Not All Tokens Are What You Need

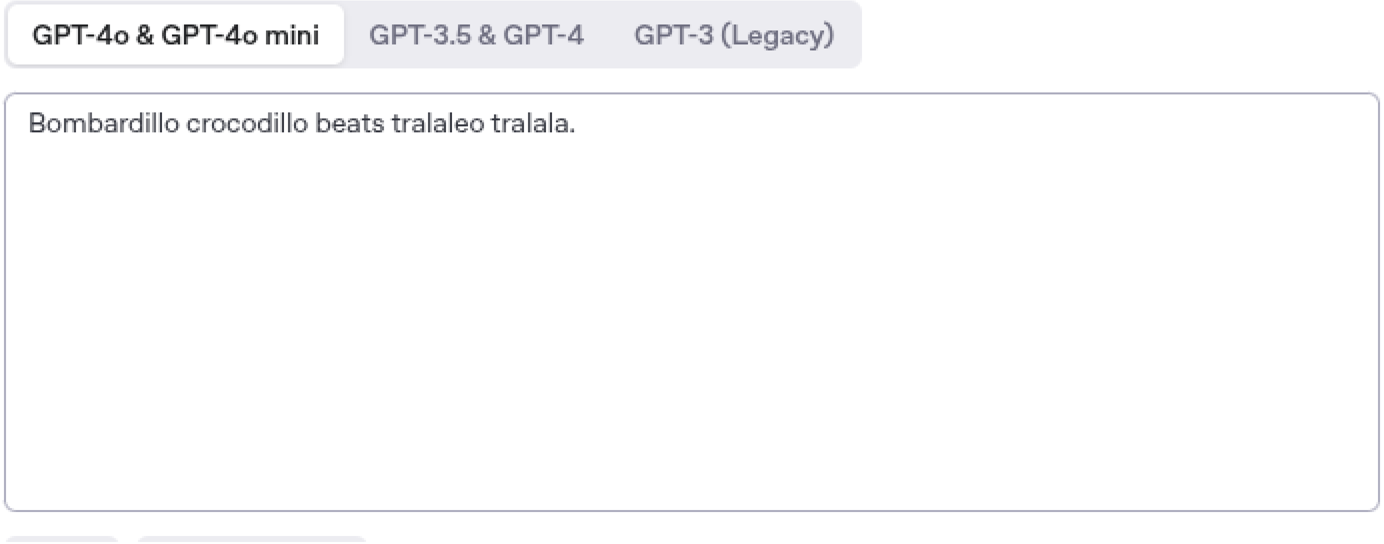
Lin et al., NeurIPS 2024*

Kwanhee Lee, Wonjun Jo, Wonseok Choi

* this paper was selected as an oral presentation, and runner-up for the best paper award

Contents

Tokenization



https://platform.openai.com/tokenizer

Tokenization

Bombardillo crocodillo beats tralaleo tralala.

Text | Token IDs

Tokenization

```
[96273, 597, 16726, 149484, 16726, 54439, 498, 280, 195399, 498, 105994,
13]
```

Text | Token IDs

Causal Language Modeling

Given a language model M parameterized by θ, and a tokenized input sequence $X = \{x_1, x_2, ..., x_n\}$, CLM aims to minimize the next-token prediction loss:

Causal Language Modeling

Given a language model M parameterized by θ, and a tokenized input sequence $X = \{x_1, x_2, ..., x_n\}$, CLM aims to minimize the next-token prediction loss:

$$\min_{\theta} \mathcal{L}(X, M_\theta) \quad \text{where} \quad \mathcal{L}(X, M_\theta) = -logP(x_{n+1}|X, M_\theta)$$

Causal Language Modeling

Given a language model M parameterized by θ, and a tokenized input sequence X = {x_1, x_2, ..., x_n}, CLM aims to minimize the next-token prediction loss:

$$\min_{\theta} \mathcal{L}(X, M_\theta) \quad \text{where} \quad \mathcal{L}(X, M_\theta) = -logP(x_{n+1}|X, M_\theta)$$

This objective encourages the model to assign high likelihood to the (probably) correct next token, given the preceding context (left-to-right)

* Perplexity = exp(L)

Introduction

Large Language Models are trained on vast amount of corpus via casual language modeling, using up to billions and trillions of tokens collected from the internet [1,2]

Training Large Language Models

Large Language Models are trained on vast amount of corpus via casual language modeling, using up to billions and trillions of tokens collected from the internet [1,2]

e.g.) GPT-3 used 300B tokens [1], Chinchilla used 1.4T tokens [2]

Training Large Language Models

Large Language Models are trained on vast amount of corpus via casual language modeling, using up to billions and trillions of tokens collected from the internet [1,2]

e.g.) GPT-3 used 300B tokens [1], Chinchilla used 1.4T tokens [2]

> extremely large corpus are *noisy*

Training Large Language Models

Large Language Models are trained on vast amount of corpus via casual language modeling, using up to billions and trillions of tokens collected from the internet [1,2]

e.g.) GPT-3 used 300B tokens [1], Chinchilla used 1.4T tokens [2]

> extremely large corpus are *noisy*

e.g. low-information, redundancy, mixed language, random words, etc.

- "asdfasdfasdfasdfasdfasdf..."(e.g., keyboard mashing, filler content)
- This article is about deep learning. Deep learning is a type of machine learning. Deep learning is…
- 오늘은 good day for learning! TensorFlowを使って..
- i want hefawef ew><<3 to fjweoifajwemn eat banana.

Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

Data Filtering

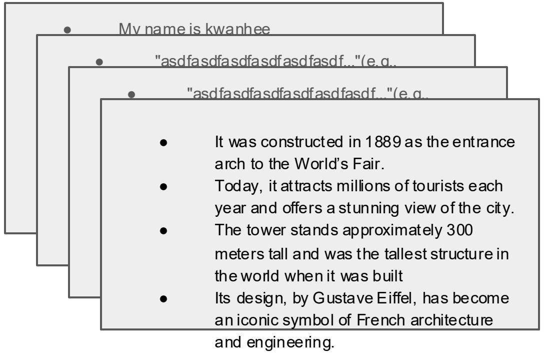
Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- document level filtering : removes entire low-quality documents based on repetition, content safety, etc. [3,4]

Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

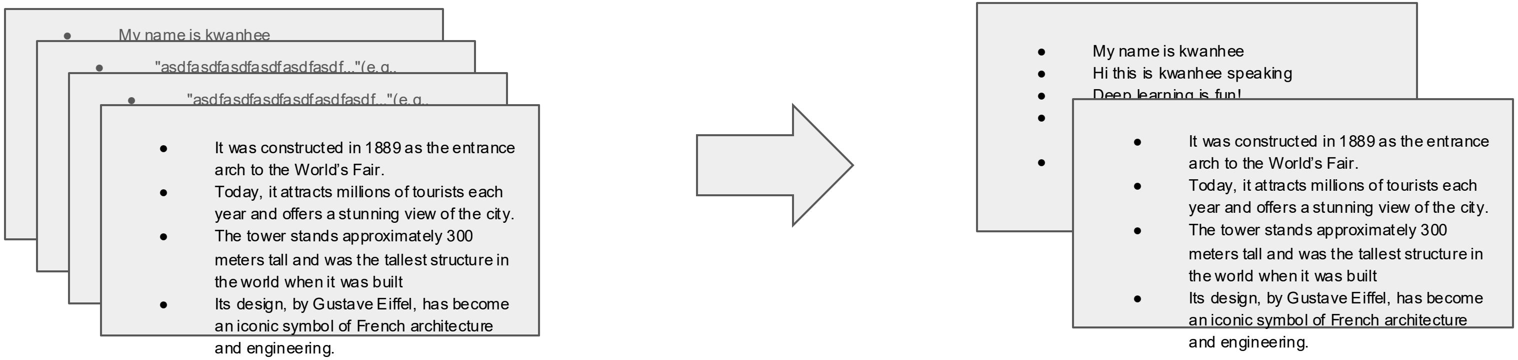
- document level filtering : removes entire low-quality documents based on repetition, content safety, etc. [3,4]



Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- document level filtering : removes entire low-quality documents based on repetition, content safety, etc. [3,4]



Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- line level filtering : removes individual data points (e.g., sentences) [5]

Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

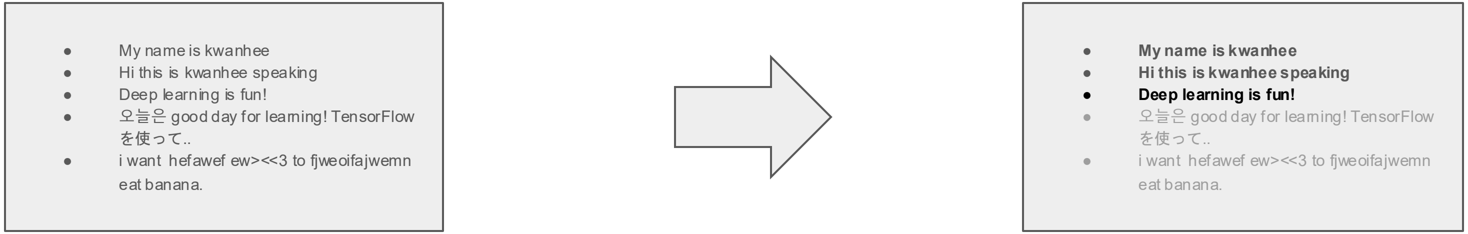
- line level filtering : removes individual data points (e.g., sentences) [5]

- My name is kwanhee
- Hi this is kwanhee speaking
- Deep learning is fun!
- 오늘은 good day for learning! TensorFlow を使って..
- i want hefawef ew><<3 to fjweoifajwemn eat banana.

Data Filtering

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- line level filtering : removes individual data points (e.g., sentences) [5]



Rho-loss

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- line level filtering : removes individual data points (e.g., sentences) [5]
    - Rho-loss
        - robust data selection method that filters data points based on reducible holdout loss

Rho-loss

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

- line level filtering : removes individual data points (e.g., sentences) [5]
  - Rho-loss
    - robust data selection method that filters data points based on reducible holdout loss

$$\underset{(x,y)\in B_t}{\arg\max} \overbrace{\underbrace{L[y \mid x; \mathcal{D}_{\mathrm{t}}]}_{\text{training loss}} - \underbrace{L[y \mid x; \mathcal{D}_{\mathrm{ho}}]}_{\text{irreducible holdout loss (IL)}}}^{\text{reducible holdout loss}}$$

More Fine-grained Filtering?

Removing noisy data - also known as *data filtering* - is crucial for improving LLM training performance/efficiency.

> is there more *fine-grained* approach?

Nature of Causal Language Modeling

e.g. i want hefawef ew><<3 to fjweoifajwemn eat banana.

Nature of Causal Language Modeling

e.g. i want hefawef ew><<3 to fjweoifajwemn eat banana.

- Humans can focus on important tokens to process the sentence

  > I want to eat banana

Nature of Causal Language Modeling

e.g. i want hefawef ew><<3 to fjweoifajwemn eat banana.

- Humans can focus on important tokens to process the sentence

  > I want to eat banana

- Language models can’t do this!

  > i want hefawef ew><<3 to fjweoifajwemn eat banana.

Research Question

Given that data filtering can improve performance and considering the nature of causal language modeling,

Q) Are all tokens necessary for pretraining?

Research Question

Given that data filtering can improve performance and considering the nature of causal language modeling,

Q) Are all tokens necessary for pretraining?

A) No!

Research Question

Given that data filtering can improve performance and considering the nature of causal language modeling,

Q) Are all tokens necessary for pretraining?

A) No!

Q)Then, how can we select tokens?

