Training Trajectories of Language Models Across Scales

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru,
Danqi Chen, Luke Zettlemoyer, Ves Stoyanov

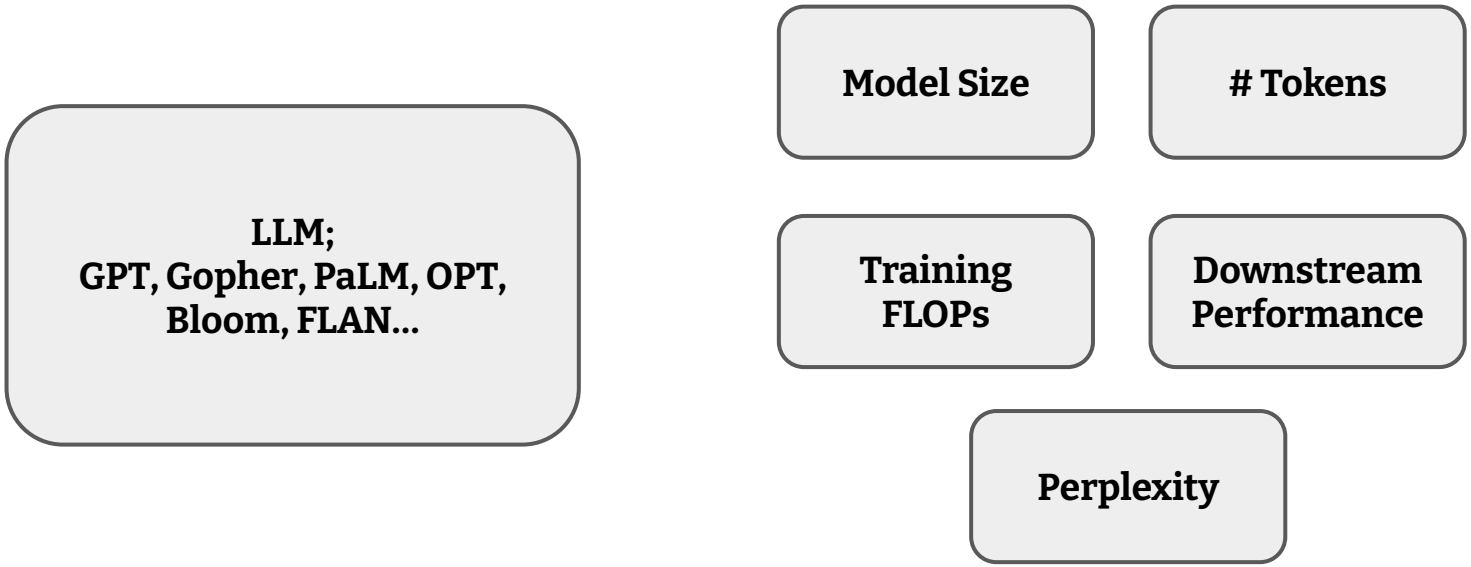
mengzhou@princeton.edu

@xiamengzhou

Mengzhou Xia

- Third year PhD student at Princeton
- Advised by Prof. Danqi Chen
- Interested in efficient methods in LLMs
  - Less training data
  - Less compute
  - Scale down
- Master at CMU, advised by Prof. Graham Neubig
- Interned at MSR and Meta
- 2022 Bloomberg Fellowship recipient

When given a language model, we consider ..



How do models of different scales learn during pre-training?

Training trajectories!

OPT Models

- Autoregressive pretrained language models of various sizes (125m, 1.3b, 6.7b, 13b, 30b, 175b)
  - **Data:** all models are trained with 300B tokens (180B corpora, around 1.67 epochs)
  - **Other hyperparameters:** Note that different-sized models are trained with different numbers of steps, different LRs (not the main focus).

Model	#L	#H	d_{model}	LR	Batch
125M	12	12	768	$6.0e{-4}$	0.5M
350M	24	16	1024	$3.0e{-4}$	0.5M
1.3B	24	32	2048	$2.0e{-4}$	1M
2.7B	32	32	2560	$1.6e{-4}$	1M
6.7B	32	32	4096	$1.2e{-4}$	2M
13B	40	40	5120	$1.0e{-4}$	4M
30B	48	56	7168	$1.0e{-4}$	4M
66B	64	72	9216	$0.8e{-4}$	2M
175B	96	96	12288	$1.2e{-4}$	2M

- OPT checkpoints up to 13B
- Pythia checkpoints from EleutherAI

What we are looking at?

