Scaling Laws with Vocabulary: Larger Models Deserve Larger Vocabularies

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff,
Zhongwei Wan, Ping Luo, Min Lin, Ngai Wong

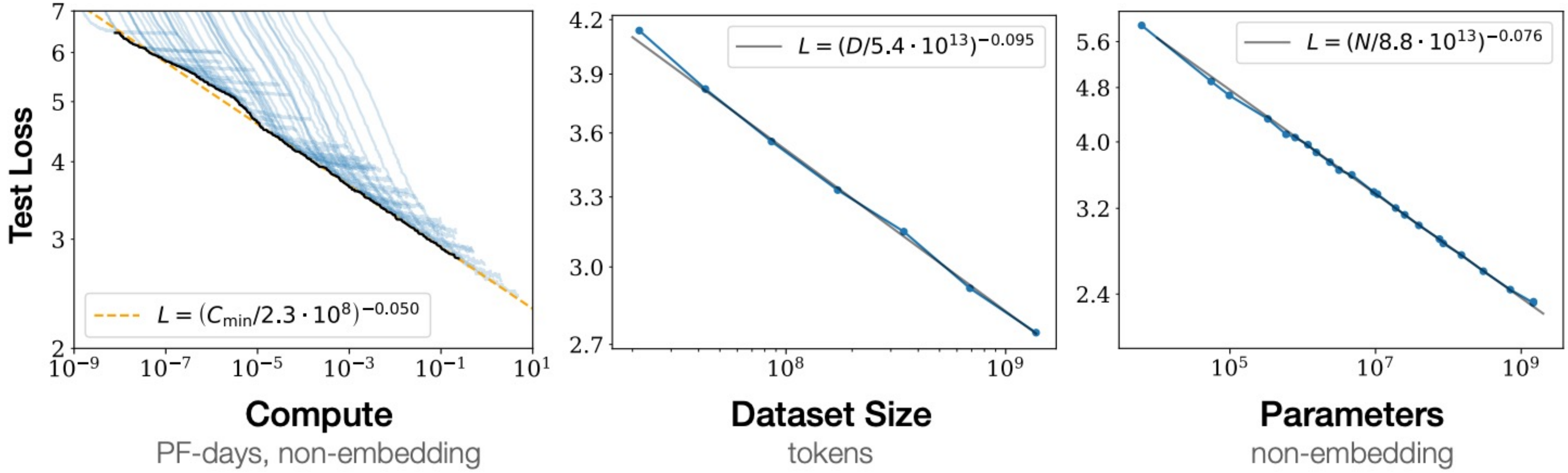
NeurIPS-2024

TL,DR: This paper introduces a framework substantiating ***a scaling law*** *that optimizes computational resources with* ***the consideration of vocabulary size, model parameters and training data jointly.***

Introduction

What is the Scaling laws?

Scaling laws describe how the performance of a neural network improves as its key attributes (e.g., number of parameters) increases.