Training Dynamics of Token Loss

Base Model:

Tinyllama-1B

Math Dataset:

15B OpenWebMath

Loss Evaluation:

Evaluate token loss every 1B tokens

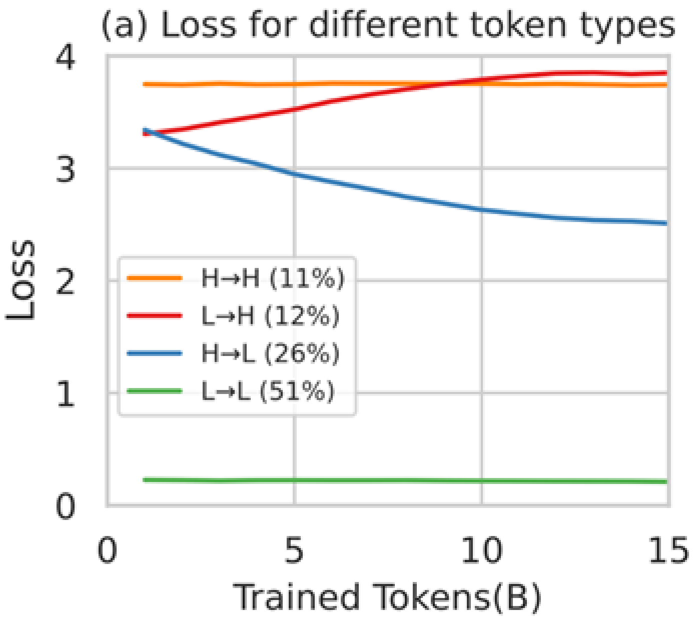
Fit loss trends for each token

Classify into four categories

Token Types:

- H→H : $(-0.2 \leq \Delta\mathcal{L} \leq 0.2 \; and \; l_n > \mathcal{L}_{mean})$
- L→H : $(\Delta\mathcal{L} > 0.2)$
- H→L : $(\Delta\mathcal{L} < 0.2)$
- L→L : $(-0.2 \leq \Delta\mathcal{L} \leq 0.2 \; and \; l_n \leq \mathcal{L}_{mean})$

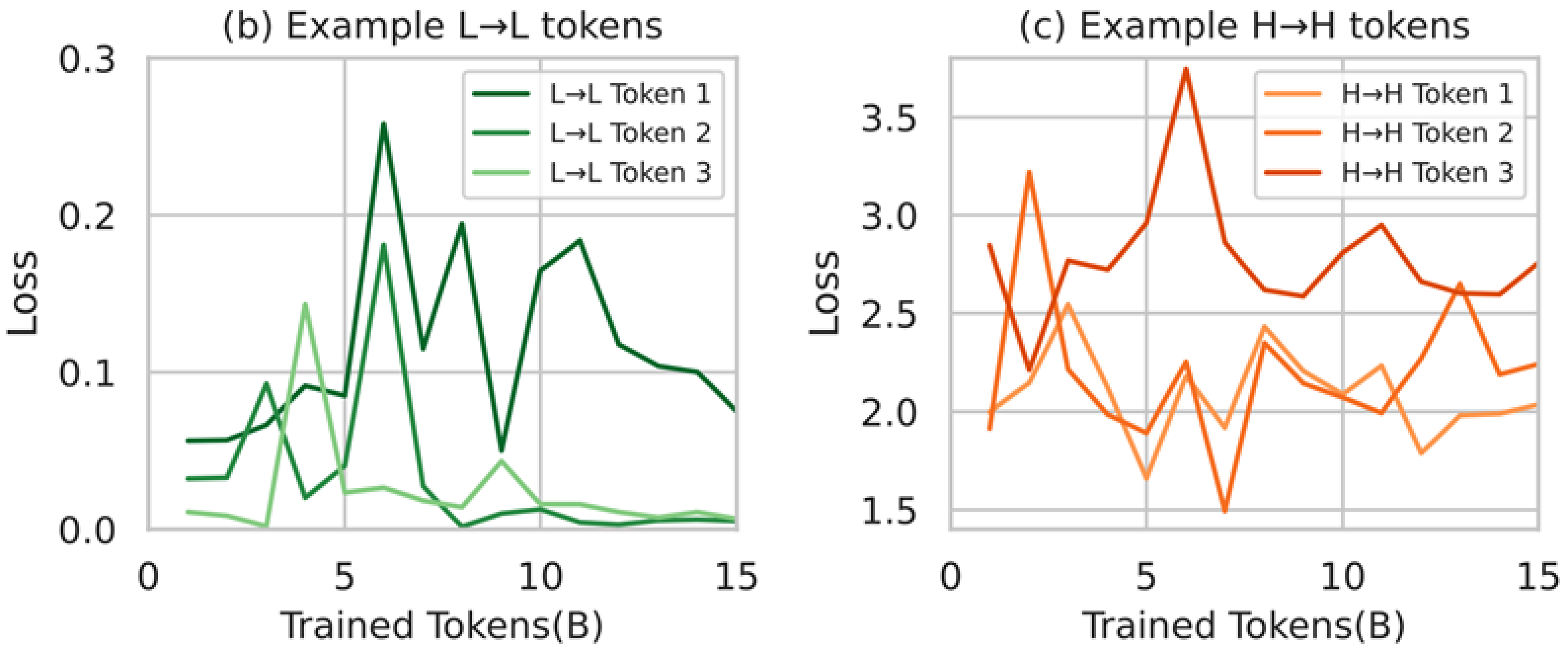
Not All Tokens Are Equal



Token Types:

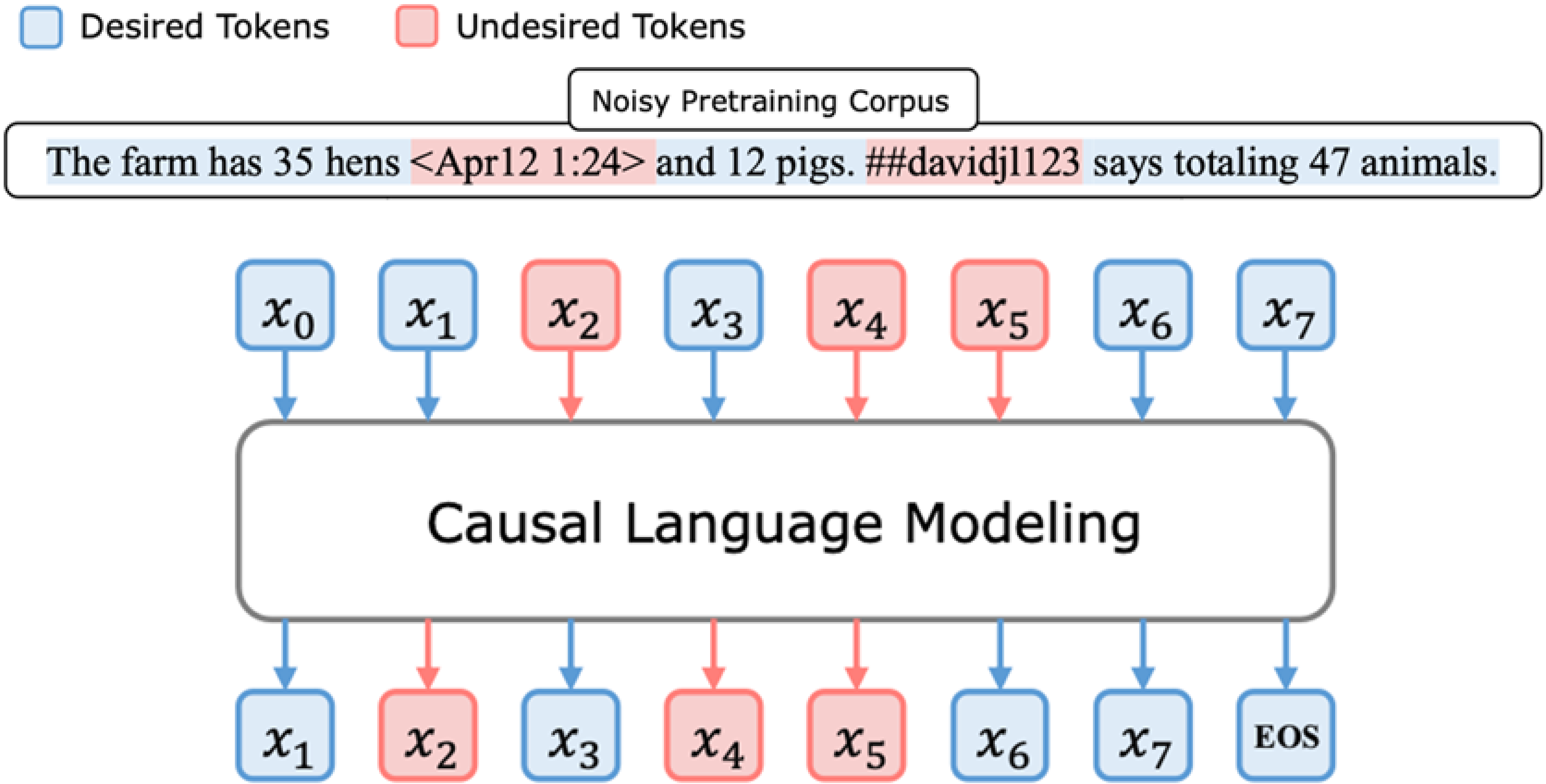
- H → H (11%): Persistent high loss, stay hard
- L → H (12%): Increasing loss, may indicate noise
- H → L (26%): Decreasing loss, ideal for learning
- L → L (51%): Consistent low loss, already known

Not All Tokens Are Equal

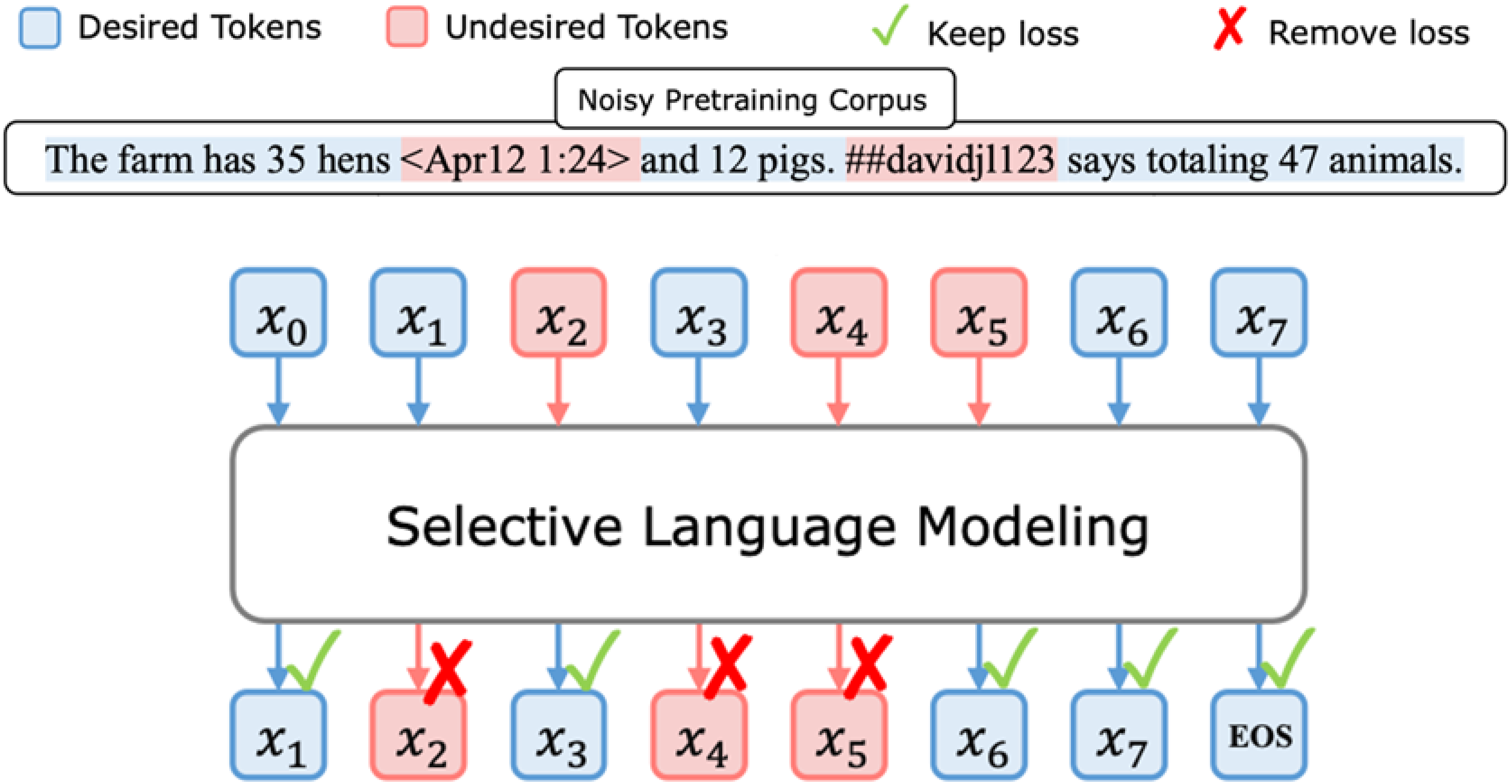


"fluctuating" tokens that resist convergence

Can We Select Useful Tokens?