- Pre-training objective: token-level predictions of language distributions

$$p(x_t \mid x_1, x_2, \cdots, x_{t-1}) \begin{cases} p(\text{be} \mid \text{I want to}) \\ p(\text{doctor} \mid \text{I want to be}) \\ \bullet \ \bullet \ \bullet \end{cases}$$

- Generalization of pretraining: perplexity of generated sequences

$$p(\overline{x_1, x_2, \cdots, x_t})$$

- Generalization to downstream tasks: in-context learning

Accuracy

We consider all these metrics as model behaviors

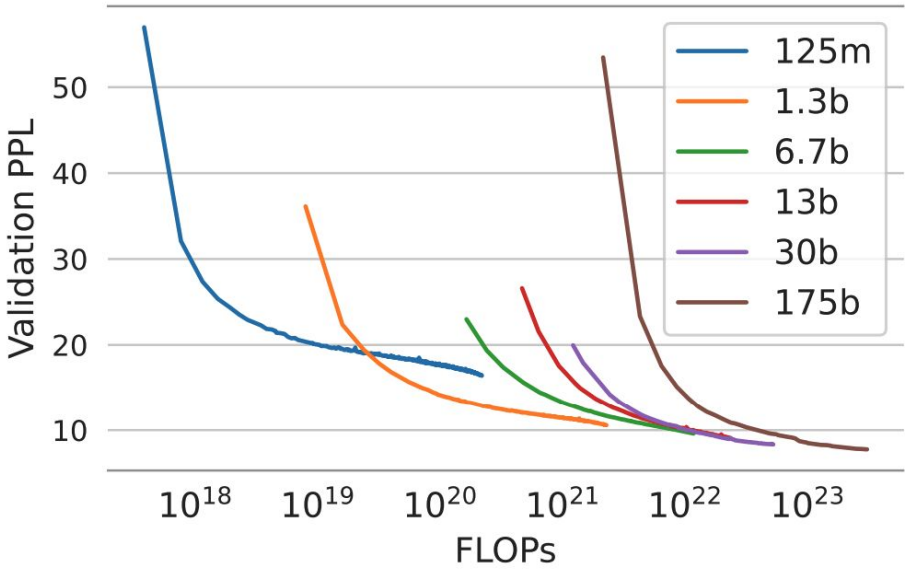
What property do model behaviors align with across scales?

 # Tokens?

FLOPs?

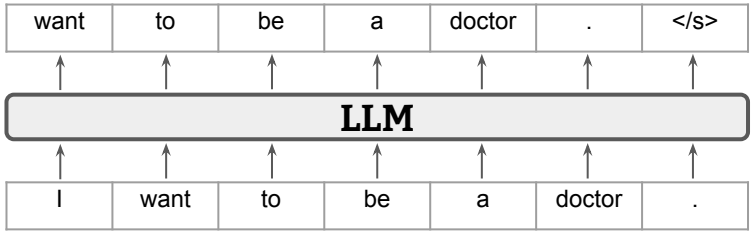
Perplexity?

Validation Perplexity



- The validation set of the pretraining task consists of 28 datasets covering a wide range of topics, e.g., wiki, stories, opensubtitle.
- General language modeling capabilities

What we are looking at?



- Pre-training objective: token-level predictions

$$p(x_t \mid x_1, x_2, \cdots, x_{t-1}) \begin{cases} p(\text{be} \mid \text{I want to}) \\ p(\text{doctor} \mid \text{I want to be}) \\ \bullet \ \bullet \ \bullet \end{cases}$$

- Generalization of pretraining: sequence-level generation

$$p(\overline{x_1, x_2, \cdots, x_t})$$

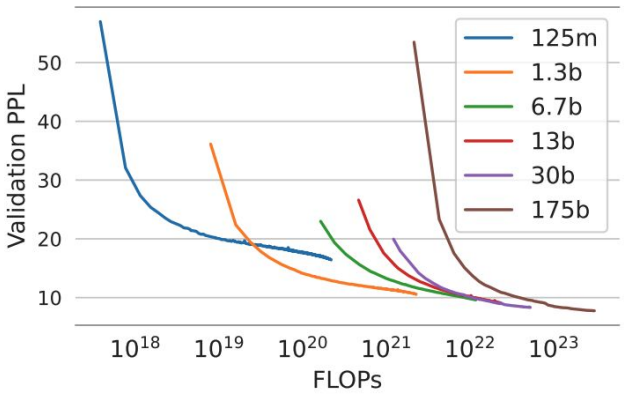
- Generalization to downstream tasks: in-context learning

