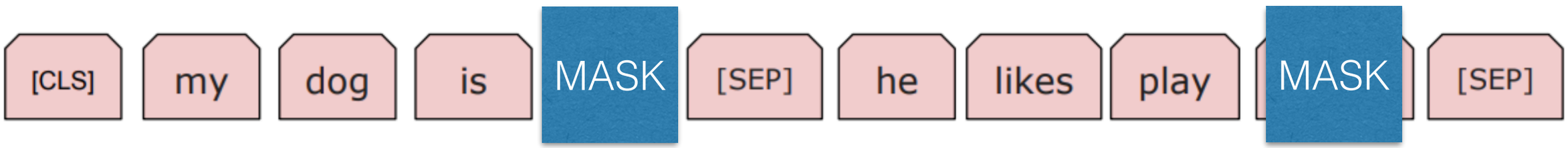
- **Data:** BooksCorpus + English Wikipedia
- **Task:** Masked language modeling

Example: BERT

(Devlin et al. 2018)

Choose 15% of token positions at random

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



Adapting a masked language model

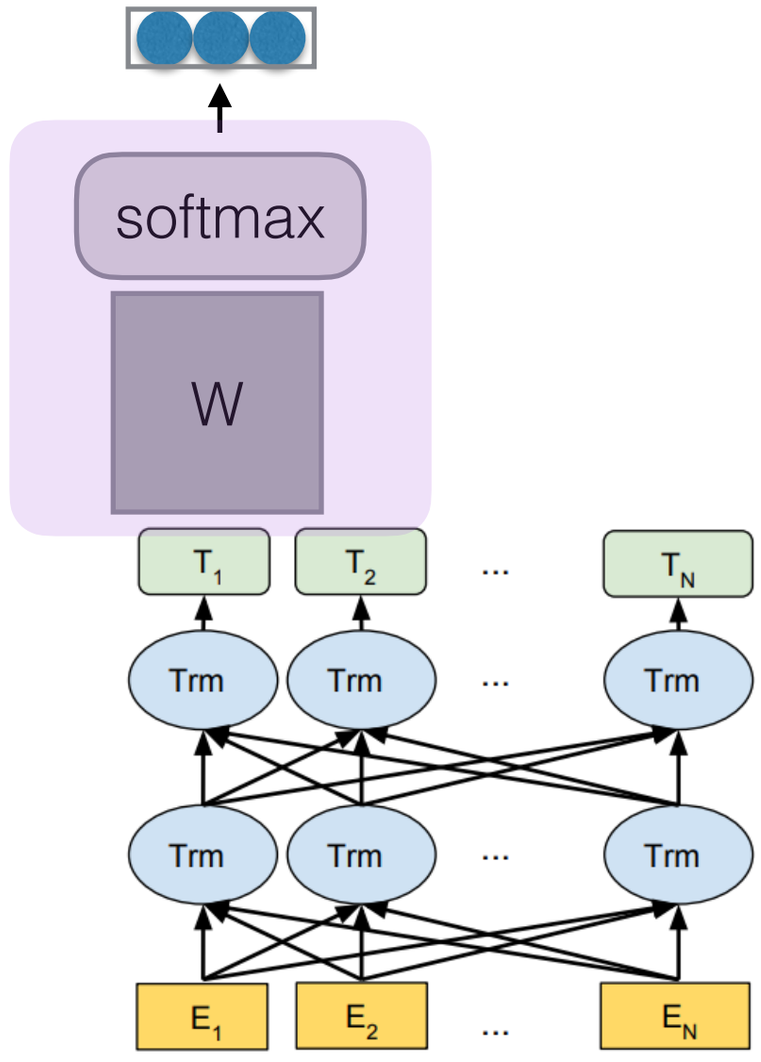
- Add an output layer that maps a hidden vector to scores

- Fine-tune the weights (either just W, or all weights). Example:

  - Data: (movie review, {positive, neutral, negative})

  - Initialize the model with BERT

  - Minimize cross-entropy loss with gradient-based optimization



Adapting a masked language model

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

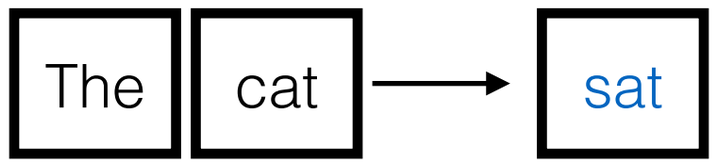
(Devlin et al. 2018)

Today's lecture

- Tasks
- Masked language modeling
- **Autoregressive language modeling**

Autoregressive language modeling

- Predict next token x_t given previous tokens $x_{<t}$



$$\mathscr{L}_{\text{MLE}}(\theta; D) = -\frac{1}{|D|}\sum_{x \in D}\sum_{t=1}^{|x|}\log p_\theta(x_t \mid x_{<t})$$

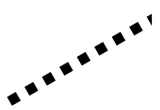
- Maximizes likelihood
  - Fits a data distribution p_*
  - Learns to *compress* data generated by p_*

Maximum likelihood: fits a data distribution

- Makes p_θ match the data distribution p_*

$$\min_\theta D_{KL}(p_* || p_\theta) =$$

Dataset:
samples from p_*



Maximum likelihood: fits a data distribution

- Makes p_θ match the data distribution p_*

$$\min_\theta D_{KL}(p_* || p_\theta) = \min_\theta - \sum_{x \in \mathcal{X}} p_*(x) \log \frac{p_\theta(x)}{p_*(x)}$$

$$\equiv \min_\theta - \sum_{x \in \mathcal{X}} p_*(x) \log p_\theta(x) + \sum_{x \in \mathcal{X}} p_*(x) \log p_*(x)$$

$$= \min_\theta - \mathbb{E}_{x \sim p_*} \log p_\theta(x)$$

$$\approx \min_\theta - \frac{1}{|D|} \sum_{x \in D} \log p_\theta(x)$$

$$\equiv \max_\theta \sum_{x \in D} \log p_\theta(x)$$

CE = KL + Entropy

When $p_\theta = p_*$

CE = Entropy

Maximum likelihood: learns to compress

- Goal: compress data from a distribution p_* into a binary code, $c(x_{1:n}) \to \{0,1\}^*$
- *Arithmetic coding* turns a distribution p into a code $c(\cdot)$
- Minimum expected code length is the entropy [Shannon 1948]:

$$H(p_*) = \mathbb{E}_{x\sim p_*}[-\sum_{i=1}^{N} \log_2 p_*(x_i \,|\, x_{<i})]$$

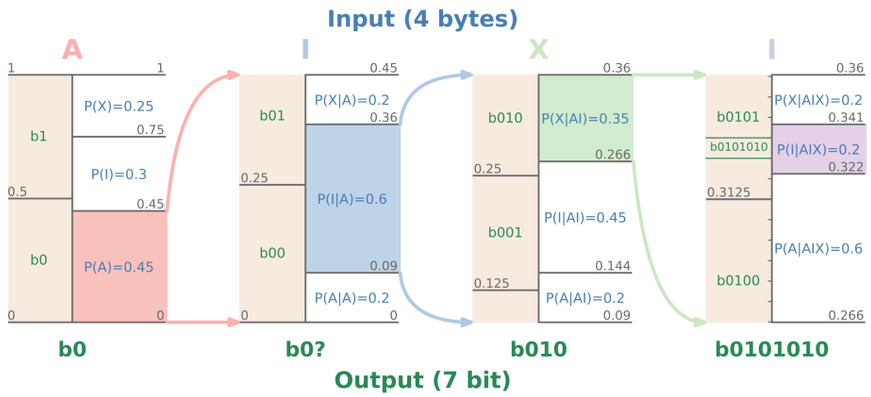


Figure: [Deletang et al 2024]

- When we use a model p_θ, the expected code length is the cross-entropy:

$$H(p_*, p_\theta) = \mathbb{E}_{x\sim p}[-\sum_{i=1}^{N} \log_2 p_\theta(x_i \,|\, x_{<i})]$$

$$H(p_*, p_\theta) = H(p_*) + KL(p_* \| p_\theta)$$

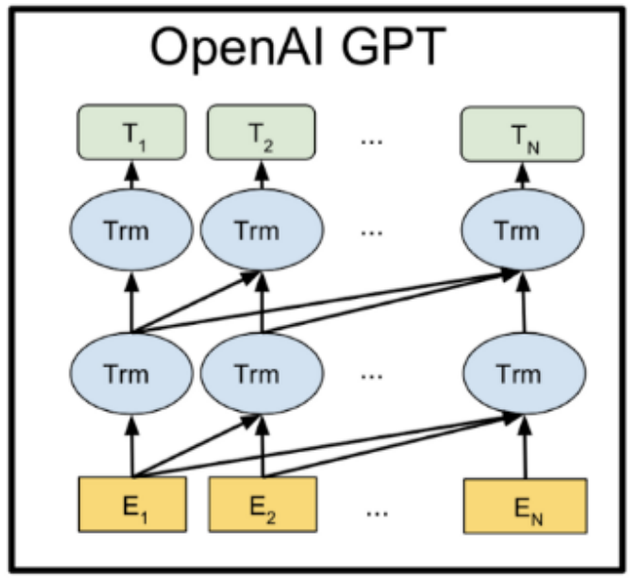
To achieve the minimum expected code length $H(p_*)$, minimize KL divergence via MLE