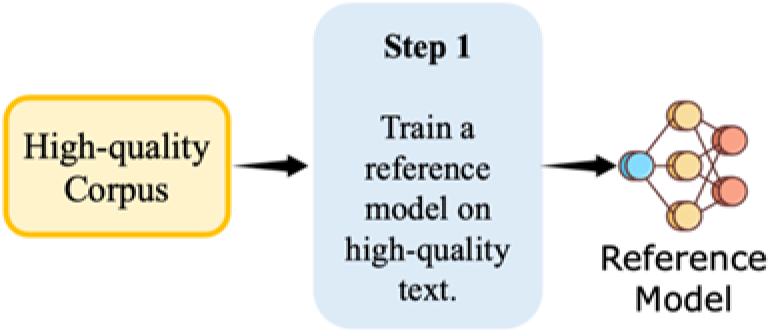


Can We Select Useful Tokens?

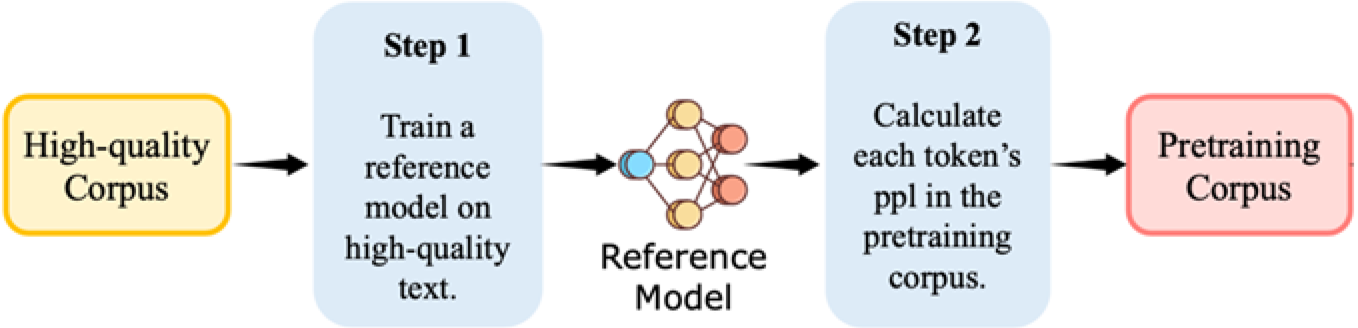


Selective Language Modeling (SLM)

Selective Language Modeling (SLM)



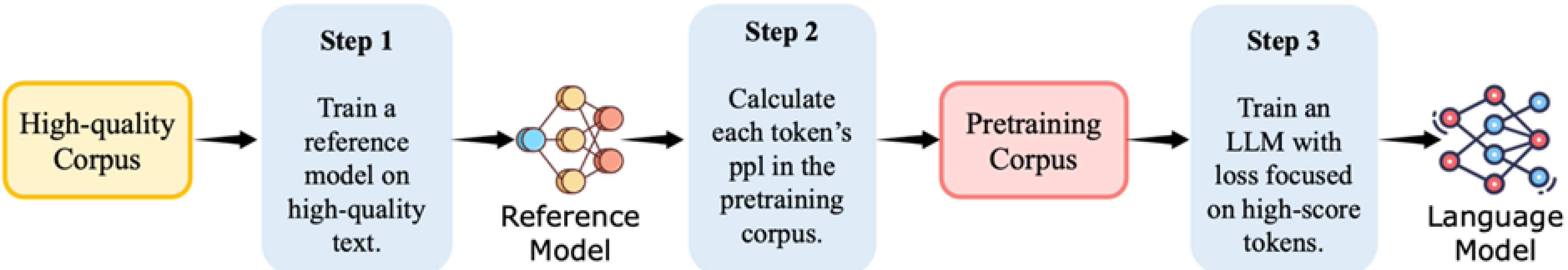
Selective Language Modeling (SLM)



Token Scoring

$$\underbrace{\mathcal{L}_\Delta(x_i)}_{\text{Margin}} = \underbrace{\mathcal{L}_\theta(x_i)}_{\text{Target model}} - \underbrace{\boxed{\mathcal{L}_{\mathrm{RM}}(x_i)}}_{\text{Reference model}}$$

Selective Language Modeling (SLM)



Token Scoring

$$\underbrace{\mathcal{L}_{\Delta}(x_i)}_{\text{Margin}} = \underbrace{\boxed{\mathcal{L}_{\theta}(x_i)}}_{\text{Target model}} - \underbrace{\mathcal{L}_{\text{RM}}(x_i)}_{\text{Reference model}}$$

Token Selection

$$I_{k\%}(x_i) = \begin{cases} 1 & \text{if } x_i \text{ ranks in the top } k\% \text{ by } S(x_i) \\ 0 & \text{otherwise} \end{cases}$$

SLM Training

$$\mathcal{L}_{\text{SLM}}(\theta) = -\frac{1}{N * k\%} \sum_{i=1}^{N} I_{k\%}(x_i) \cdot \log P(x_i | x_{<i}; \theta)$$

Token Selection Example

$$\underbrace{\mathcal{L}_{\Delta}(x_i)}_{\text{Margin}} = \underbrace{\mathcal{L}_{\theta}(x_i)}_{\text{Target model}} - \underbrace{\mathcal{L}_{\mathrm{RM}}(x_i)}_{\text{Reference model}}$$

"Tom had 4 apples. He ate 2. How many are left?"

	$\mathcal{L}_{\theta}$	$\mathcal{L}_{RM}$	$\mathcal{L}_{\Delta}$	Selected
4	1.85	0.90	0.95	✓
apples	0.75	0.55	0.20	✓
2	1.95	0.88	1.07	✓
How	1.10	0.70	0.40	✓
left	1.00	0.60	0.40	✓
Tom	0.35	0.25	0.10	✗
ate	0.65	0.55	0.10	✗

Experimental Setup

- Reference Model (**RM**) Training
    - Dataset
        - Math domain
            - **0.5B** data from GPT and manually curated data
        - General domain
            - **1.9B** tokens from open-source datasets
    - Model
        - Tinyllama-1.1B (Pre-trained)
        - Mistral-7B (Pre-trained)

Experimental Setup

- Reference Model (**RM**) Training
  - Dataset
    - Math domain
      - **0.5B** data from GPT and manually curated data
    - General domain
      - **1.9B** tokens from open-source datasets
  - Model
    - Tinyllama-1.1B (Pre-trained)
    - Mistral-7B (Pre-trained)