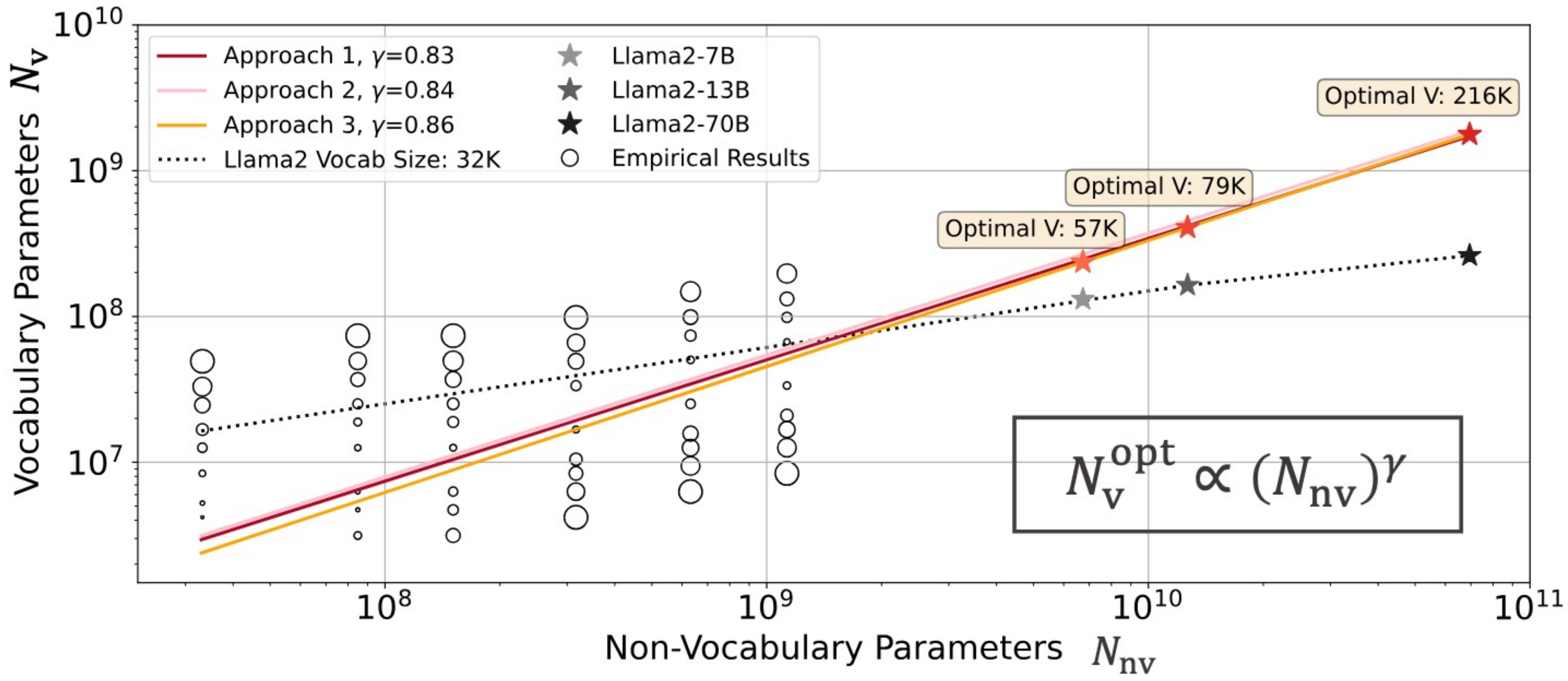
Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Introduction

Previous Scaling laws disregard the impact of the vocabulary size.
→ Substantial **variability** in the vocabulary size of current LLMs.

	Gemma-7B	**OLMo-7B**	**Chinchilla-70B**	**Llama2-70B**	**Llama3-70B**	**Falcon-180B**
Vocabulary Size	256K	50K	32K	32K	128K	65K
Model Parameters	7B	7B	70B	70B	70B	180B
Training Data	6T	2.5T	1.4T	2T	15T	3.5T