Key factors

$$\mathscr{L}_{\text{MLE}}(\theta; D) = -\mathbb{E}_{x \sim D} \sum_{t=1}^{|x|} \log p_\theta(x_t \mid x_{<t})$$

- Things we can change:
  - θ: model architecture and size
  - D: training data
  - Optimization hyper-parameters, e.g. learning rate, batch size

Example: GPT-2
[OpenAI 2019]



”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Model	LAMBADA (PPL)	LAMBADA (ACC)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	1.08	18.3	**21.8**
117M	**35.13**	45.99	1.17	37.50	75.20
345M	**15.60**	55.48	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many data

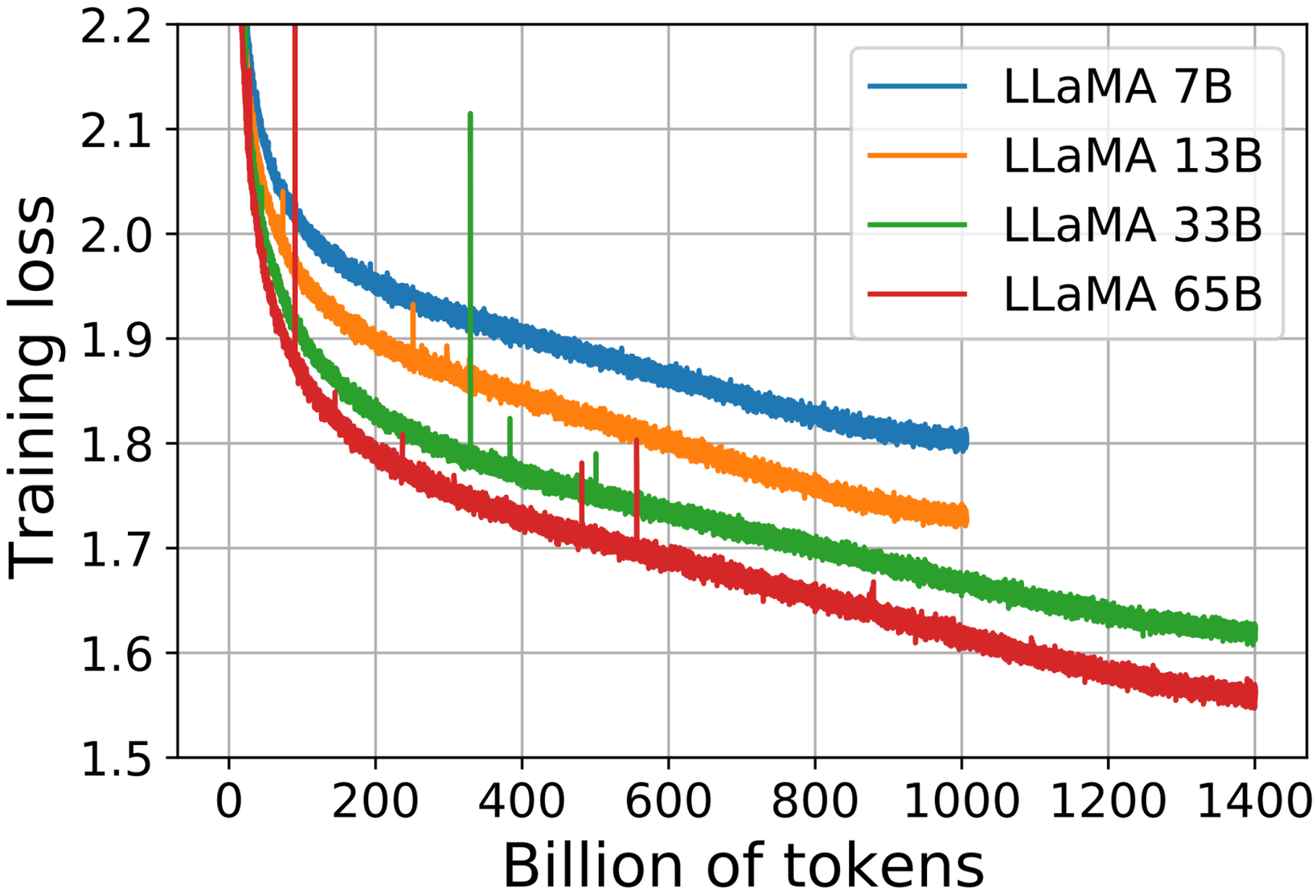
these results. PTB and WikiText-2

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

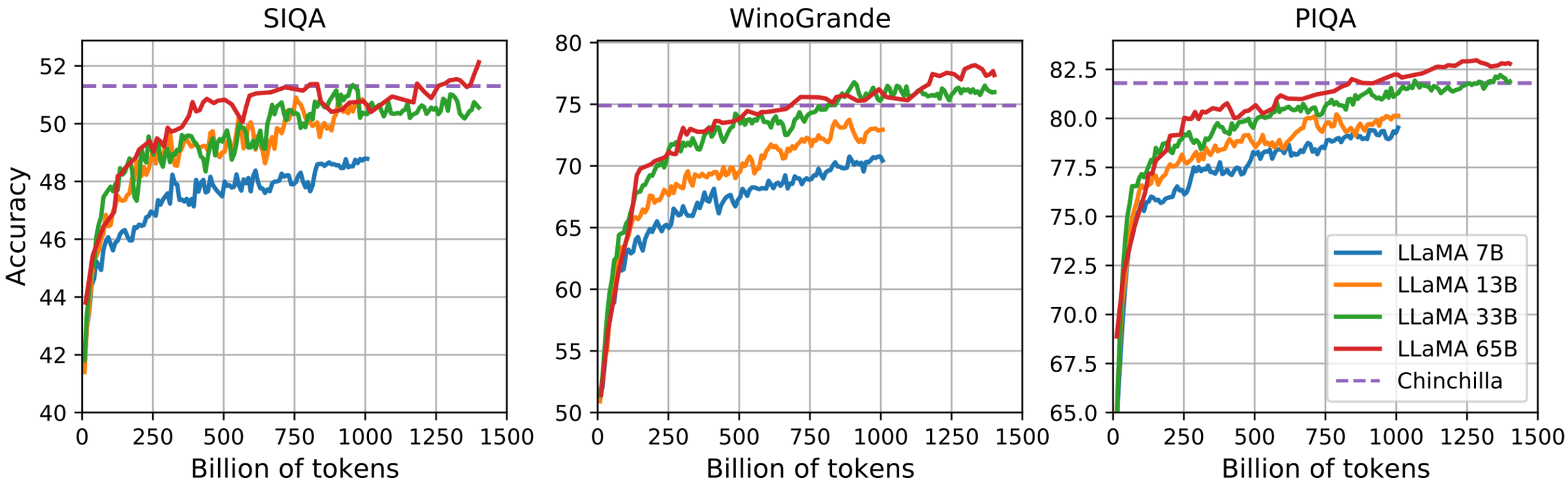


Evaluating a model

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

```
1  Translate English to French:          ← task description
2  sea otter => loutre de mer            ← examples
3  peppermint => menthe poivrée          ←
4  plush girafe => girafe peluche        ←
5  cheese =>                             ← prompt
```

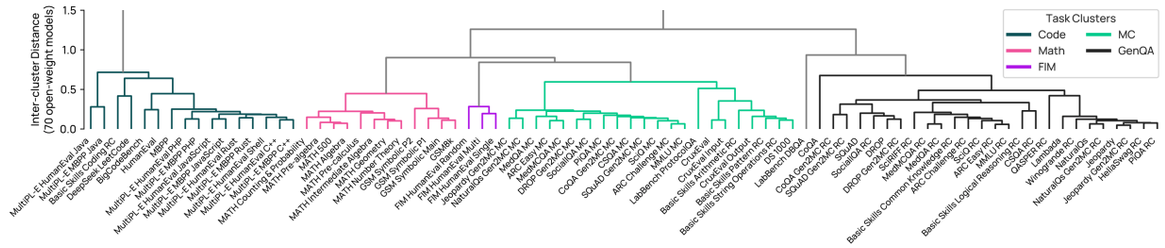
Llama: few-shot performance trajectory



How to choose evaluation tasks?
[AI2 2025]