- Language Model (**LM**) Training
  - Dataset
    - Math domain
      - **14B** OpenWebMath (OWM) dataset
    - General domain
      - **80B** tokens from open-source datasets
  - Model
    - Tinyllama-1.1B (Pre-trained)
    - Mistral-7B (Pre-trained)

Experimental Setup

- Reference Model (**RM**) Training
  - Dataset
    - Math domain
      - **0.5B** data from GPT and manually curated data
    - General domain
      - **1.9B** tokens from open-source datasets
  - Model
    - Tinyllama-1.1B (Pre-trained)
    - Mistral-7B (Pre-trained)

- Language Model (**LM**) Training
  - Dataset
    - Math domain
      - **14B** OpenWebMath (OWM) dataset
    - General domain
      - **80B** tokens from open-source datasets
  - Model
    - Tinyllama-1.1B (Pre-trained)
    - Mistral-7B (Pre-trained)

- Baseline (-CT)
  - Without token selection
- RHO-1
  - With token selection

Pre-training Results on Math Domain

Model	$\lvert\theta\rvert$	Data	Uniq. Toks*	Train Toks	GSM8K	MATH†	SVAMP	ASDiv	MAWPS	TAB	MQA	MMLU STEM	SAT‡	AVG
1-2B Base Models														
Tinyllama	1.1B	-	-	-	2.9	3.2	11.0	18.1	20.4	12.5	14.6	16.1	21.9	13.4
Phi-1.5	1.3B	-	-	-	32.4	4.2	43.4	53.1	66.2	24.4	14.3	21.8	18.8	31.0
Qwen1.5	1.8B	-	-	-	36.1	6.8	48.5	63.6	79.0	29.2	25.1	31.3	40.6	40.0
Gemma	2.0B	-	-	-	18.8	11.4	38.0	56.6	72.5	36.9	26.8	34.4	50.0	38.4
DeepSeekLLM	1.3B	OWM	14B	150B	11.5	8.9	-	-	-	-	-	29.6	31.3	-
DeepSeekMath	1.3B	-	120B	150B	23.8	13.6	-	-	-	-	-	33.1	56.3	-
Continual Pretraining on Tinyllama-1B														
Tinyllama-CT	1.1B	OWM	14B	15B	6.4	2.4	21.7	36.7	47.7	17.9	13.9	23.0	25.0	21.6
RHO-1-Math	1.1B	OWM	14B	9B	29.8	14.0	49.2	61.4	79.8	25.8	30.4	24.7	28.1	38.1
Δ				-40%	+23.4	+11.6	+27.5	+24.7	+32.1	+7.9	+16.5	+1.7	+3.1	**+16.5**
RHO-1-Math	1.1B	OWM	14B	30B	36.2	15.6	52.1	67.0	83.9	29.0	32.5	23.3	28.1	40.9
≥ 7B Base Models														
LLaMA-2	7B		-	-	14.0	3.6	39.5	51.7	63.5	30.9	12.4	32.7	34.4	31.4
Mistral	7B		-	-	41.2	11.6	64.7	68.5	87.5	52.9	33.0	49.5	59.4	52.0
Minerva	8B	-	39B	164B	16.2	14.1	-	-	-	-	-	35.6	-	-
Minerva	62B	-	39B	109B	52.4	27.6	-	-	-	-	-	53.9	-	-
Minerva	540B	-	39B	26B	58.8	33.6	-	-	-	-	-	63.9	-	-
LLemma	7B	PPile	55B	200B	38.8	17.2	56.1	69.1	82.4	48.7	41.0	45.4	59.4	50.9
LLemma	34B	PPile	55B	50B	54.2	23.0	67.9	75.7	90.1	57.0	49.8	54.7	68.8	60.1
Intern-Math	7B	-	31B	125B	41.8	14.4	61.6	66.8	83.7	50.0	57.3	24.8	37.5	48.7
Intern-Math	20B	-	31B	125B	65.4	30.0	75.7	79.3	94.0	50.9	38.5	53.1	71.9	62.1
DeepSeekMath	7B	-	120B	500B	64.1	34.2	74.0	83.9	92.4	63.4	62.4	56.4	84.4	68.4
Continual Pretraining on Mistral-7B														
Mistral-CT	7B	OWM	14B	15B	42.9	22.2	68.6	71.0	86.1	45.1	47.7	52.6	65.6	55.8
RHO-1-Math	7B	OWM	14B	10.5B	66.9	31.0	77.8	79.0	93.9	49.9	58.7	54.6	84.4	66.2
Δ				-30%	+24.0	+8.8	+9.2	+8.0	+7.8	+4.8	+11.0	+2.0	+18.8	**+10.4**

Experimental Setup

- Dataset
  - 14B OpenWebMath
- Model
  - Tinyllama-1.1B
  - Mistral-7B
- Task
  - Few-shot CoT Reasoning