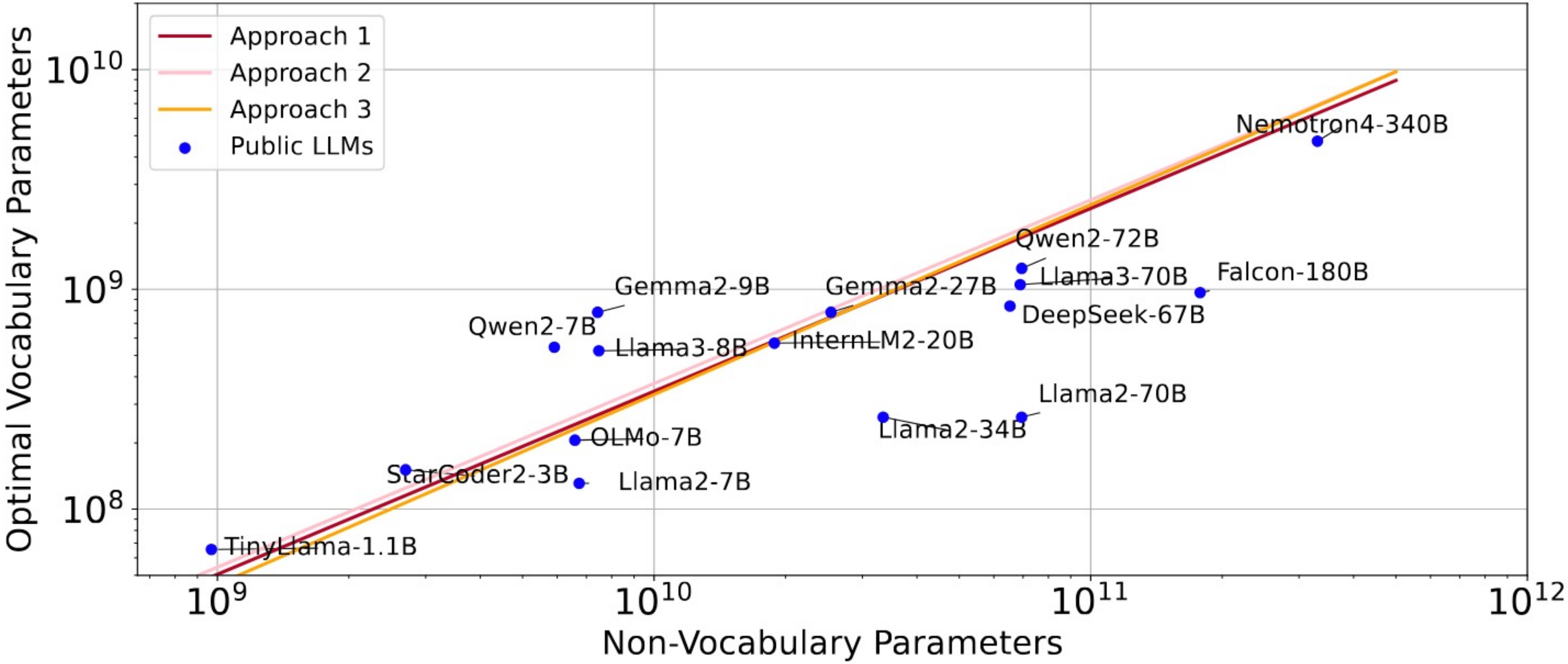
Introduction



The relationship between non-vocabulary parameters N_{nv} and the corresponding optimal vocabulary parameters N_v^{opt} follows a power law, where N_v^{opt} should be scaled slower than N_{nv} .

Introduction

Most existing LLMs have insufficient vocabulary parameters.



Vocabulary parameters of popular LLMs and the predicted optimal vocabulary parameters at a compute-optimal number of training tokens.

Preliminary

Preliminary for Scaling law:

$$(N_{opt}, D_{opt}) = \arg\min_{N,D} \mathcal{L}(N, D) \quad \text{s.t. FLOPs}(N, D) = C,$$

The goal is to optimally allocate this compute budget *C* to model parameters *N* and the number of training tokens *D, where the loss function is modeled as the language modeling loss*.

Preliminary

Our adaptation for Scaling law with vocabulary:

1. Attributes:

→Break the parameters into vocabulary parameters and non-vocabulary parameters

2. From training characters (H) to tokens (D) :

→We predict the number of tokens given the characters in the corpus and the vocabulary size