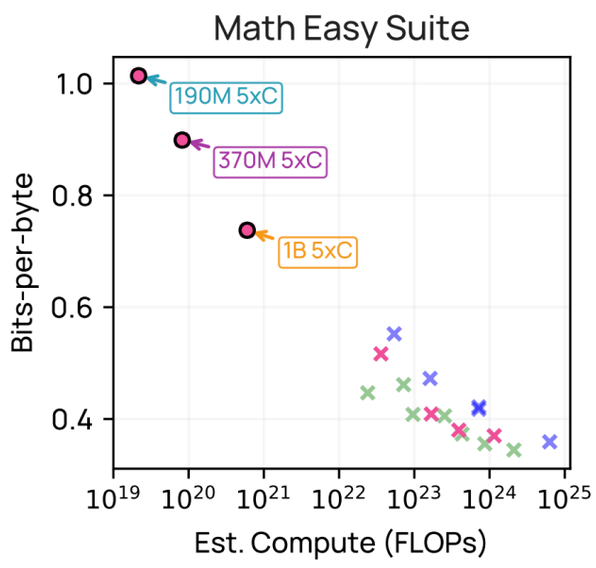
- Cluster into categories

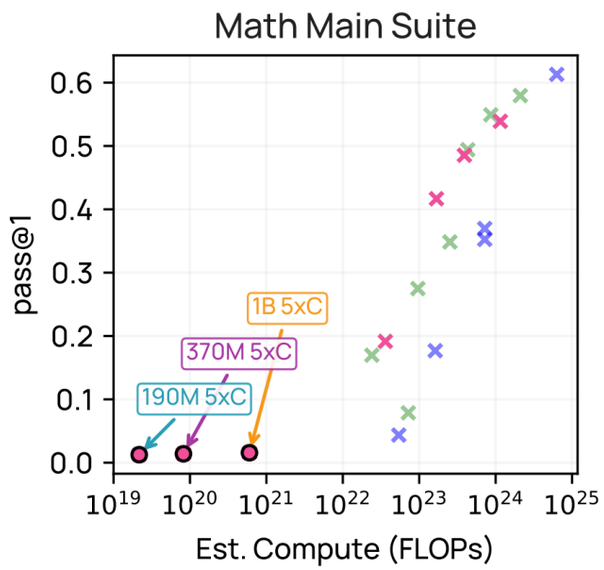
  code math FIM MC QA



  - Tasks are in the same category if they tend to rank models in a similar way

- Separate into tasks that:
  - Give good signal at a small scale
  - Downstream metrics





Practical tools: HuggingFace

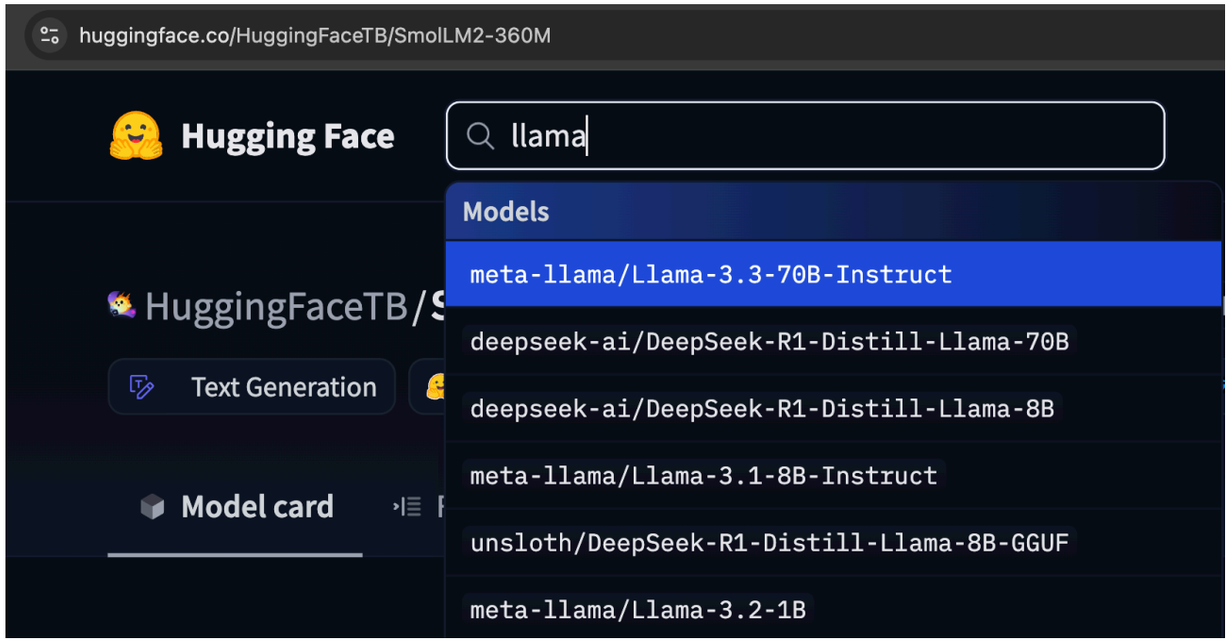
Load tokenizer and model

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

```python
from transformers import AutoTokenizer, AutoModelForCausalLM

model = "HuggingFaceTB/SmolLM2-360M"

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

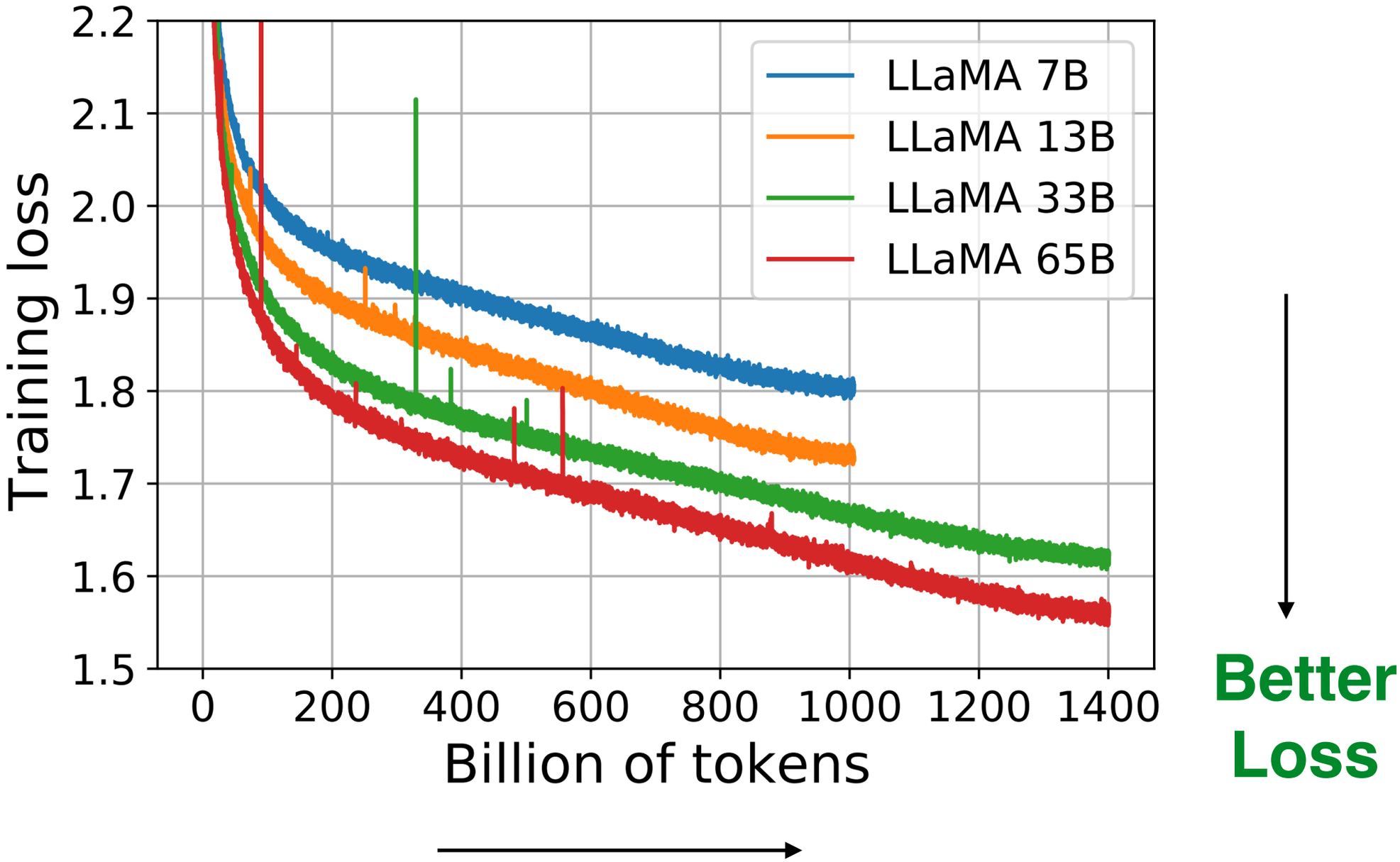


https://github.com/cmu-l3/anlp-spring2026-code/blob/main/06_pretraining/pretraining.ipynb

Today's lecture

- Tasks

- **Data**: sources, quality, and quantity


Training loss
2.2
2.1
2.0
1.9
1.8
1.7
1.6
1.5
0
200
400
600
800
1000
1200
1400
Billion of tokens
LLaMA 7B
LLaMA 13B
LLaMA 33B
LLaMA 65B
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?

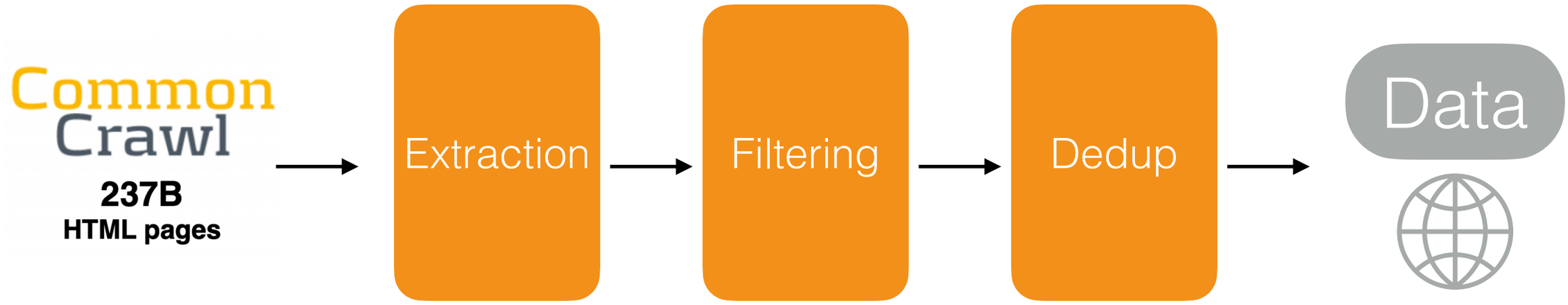
Data quantities

	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

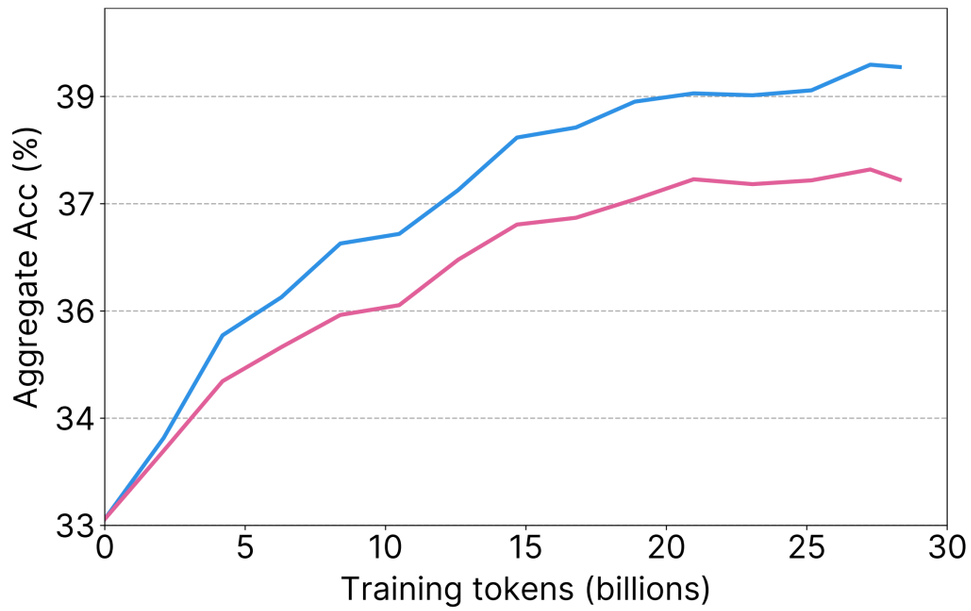
Web data: common crawl

- 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.