Accuracy

We consider all these metrics as model behaviors

Token-level predictions on language distributions

OPT Models PPLs



Corpora PPL

$$p(x_1, x_2, \cdots, x_t)$$

Single next-word prediction PPL

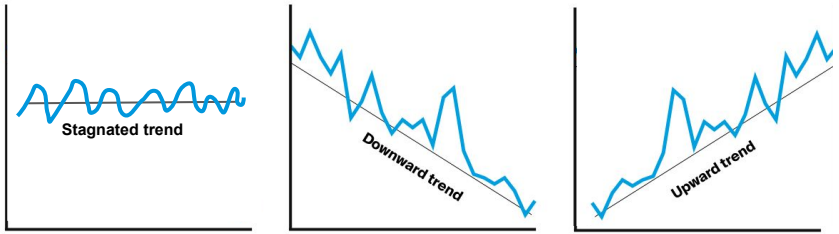
$$p(x_t \mid x_1, x_2, \cdots, x_{t-1}) \begin{cases} p(\text{be} \mid \text{I want to}) \\ p(\text{doctor} \mid \text{I want to be}) \\ \bullet \ \bullet \ \bullet \end{cases}$$

PPL of human corpora decreases as training progresses,
doe it mean all tokens' PPLs decrease?

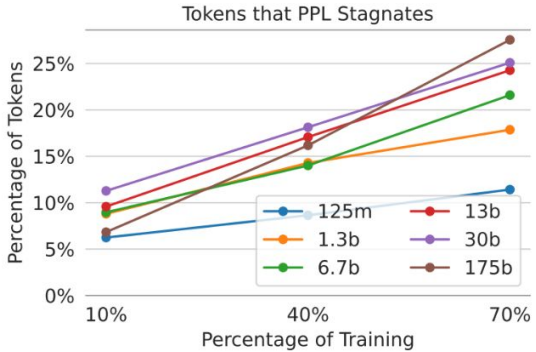
Categorize tokens based on its perplexity trend

A single PPL(x|c) is very unstable

- Given a perplexity series $\mathrm{PPL}_{t1}(x|c), \mathrm{PPL}_{t2}(x|c), \cdots, \mathrm{PPL}_{tn}(x|c)$
- We categorize each series to a
  - Stagnated trend (already learned)
  - Downward trend (still learning)
  - Upward trend (unlearning)
- By fitting the series with linear regression
- We cut first P% of training as it always shows a downward trend (P=10)