Supervised Fine-Tuning Results on Math Domain

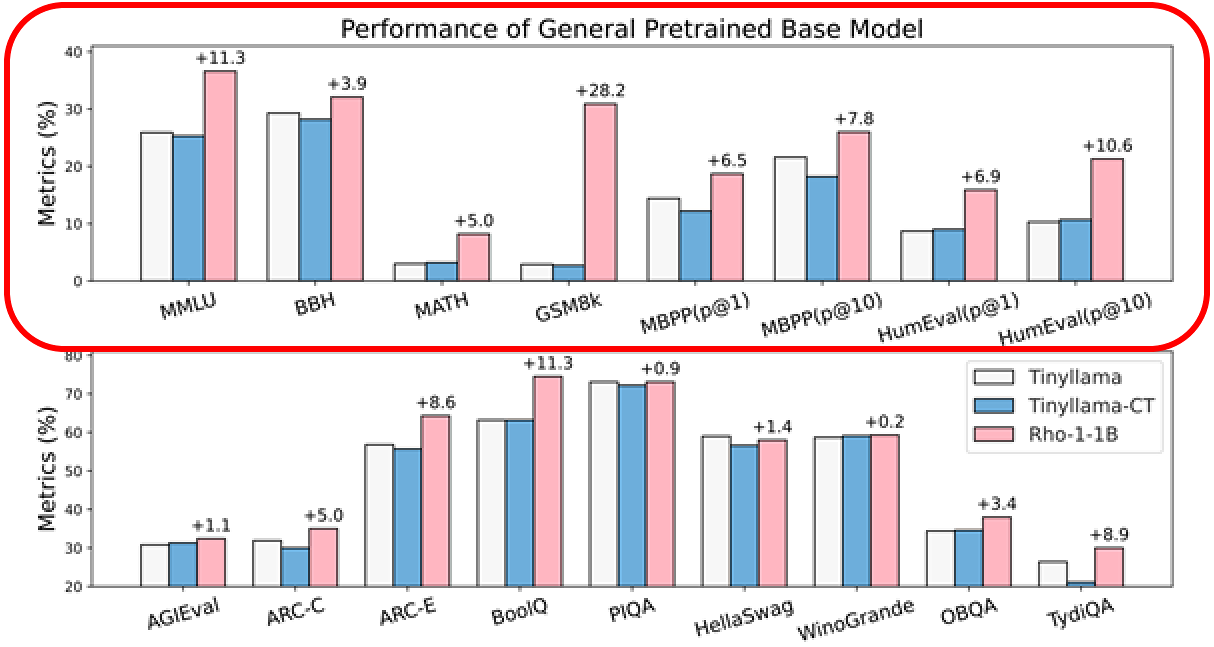
Model	Size	Tools	SFT Data	GSM8k	MATH	SVAMP	ASDiv	MAWPS	TAB	GSM-H	AVG
Used for SFT?				✓	✓	✗	✗	✗	✗	✗	
Previous Models											
GPT4-0314	-	✗	-	92.0	42.5	93.1	91.3	97.6	67.1	64.7	78.3
GPT4-0314 (PAL)	-	✓	-	94.2	51.8	94.8	92.6	97.7	95.9	77.6	86.4
MAmmoTH	70B	✓	MI-260k	76.9	41.8	82.4	-	-	-	-	-
ToRA	7B	✓	ToRA-69k	68.8	40.1	68.2	73.9	88.8	42.4	54.6	62.4
ToRA	70B	✓	ToRA-69k	84.3	49.7	82.7	86.8	93.8	74.0	67.2	76.9
DeepSeekMath	7B	✓	ToRA-69k	79.8	52.0	80.1	87.1	93.8	85.8	63.1	77.4
Our Pretrained Models											
TinyLlama-CT	1B	✓	ToRA-69k	51.4	38.4	53.4	66.7	81.7	20.5	42.8	50.7
RHO-1-Math	1B	✓	ToRA-69k	59.4	40.6	60.7	74.2	88.6	26.7	48.1	56.9
Δ				+8.0	+2.2	+7.3	+7.5	+6.9	+6.2	+5.3	**+6.2**
Mistral-CT	7B	✓	ToRA-69k	77.5	48.4	76.9	83.8	93.4	67.5	60.4	72.6
RHO-1-Math	7B	✓	ToRA-69k	81.3	51.8	80.8	85.5	94.5	70.1	63.1	75.3
Δ				+3.8	+3.4	+3.9	+1.7	+1.1	+2.6	+2.7	**+2.7**

Experimental Setup

- Dataset
  - ToRA-69k
- Model
  - Rho-1-Math-1B
  - Rho-1-Math-7B
- Task
  - Tool-Integrated Reasoning

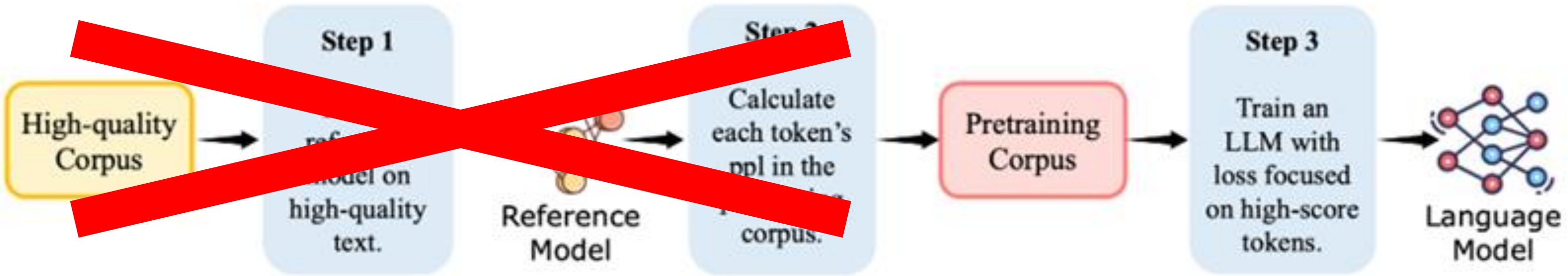
Results on General Domain

- Around 6% average boost in performance in general domain
- Improvement is significant on math-related benchmarks
  - Likely due to clear structure and explicit attention targets such as formulas.



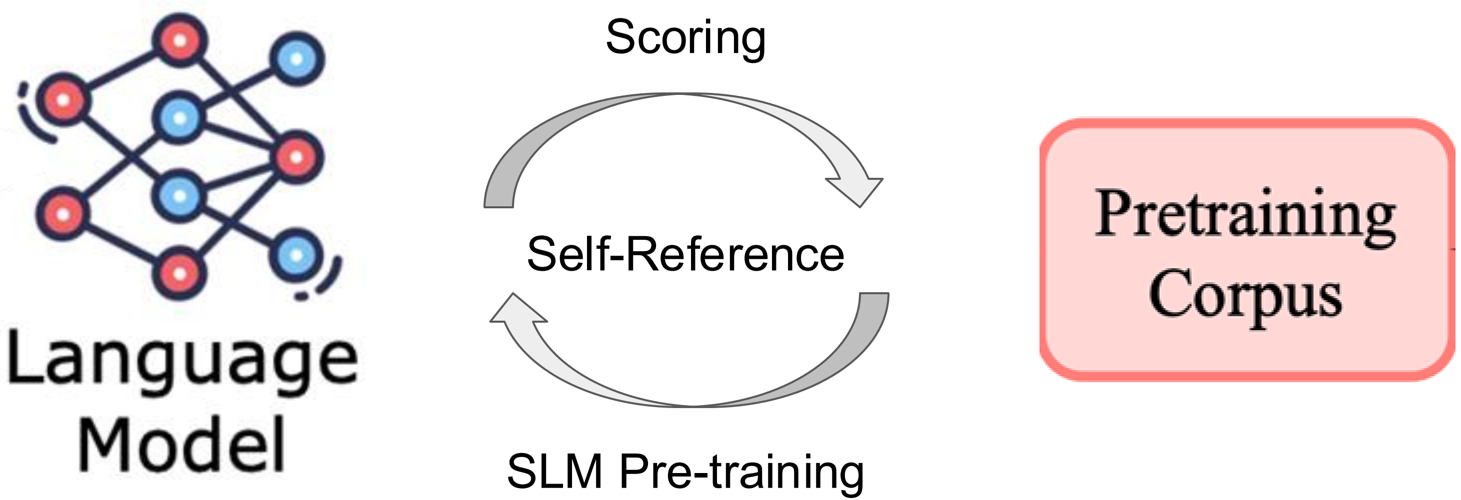
Can SLM Works w/o High-Quality Corpus?