```
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

Penedo et al 2024

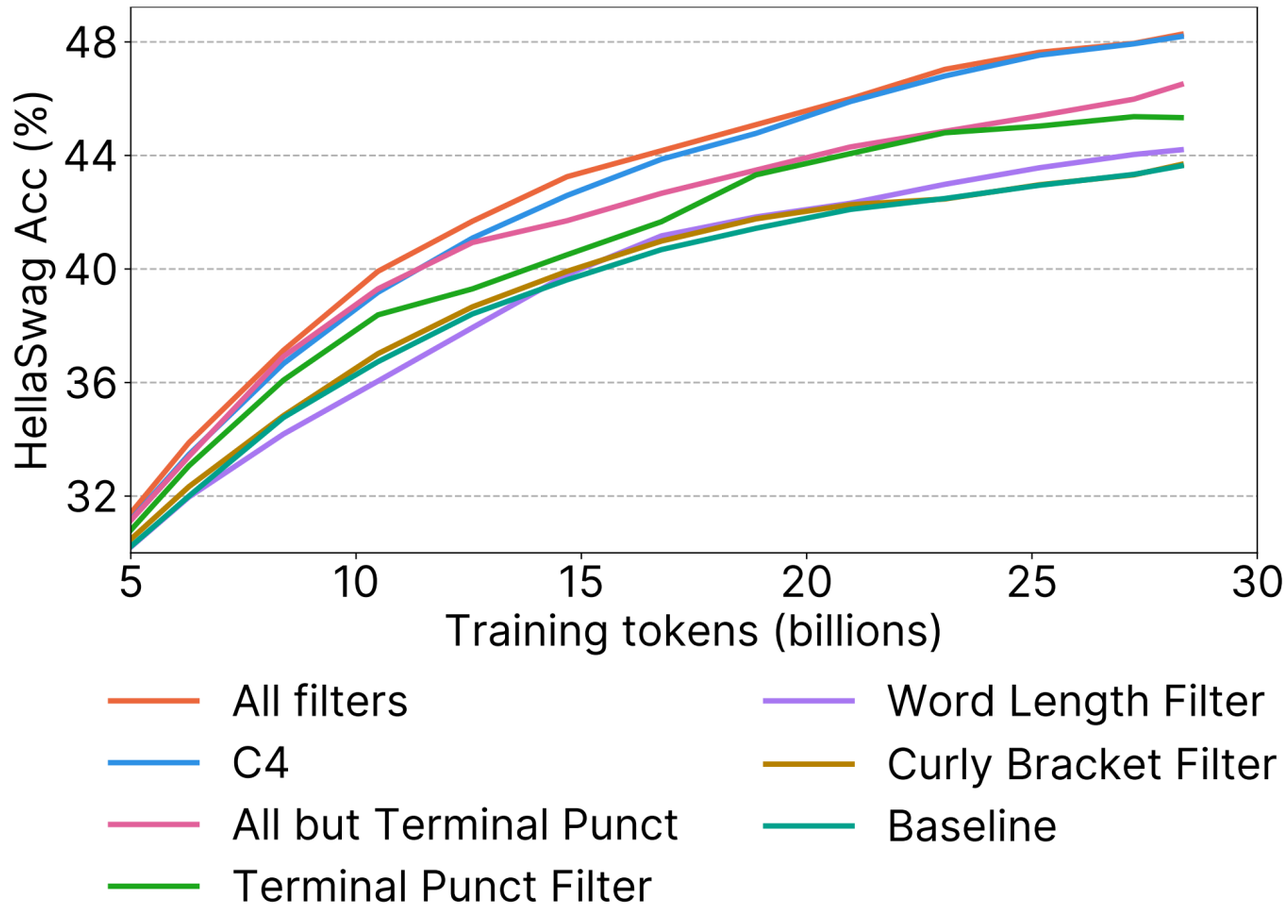
```
    ...
    <annotation ...>
        {\displaystyle \mathrm {MA}
        ={\frac{f_{O}}{f_{E}}}}
    </annotation>
  </semantics>
</math>
```