3. Vocabulary-insensitive loss:

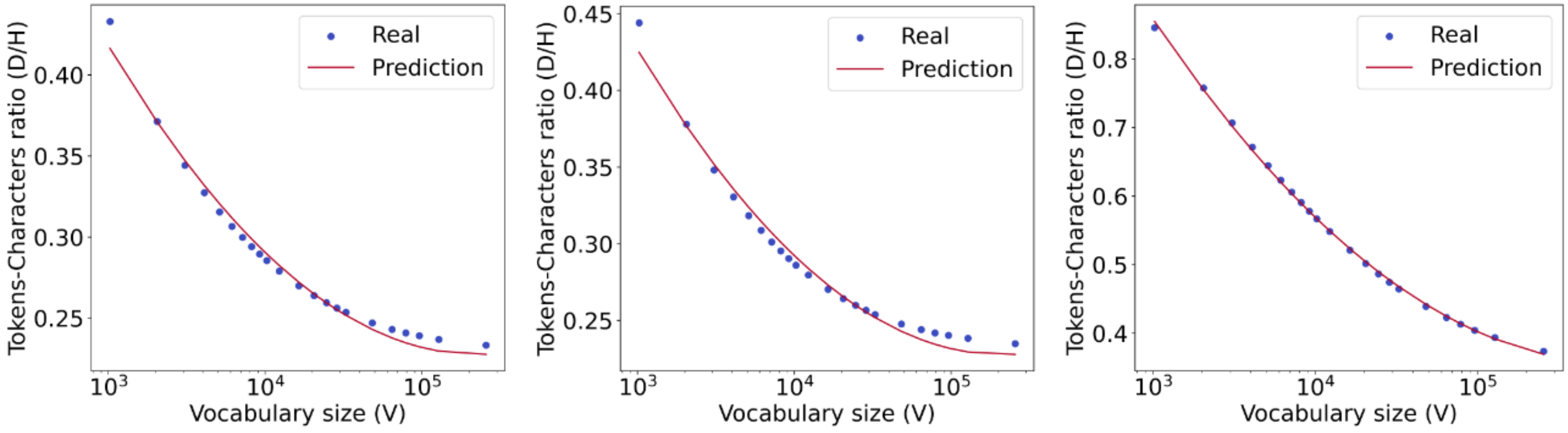
→Adjust the language modeling loss with a normalization

Analysis

How does the vocabulary size (V) affect the performance of language models?

From training characters (H) to tokens (D) : The number of tokens D divided from the sentences decreases as the V getting larger. We model it by a designed function f(V)=D/H.

$$f(V) = a\log^2(V) + b\log(V) + c$$



(a) BPE tokenizer (b) Unigram tokenizer (c) Word-based tokenizer

Analysis

How does fairly evaluate the language models with different vocabulary size?

We design the unigram-normalized language model loss as

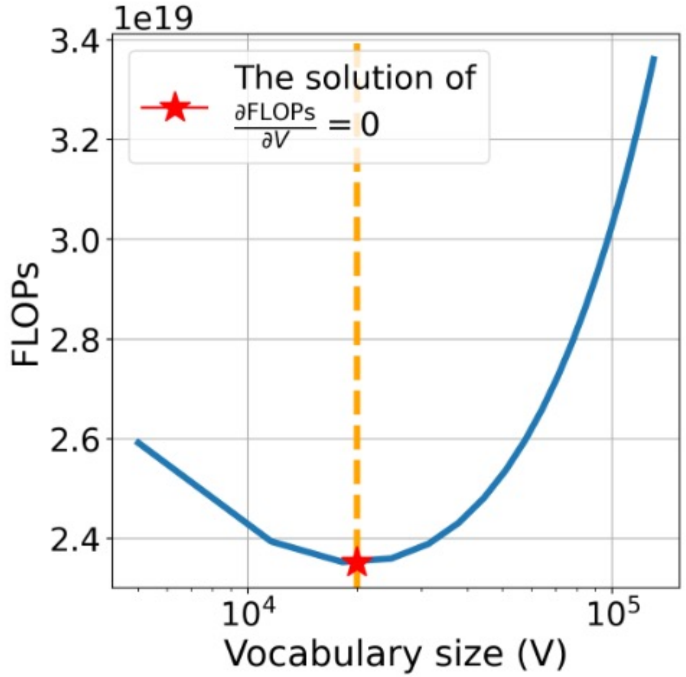
$$\mathcal{L}_u = -\frac{1}{T} \sum_{i=1}^{T} \log \frac{p(w_i | w_{1:i-1}, V)}{p(w_i | V)},$$

where $p(w_i|V)$ is the frequency of word w_i in the tokenized corpus, given the tokenizer with vocabulary size V. The loss is basically equivalent to the per-character language model loss normalized by the frequency of each character.