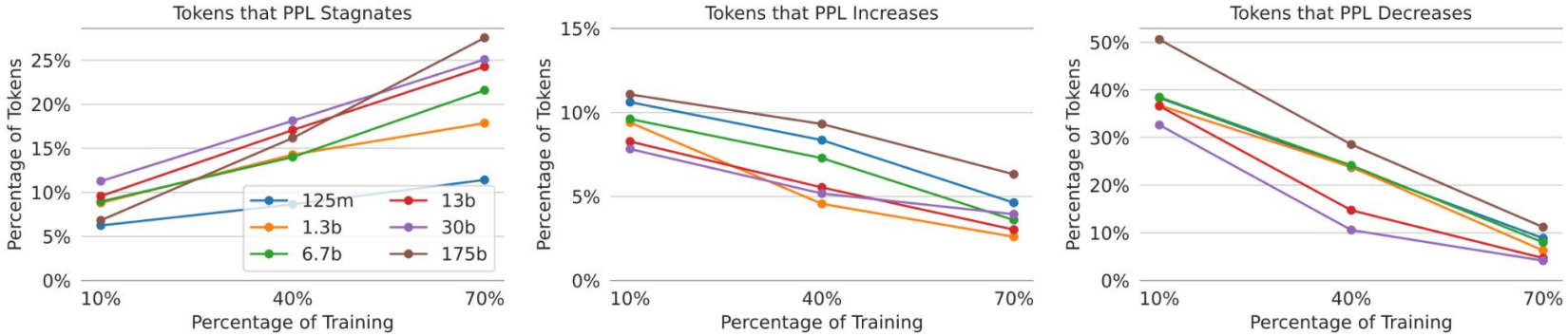


The percentage of these tokens across scales



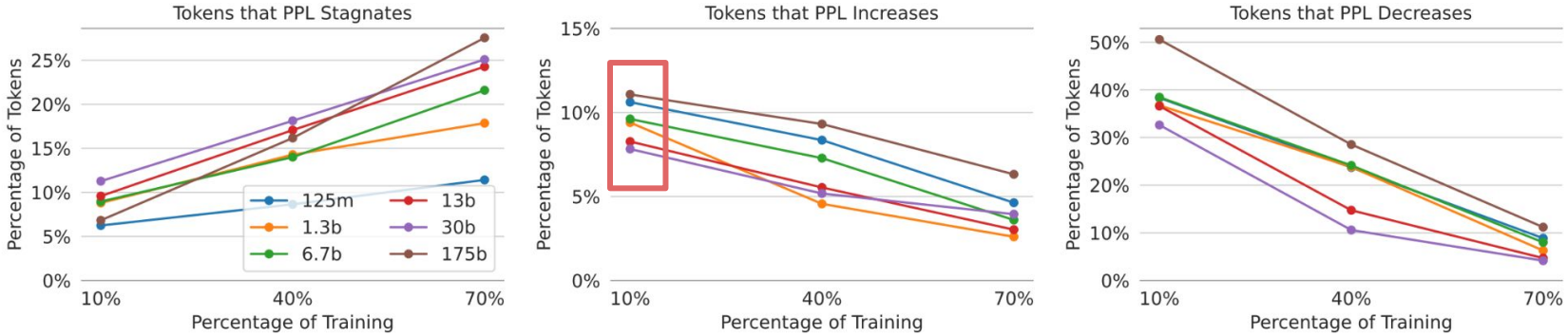
- More tokens stop learning (stagnated) as model trains

The percentage of these tokens across scales



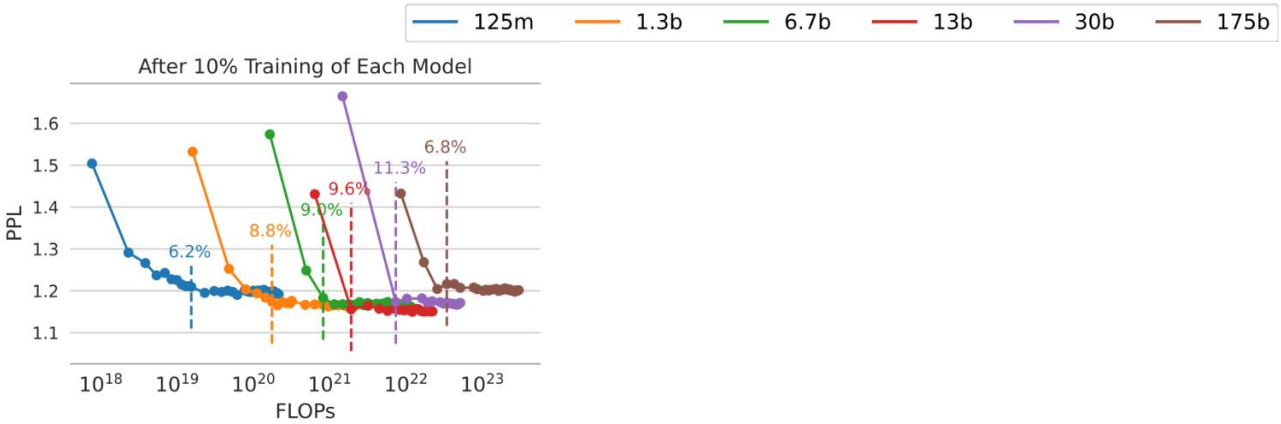
- More tokens stop learning (stagnated) as model trains
- Fewer tokens present a downward/upward trend as model trains

The percentage of these tokens across scales



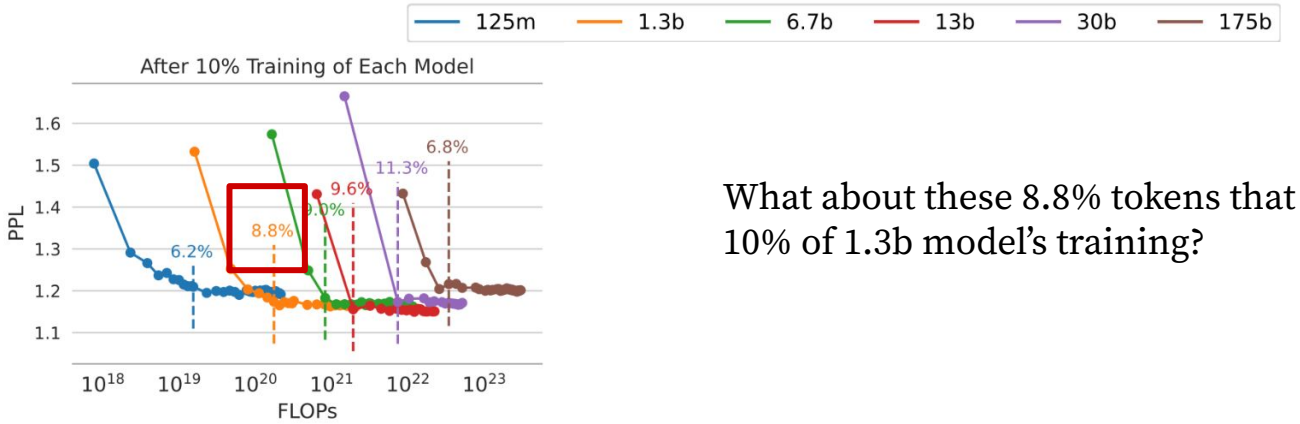
- More tokens stop learning (stagnated) as model trains
- Fewer tokens present a downward/upward trend as model trains
- 8-11% tokens present an upward trend after 10% of training
- Smaller models has fewer tokens that present a clear trend, e.g. 125M