Special Tags

Paster et al 2023

Quality: Filtering

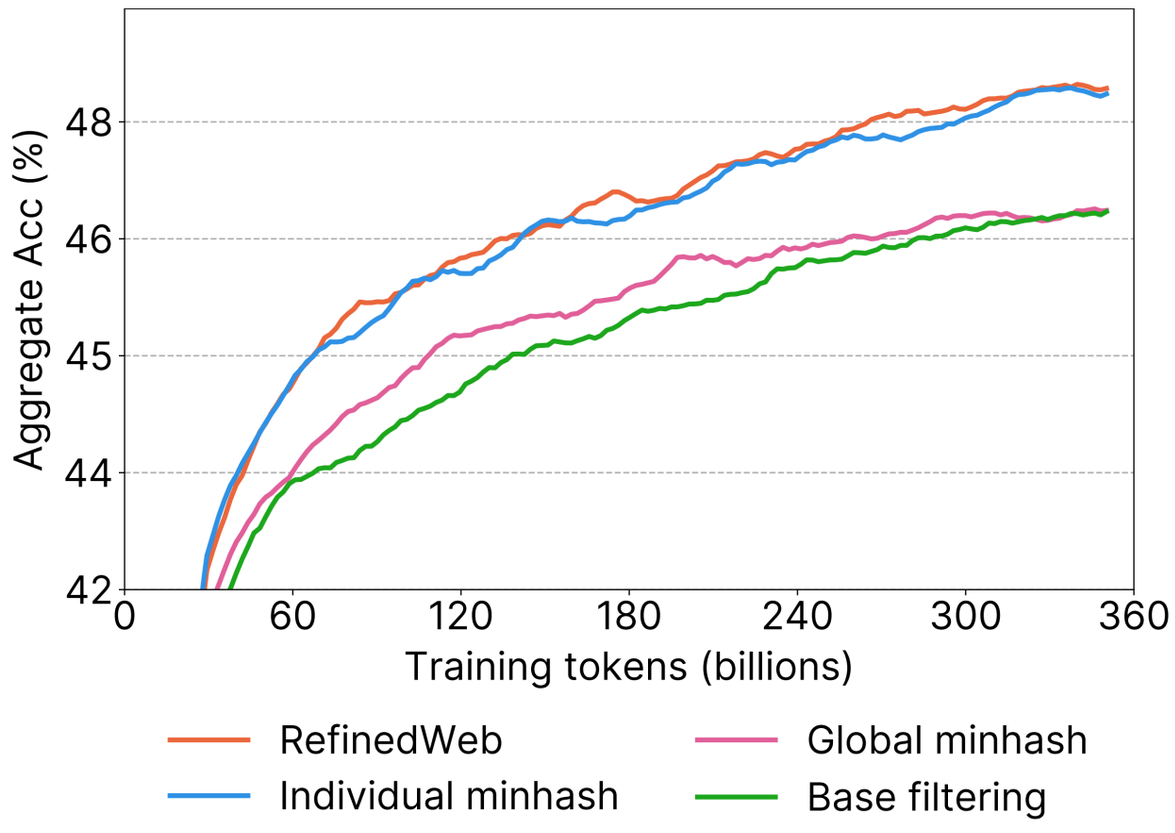
- Filter out unwanted text
  - Language filter
  - Repetitions
  - Too many short lines
  - …



Penedo et al 2024

Quality: Deduplication

- Remove duplicate content
  - Fuzzy strategy: *minhash*
- Too much deduplication can be harmful
  - [Penedo et al 2024]: Deduplicate per-snapshot 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-spring2026-code/blob/main/06_pretraining/pretraining.ipynb

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?

Coverage

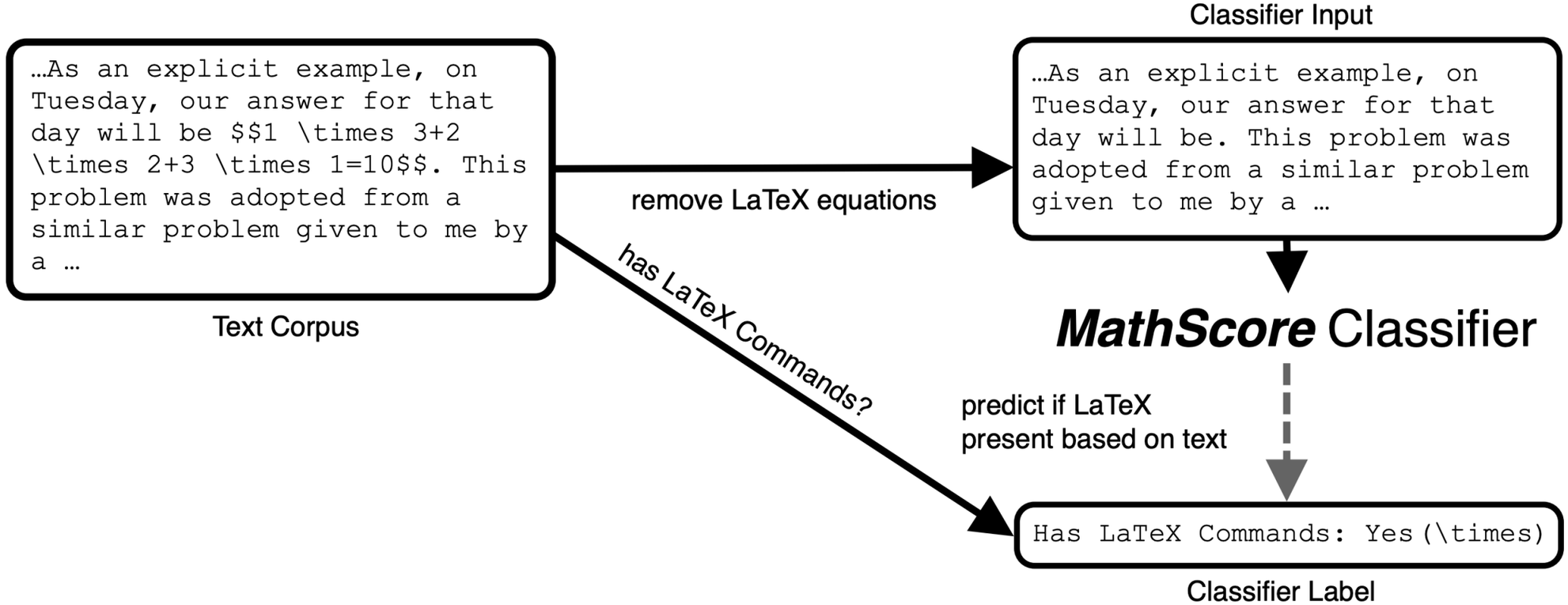
- The data determines the data distribution
  - And hence the model, $p_\theta \approx p_{data}$
- Web data ≠ math data
- Web data ≠ educational data
- Web data ≠ code data
- …