- We can't always assume there is a high-quality corpus
- What if there is no high-quality data?
    - We cannot do step 1 and 2



Can SLM Works w/o High-Quality Corpus?

- Self-reference scenario
  - **Case1**: Train a model with full data to the end first, and use it as the reference model
  - Case2: Use different previous checkpoints as reference model



Can SLM Works w/o High-Quality Corpus?

- SLM also performs well in self-reference scenarios
- With information entropy scoring function, SLM achieves better results
  - Higher information entropy indicates greater uncertainty of a token in its context

$$\mathcal{H}_{\text{RM}}(x_i) = -\sum_{k=1}^{V} P(t_k|x_{<i}) \log P(t_k|x_{<i})$$

Model	Score Function	Data	Uniq. Toks	Train Toks	GSM8K	MATH	SVAMP	ASDiv	MAWPS	MQA	AVG
Tinyllama-CT (RM)	-	OWM	14B	15B	6.3	2.6	21.7	36.7	47.7	13.9	21.5
Tinyllama-SLM	$\mathcal{L}_{\text{RM}}$	OWM	14B	10.5B	6.7	4.6	23.3	40.0	54.5	14.3	23.9
Tinyllama-SLM	$\mathcal{H}_{\text{RM}}$	OWM	14B	10.5B	7.0	4.8	23.0	39.3	50.5	13.5	23.0
Tinyllama-SLM	$\mathcal{L}_{\text{RM}} \cap \mathcal{H}_{\text{RM}}$	OWM	14B	9B	7.1	5.0	23.5	41.2	53.8	18.0	24.8
Tinyllama-CT	-	PPile	55B	52B	8.0	6.6	23.8	41.0	54.7	14.2	24.7
Tinyllama-SLM	$\mathcal{L}_{\text{RM}} \cap \mathcal{H}_{\text{RM}}$	PPile	55B	36B	8.6	8.4	24.4	43.6	57.9	16.1	26.5

Take-Home Message

- Not all tokens are useful during language model (LM) pretraining
  - Some tokens are already learned or noisy, and training on them is a waste

- SLM enhances data efficiency in LM training through token selection
  - It selects tokens based on how much they help the model improve

- SLM is more efficient and works better
  - It needs fewer tokens but gives higher or comparable performance

Limitation & Discussion

- SLM has only been validated on 1B and 7B models with <100B tokens
  - Scalability to larger models and corpora remains an open question
- SLM needs many steps like reference model training and scoring
  - Real training efficiency may not always improve
- Instead of training a LM after scoring, why not just use the RM directly?
  - The RM is only used for scoring, but its performance is not shown in the result tables.
  - Including RM's result and analysis would help clarify whether SLM truly improves over it.
- SLM does not work for specific downstream tasks
  - Token selection is not directly based on downstream task performance

Reference

1. Brown, T. et al., Language Models are Few-shot Learners, NeurIPS 2020
2. Training Compute-Optimal Large Language Models, NeurIPS 2022
3. Raffel, C. et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, JMLR 2020
4. Penedo et al., The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only, NeurIPS 2023
5. Mindermann et al., Prioritized Training on Points that are Learnable, Worth Learning, and Not Yet Learnt, ICML 2022

Appendix: What Tokens are Selected with SLM?

- Visualization of token selection during the training on OpenWebMath
- Blue tokens are retained during actual pretraining
- The majority of tokens chosen by the SLM method are closely related to math

Token Selected Examples
• Process the student answer as a Math Object Formula, and break down its parse tree by its top-level operators. The idea is to create an array of the student's primitive factors, so say 3(x+1)(x+2)^2 gives (3,x+1,x+2). • Because we may want factoring over Z, checking the gcd of coefficients within each factor. • Pass each of these things to SAGE and ask if the nonconstant factors are reducible over Z or Q. Also ask if they are monic. These things at least we learned how to do at the Vancouver code camp. The end goal is to count the following forms as correct, possibly controlled by flags: n \{ }prod (factor)^power, where each factor is irreducible in Z[X], n in Z r \{ }prod (factor)^power, where each factor is irreducible and monic in Q[X], r in Q I suppose on the last one the monic requirement could be dropped with a flag. I have no plans to check that the form is fully condensed, e.g. forcing (x+1)^2 and rejecting (x+1)(1+x)
The equation of the path traversed by a projectile is called equation of trajectory. \n \n Suppose, the body reaches the point P after time (t) . \n \n Horizontal motion has no acceleration. Thus, using kinematic equation, horizontal distance covered will be – \n \n x = u \cos \theta t \n \n Or, \quad t = (\frac { x }{ u \cos \theta }) \n \n Vertical motion has constant acceleration (g) . Thus, distance covered will be – \n \n y = (u \sin \theta) t - \left (\frac {1}{2} \right) g t^2 \n \n = (u \sin \theta) \left (\frac {x}{u \cos \theta} \right) - \left (\frac {1}{2} \right) g \left (\frac {x}{u \cos \theta} \right)^2 \n \n = \left (\tan \theta \right) x - \left (\frac {g}{2 u^2 \cos^2 \theta} \right) x^2 \n \n In this equation, (\theta, \ u \ \text {and} \ g) are constants. Thus, \n \n 1. Term \left (\tan \theta \right) is a constant, let it is (p) \n 2. Term \left [\left (\frac {g}{2 u^2 \cos^2 \theta} \right) \right] is also a constant, let it is (q) \n \n So, \quad y = p x - q x^2 \n \n Therefore, (y \propto x^2) , which is a required condition of a parabola.