Analysis

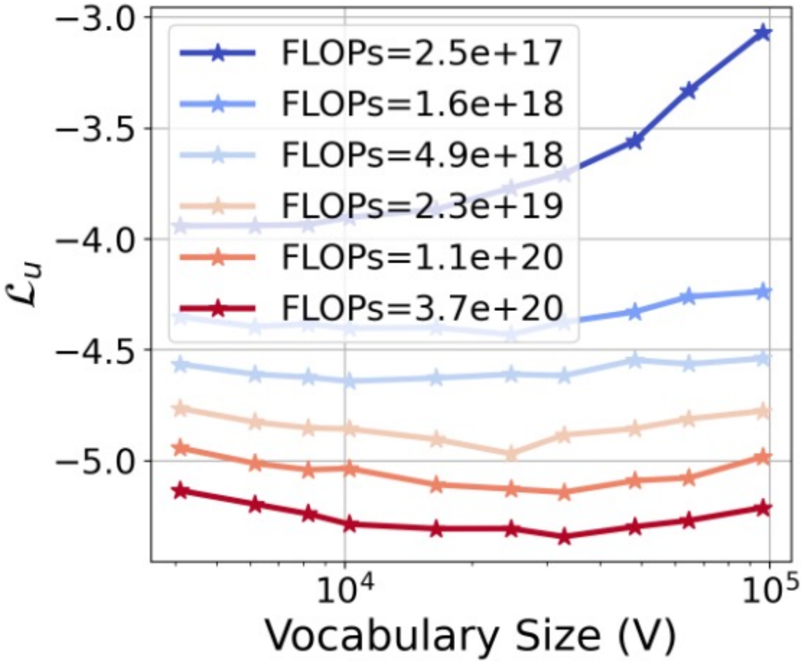
Why the optimal vocabulary size is bounded by compute?

The perspective of
fixed normalized loss



There exists an optimal vocabulary size that minimizes FLOPs.

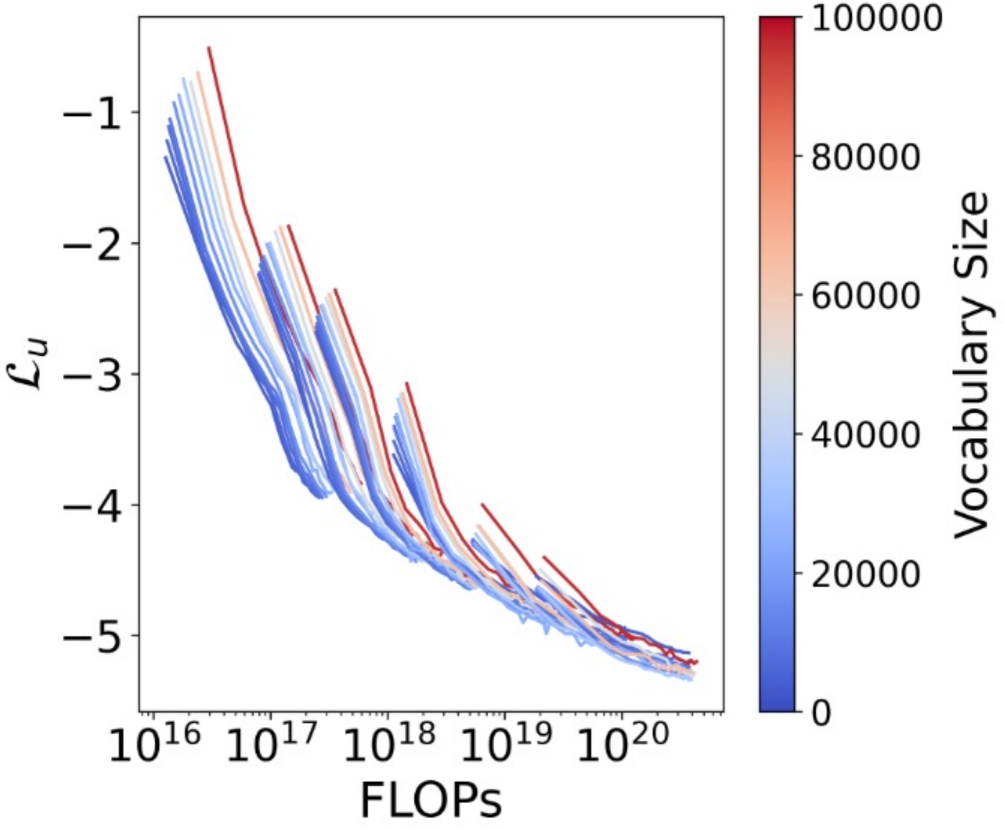
The perspective of
fixed FLOPs budge



For each FLOPs budget there exists an optimal vocabulary size that minimizes loss.