Approach: classifier filtering

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

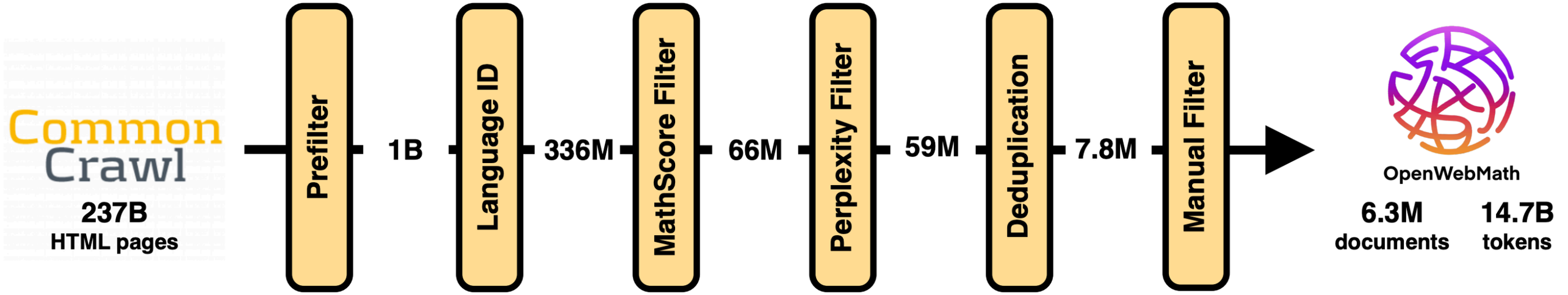
Approach: classifier filtering

- 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

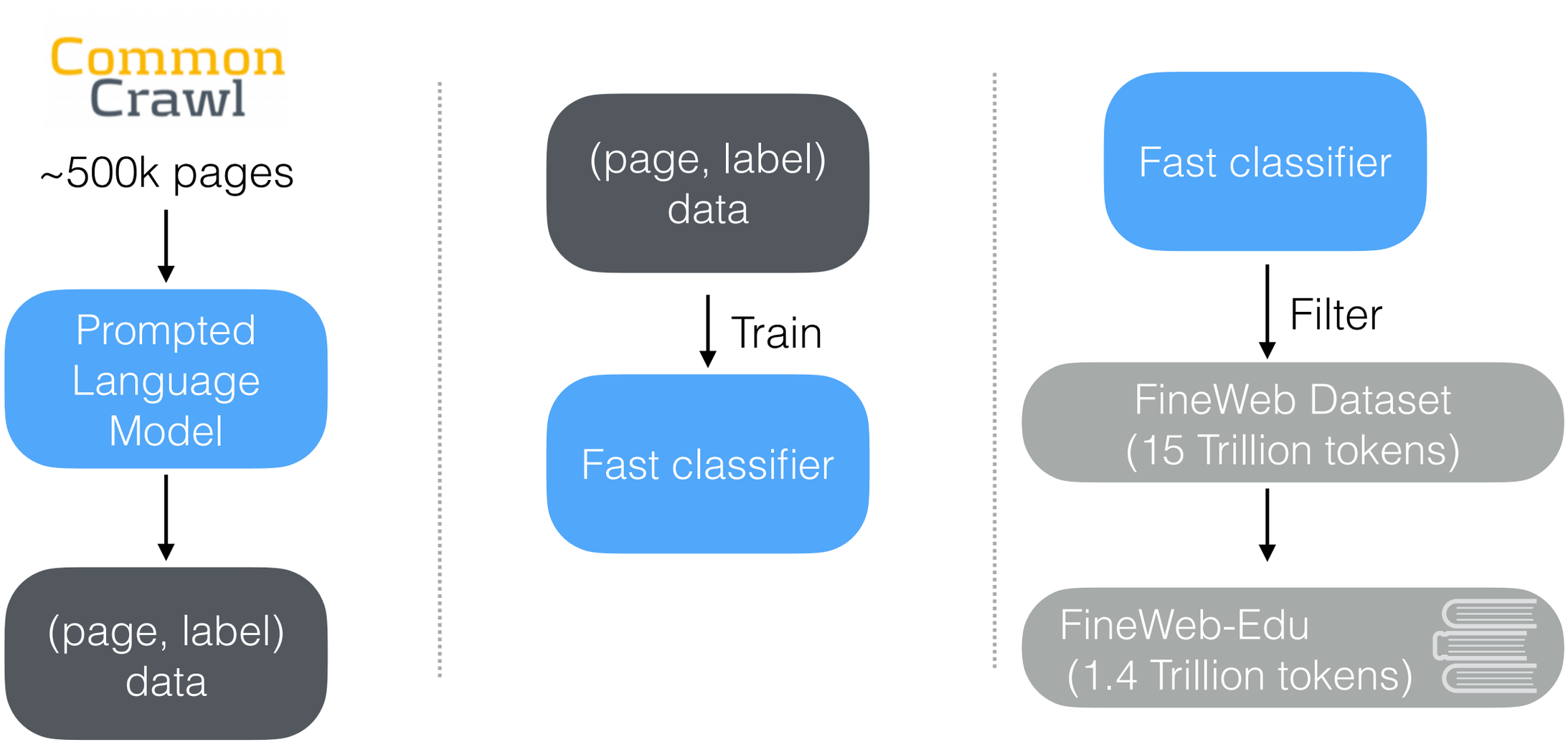
- Example: **OpenWebMath** [Paster et al 2023]

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

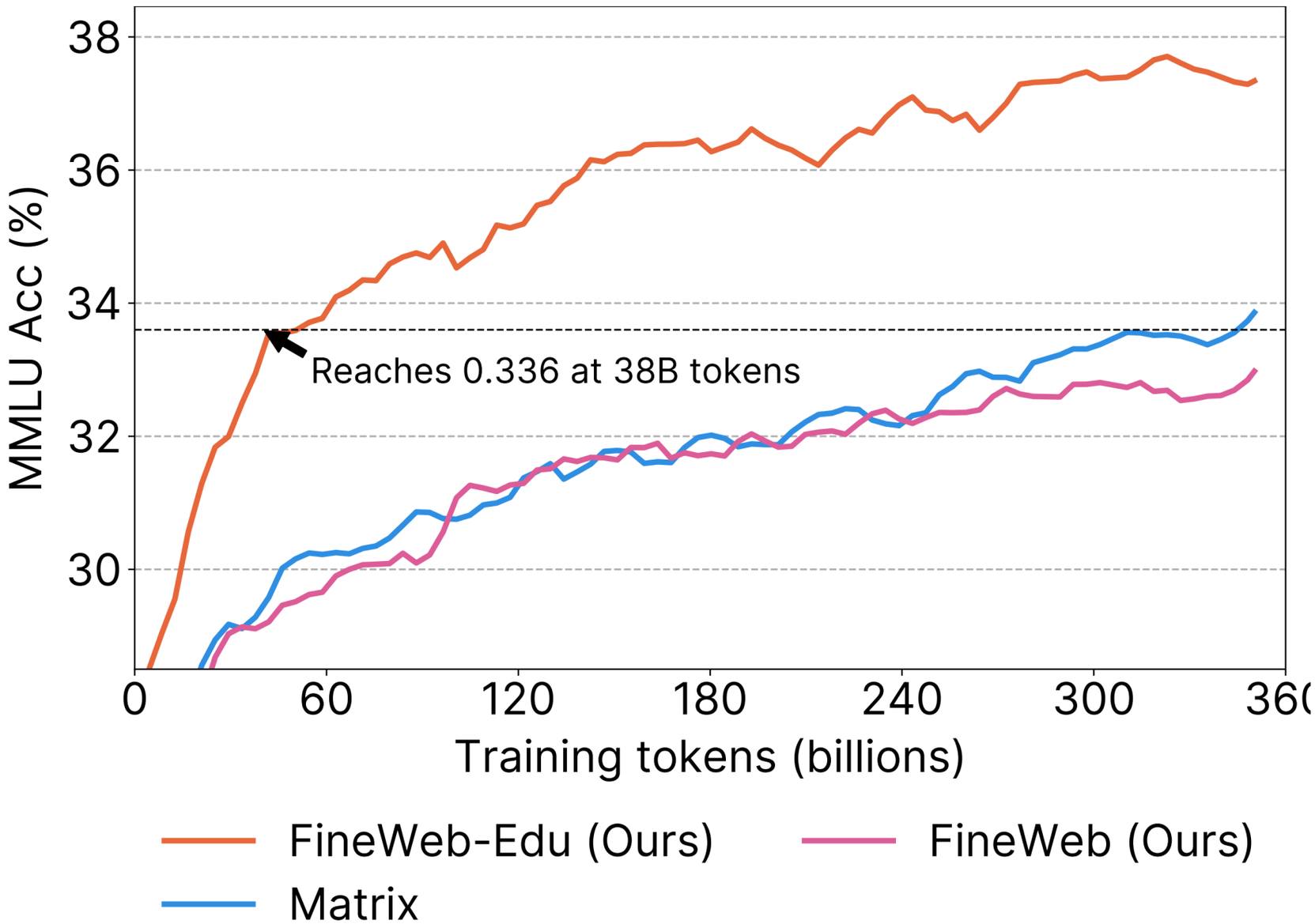
Approach: classifier filtering

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



Approach: classifier filtering

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



Mixtures

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

Source	Type	9T Pool		6T Mix	
		Tokens	Docs	Tokens	Docs
Common Crawl	Web pages	8.14T	9.67B	4.51T (76.1%)	3.15B
olmOCR science PDFs	Academic documents	972B	101M	805B (13.6%)	83.8M
Stack-Edu (Rebalanced)	GitHub code	137B	167M	409B (6.89%)	526M
arXiv	Papers with LaTeX	21.4B	3.95M	50.8B (0.86%)	9.10M
FineMath 3+	Math web pages	34.1B	21.4M	152B (2.56%)	95.5M
Wikipedia & Wikibooks	Encyclopedic	3.69B	6.67M	2.51B (0.04%)	4.24M
Total		**9.31T**	**9.97B**	**5.93T (100%)**	**3.87B**

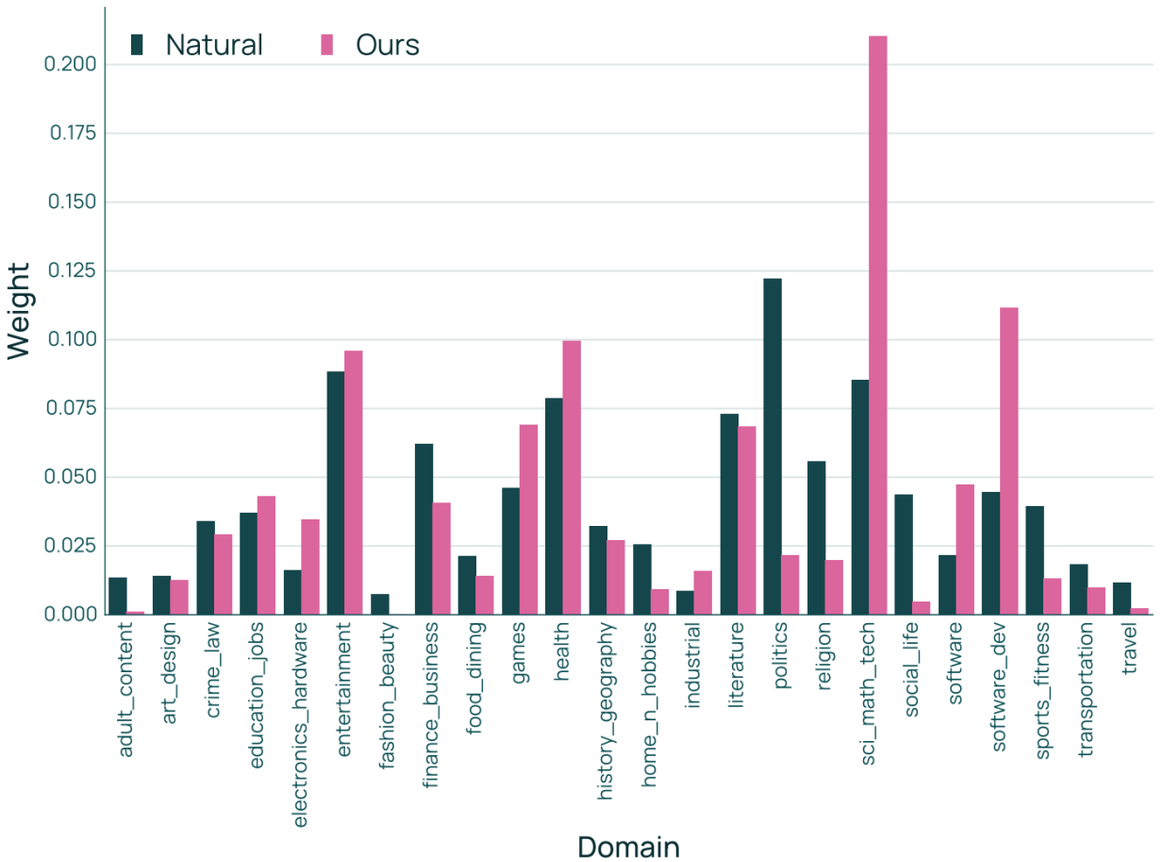
Table 4 Composition of Dolma 3 Mix including our 9T pool of data, the 6T mix we used for f

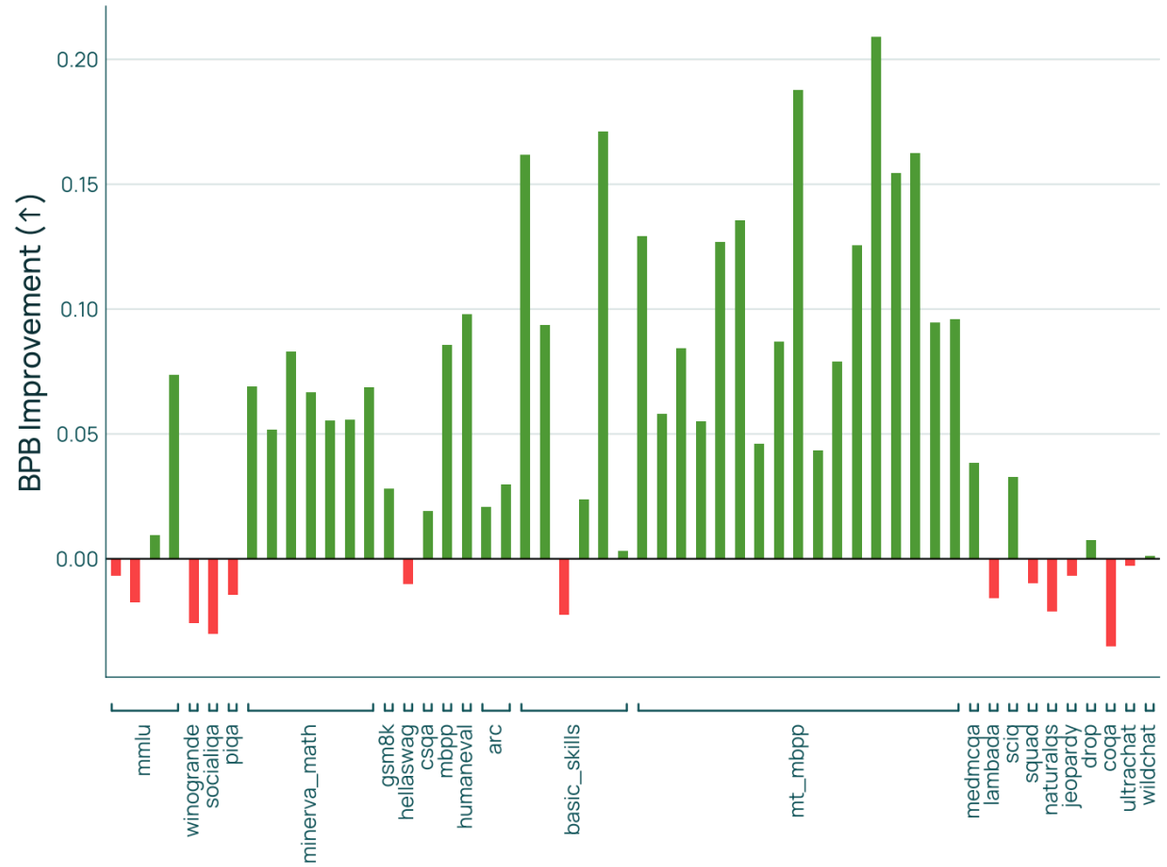
How to choose the mixture?
[AI2 2025]

- Classify documents into topics and quality levels
- Train many small models, each with a different mixture over topics
- For each evaluation task, fit a regression model f(mixture) -> task performance
- Solve for mixture that minimizes average task performance subject to constraints (token budget, maximum repeats)
- Favor high-quality documents when upsampling

How to choose the mixture?

[AI2 2025]





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

Example: Dolma 3 [AI2 2025]

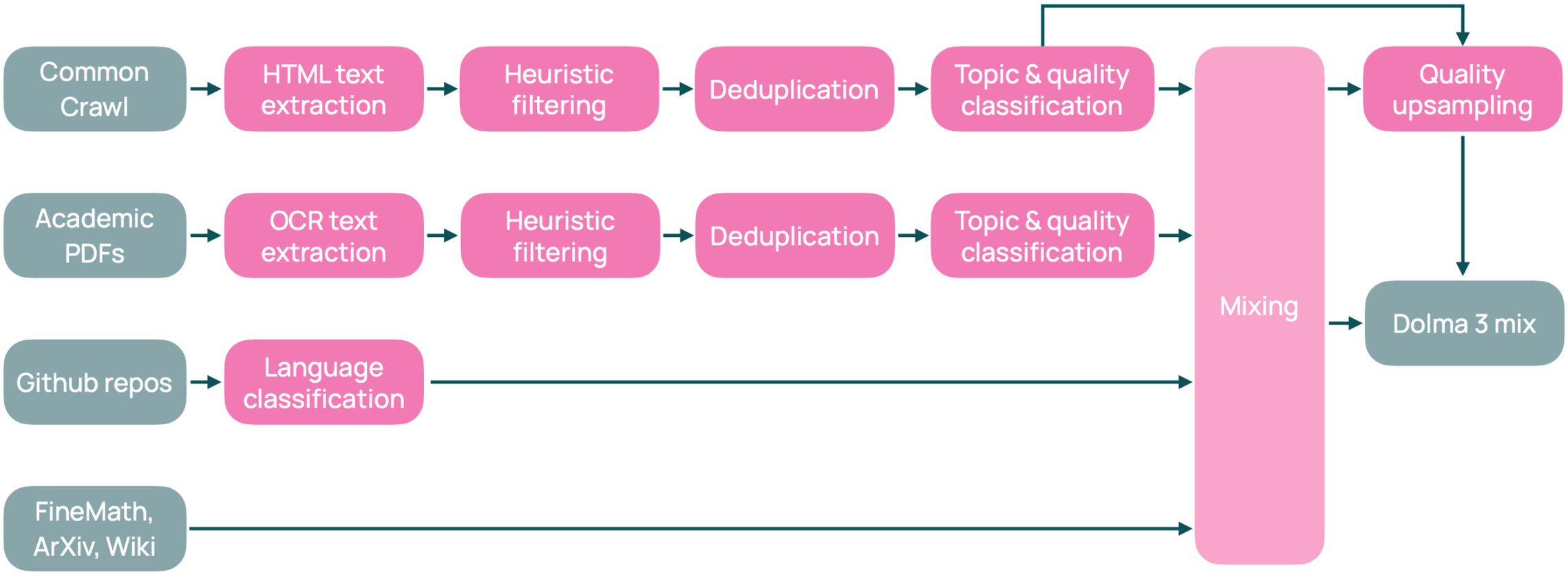
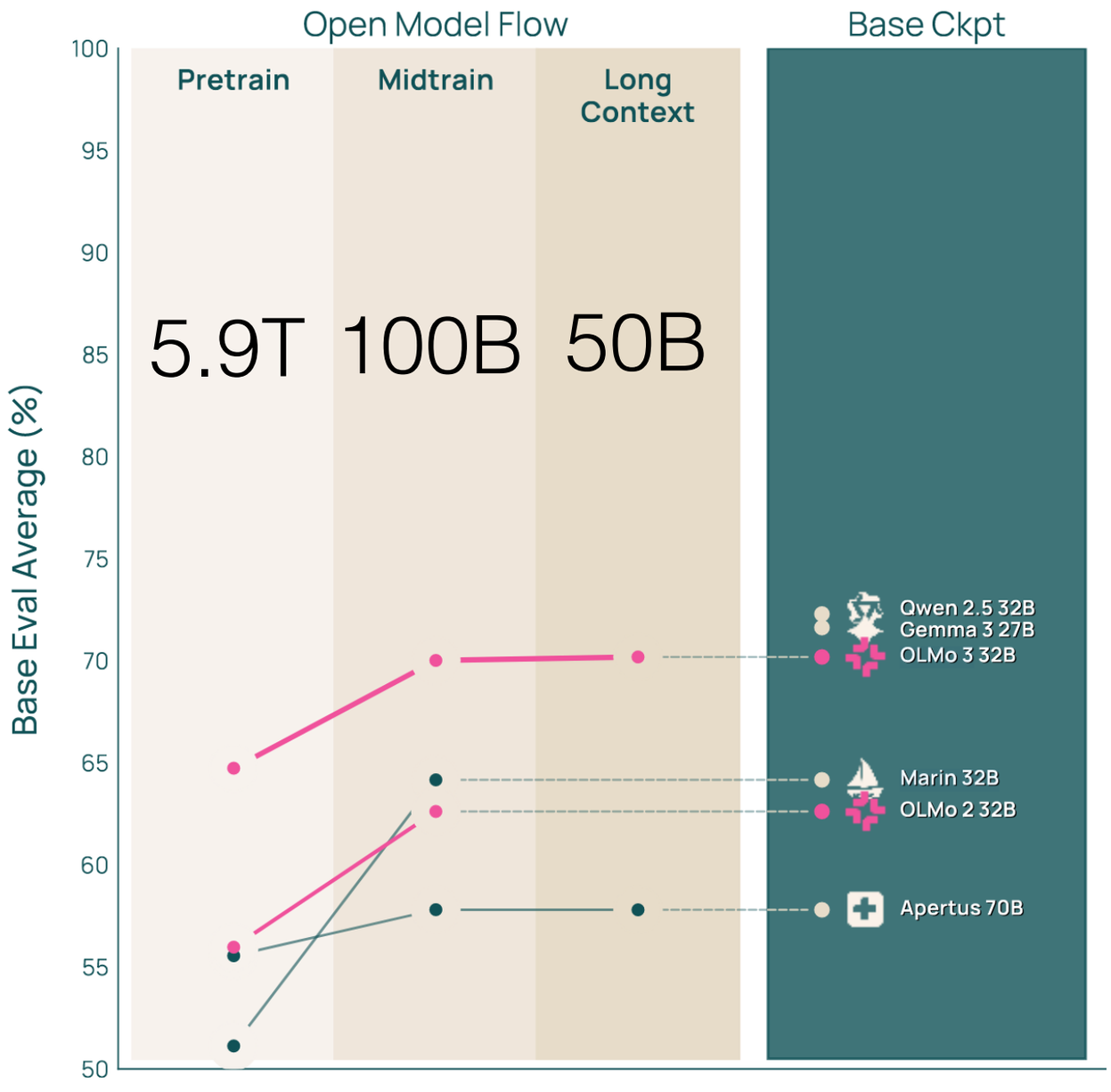