Estimating the optimal vocabulary size

Approach 1: Estimating power laws via IsoFLOPs

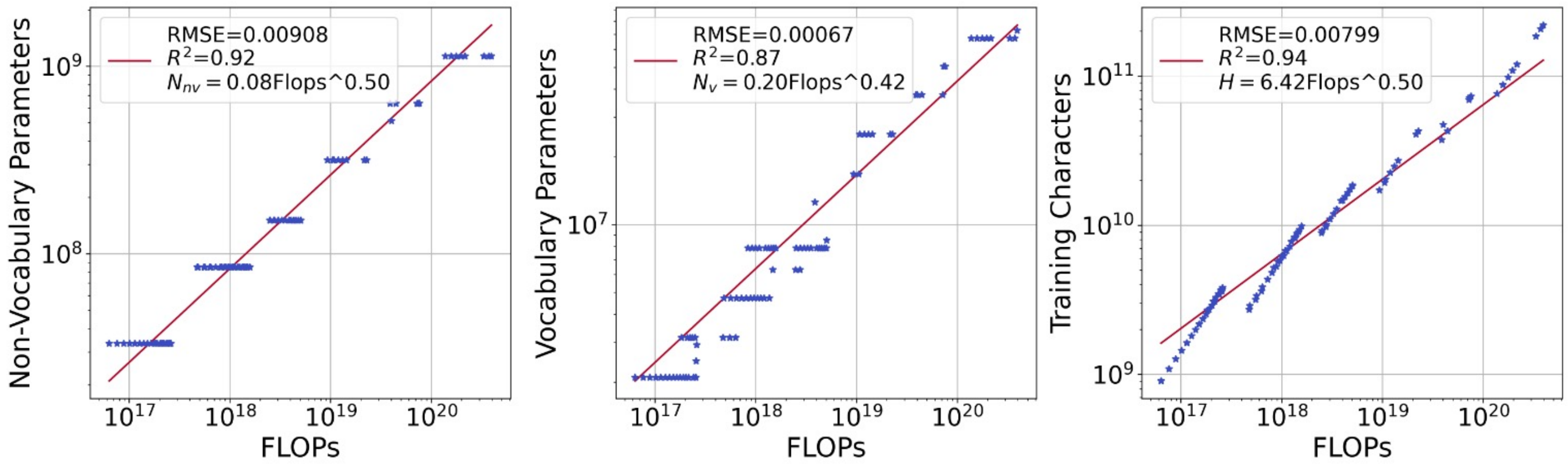


We define 6 groups of models with non-vocabulary parameters ranging from 33M to 1.13B.

Within each group, all models use the same FLOPs, and we solely vary the vocabulary size from 4K to 96K.

Estimating the optimal vocabulary size

Approach 1: IsoFLOPs with varying vocabulary sizes



1. LLMs are data-hungry
2. Vocabulary parameters scale in a power-law relation with FLOPs
3. Vocabulary parameters should be scaled slower than non-vocabulary parameters

Estimating the optimal vocabulary size

Approach 2: Derivative-based fast estimation

We aim to find the minimum FLOPs to achieve a certain loss through optimal allocation of the optimal vocabulary size.

$$V^{opt} = \underset{V|\mathcal{L}_u(N_{nv},V,H)=\ell}{\arg\min} \text{FLOPs}(N_{nv}, V, H).$$

By computing the minimum point of FLOPs with respect to V via derivative and find the zero solution of:

$$\frac{\partial C}{\partial V} = 6H\left[(N_{\text{nv}} + Vd)\frac{2a\log(V)+b}{V} + \left[a(\log(V))^2 + b\log(V) + c\right]d\right],$$

Estimating the optimal vocabulary size

Approach 2: Derivative-based fast estimation

We obtain a set of derivative-optimal vocabulary parameters N_v for different non-vocabulary parameters N_{nv} represented as $\{(N^i_{\mathrm{nv}}, N^i_{\mathrm{v}}) | i = 1, \cdots, n\}$

We then fit the relationship between N_{nv} and N_v using the power-law function $N_{\mathrm{v}} \propto N^{\gamma}_{\mathrm{nv}}$ and then we experimentally search an optimal vocabulary parameter N^0_{v} given a small model N^0_{nv}

$$N^{\mathrm{opt}}_{\mathrm{v}} = N^0_{\mathrm{v}} * (\frac{N_{\mathrm{nv}}}{N^0_{\mathrm{nv}}})^{\gamma},$$

Estimating the optimal vocabulary size

Approach 3: Parametric fit of loss formula

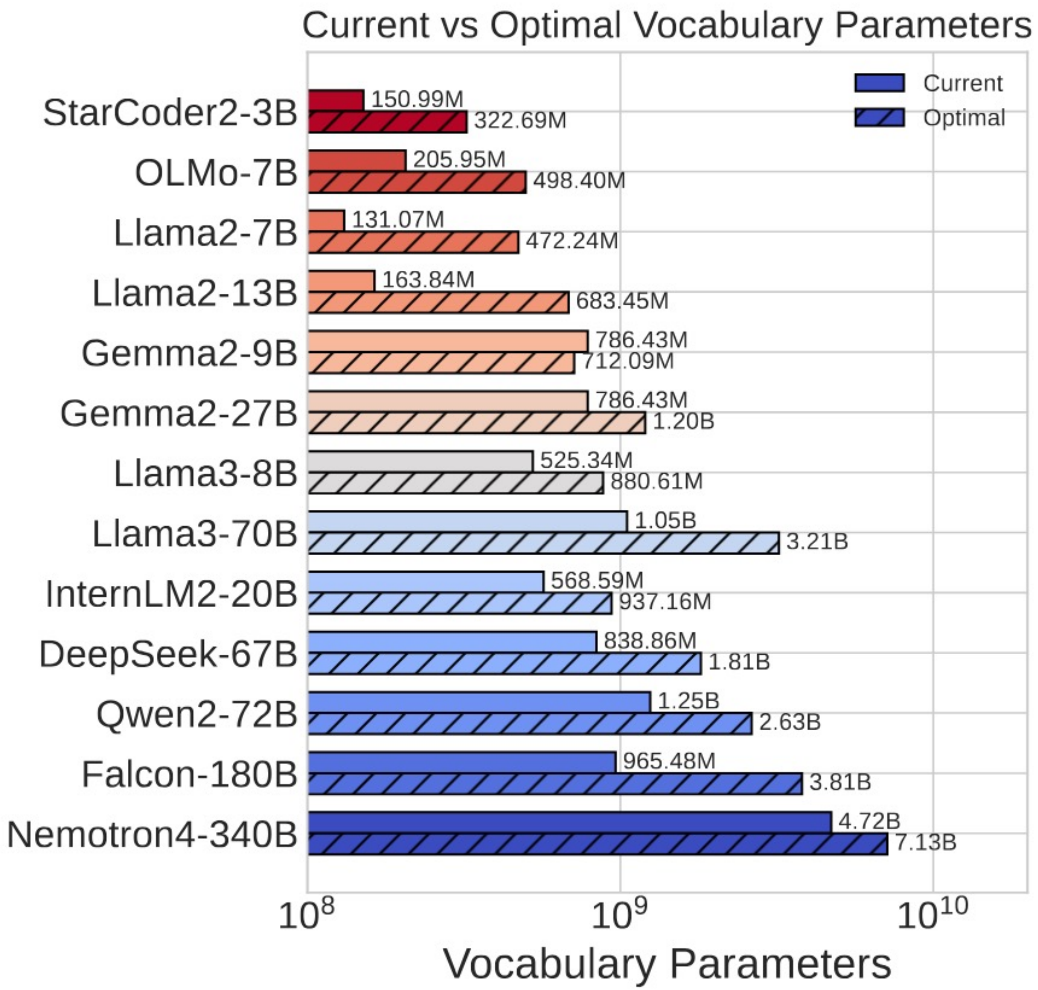
Finally, we directly predict the loss given the non-vocabulary parameter N_{nv}, vocabulary parameter V and the amount of training characters $H = Df(V)$. We design the vocabulary-dependent loss formula as

$$\mathcal{L}_u = -E + \frac{A_1}{N_{\mathrm{nv}}^{\alpha_1}} + \frac{A_2}{N_{\mathrm{v}}^{\alpha_2}} + \frac{B}{D^{\beta}},$$

where $E, A_1, A_2, B, \alpha_1, \alpha_2, \beta$ are learned parameters.