Figure 8 Data curation flow for pretraining data sources in DOLMA 3 MIX.

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
DOLMA 3	2025	Mix	6 trillion
OpenWebMath	2023	Math web pages	15 billion
AlgebraicStack	2023	Math code	11 billion
FineWeb-Edu	2024	Educational (middle-school)	1.4 trillion

OLMO 3 example
[AI2 Dec 2025]



Today's lecture

- Tasks
- Data
- Thinking about pretraining [next lecture]
  - Tokens, model size, compute
  - Scaling laws
- Today: Adam optimizer

Optimizer: Adam

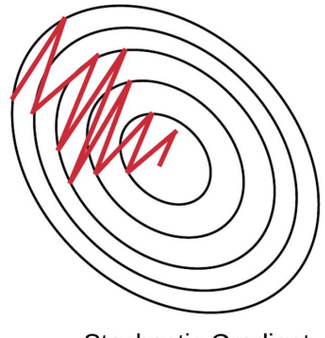
- Most standard optimization option in NLP and beyond
  - Each parameter has an adaptive learning rate
  - Incorporates 2 key ideas: momentum and RMSProp

Optimizer: Adam

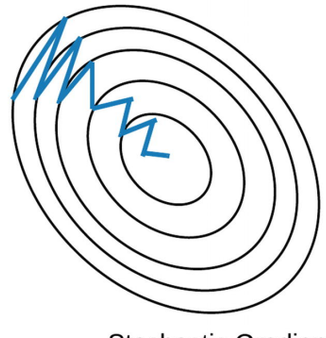
- Momentum

$$\theta_{t+1} = \theta_t - \alpha m_t$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta$$



Stochastic Gradient Descent **withhout** Momentum



Stochastic Gradient Descent **with** Momentum

source

Intuition: reduces oscillations

- $m_t \in \mathbb{R}^{|\theta|}$, i.e. per-parameter adjustment
- Running estimate of $\mathbb{E}[\nabla_\theta]$

Optimizer: Adam

- RMSProp

$$\theta_{t+1} = \theta_t - \frac{\alpha_t}{\sqrt{v_t + \epsilon}} \nabla_\theta$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta)^2$$

- v is per-parameter
- Normalizes the update magnitude
  - $(\nabla_\theta[i,j])^2$ large: update gets smaller
  - $(\nabla_\theta[i,j])^2$ small: update gets larger
- Running estimate of $\mathbb{E}[(\nabla_\theta)^2]$

Optimizer: Adam

- Running estimate of $\mathbb{E}[\nabla_\theta]$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta$$

- Running estimate of $\mathbb{E}[(\nabla_\theta)^2]$

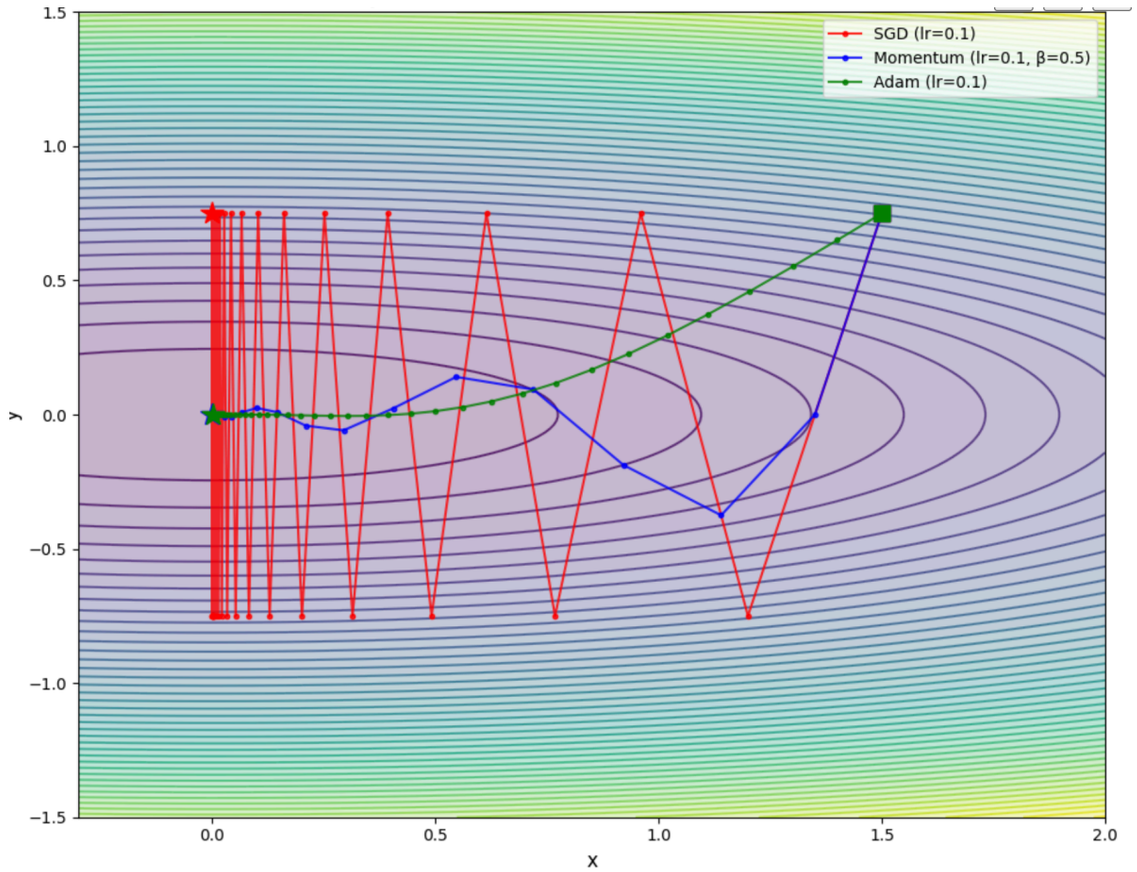
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta)^2$$

- Correction of early bias

$$\hat{m}_t = \frac{m_t}{1 - (\beta_1)^t} \quad \hat{v}_t = \frac{v_t}{1 - (\beta_2)^t}$$

- Final update

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$



Thank you