Estimating the optimal vocabulary size

Approach 3: Parametric fit of loss formula



Vocabulary parameters of popular LLMs and predicted optimal vocabulary parameters at their reported number of training tokens.

Discussions

Predicting allocations for larger models

N_{nv}	$N_{\mathrm{v}}^{\mathrm{opt}}$**-App1**	$N_{\mathrm{v}}^{\mathrm{opt}}$**-App2**	$N_{\mathrm{v}}^{\mathrm{opt}}$**-App3**	**Dim.**	V^{opt}**-App1**	V^{opt}**-App2**	V^{opt}**-App3**	**FLOPs Budget**
3B	0.1B	0.1B	0.1B	3200	39K	43K	37K	$1.3e21$
7B	0.3B	0.3B	0.2B	4096	62K	67K	60K	$7.1e21$
13B	0.4B	0.5B	0.4B	5120	83K	91K	81K	$2.4e22$
30B	0.9B	0.9B	0.9B	6048	142K	154K	142K	$1.3e23$
70B	1.7B	1.9B	1.8B	8192	212K	231K	218K	$7.1e23$
130B	2.9B	3.2B	3.0B	12888	237K	258K	248K	$2.4e24$
300B	5.8B	6.4B	6.3B	16384	356K	389K	383K	$1.3e25$

The predictions from all proposed approaches align closely.

Discussions

Empirically proving our compute allocations:

	N_v	D	H	**ARC-C**	**ARC-E**	**Hellaswag**	**OBQA**	**WG**	**PIQA**	**BoolQ**	**Average**
FLOPs Budget 1.2e21 (Optimally-Allocated Training Data)											
V=32K	0.10B	67.3B	266.6B	28.5±1.3	49.2±1.0	47.5±0.5	31.6±2.1	50.4±1.4	71.4±1.1	56.4±0.9	47.9
V^{opt}=35K	0.11B	67.1B	268.2B	**29.1**±1.3	**50.6**±1.0	**48.1**±0.5	**31.6**±2.1	**51.9**±1.4	**71.4**±1.1	**57.1**±0.9	**48.5**

	N_v	D	H	**ARC-C**	**ARC-E**	**Hellaswag**	**OBQA**	**WG**	**PIQA**	**BoolQ**	**Average**
FLOPs Budget 2.8e20 (Insufficient Training Data, "Undertraining")											
V=32K	0.10B	15.7B	62.2B	23.6±1.2	40.8±1.0	34.4±0.5	**29.0**±2.0	49.7±1.4	64.9±1.1	59.8±0.9	43.2
V^{opt}=24K	0.08B	15.8B	60.8B	**24.2**±1.3	**42.2**±1.0	**36.0**±0.5	28.6±2.0	**50.0**±1.4	**64.9**±1.1	**61.5**±0.9	**43.9**
FLOPs Budget 2.3e21 (Overly Sufficient Training Data, "Overtraining")											
V=32K	0.10B	128.5B	509.1B	29.1±1.3	53.5±1.0	53.0±0.5	33.0±2.1	52.0±1.4	72.0±1.1	59.5±0.9	50.3
V^{opt}=43K	0.14B	127.0B	517.5B	**32.0**±1.4	**54.7**±1.0	**54.1**±0.5	**33.0**±2.1	**52.8**±1.4	**72.6**±1.0	**61.9**±0.9	**51.6**

By increasing the vocabulary size from the conventional 32K to 43K, we improve performance on ARC-Challenge from 29.1 to 32.0 with the same 2.3e21 FLOPs

Conclusions

1. We investigate the impact of the vocabulary size when scaling language models.

2. We analyze and verify that there exists an optimal vocabulary size for a given FLOP budget.

3. We develop three approaches to predict the optimal vocabulary size.

Q&A

Thank you for listening!