Perplexity of stagnated tokens



- Yes, these tokens are truly stagnated in training

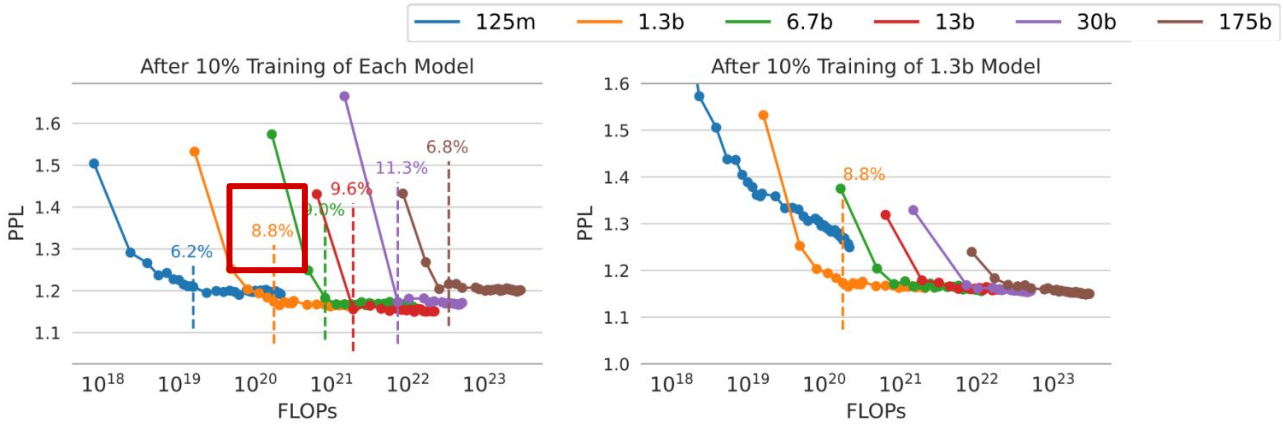
Perplexity of stagnated tokens



What about these 8.8% tokens that stagnated after 10% of 1.3b model's training?

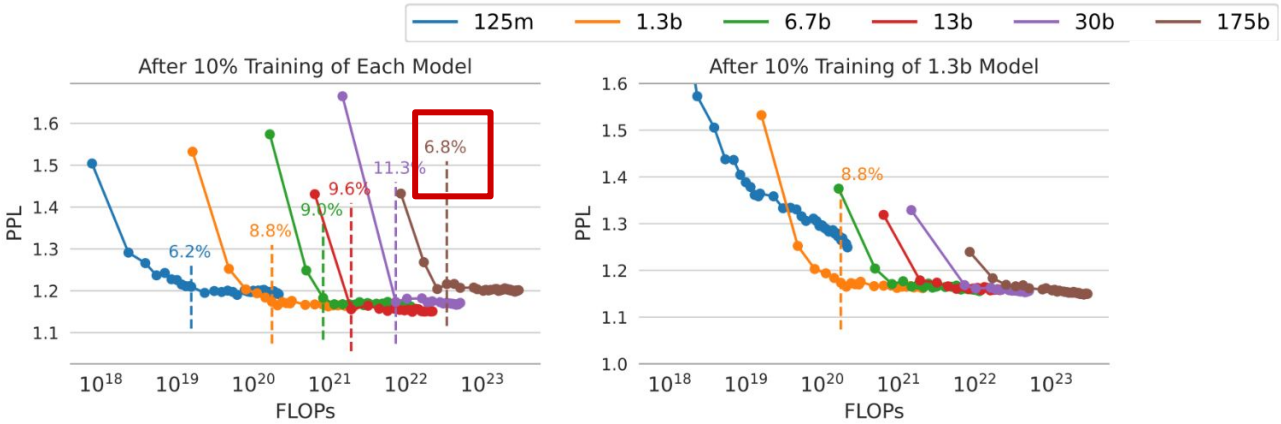
- Yes, these tokens are truly stagnated in training

Perplexity of stagnated tokens



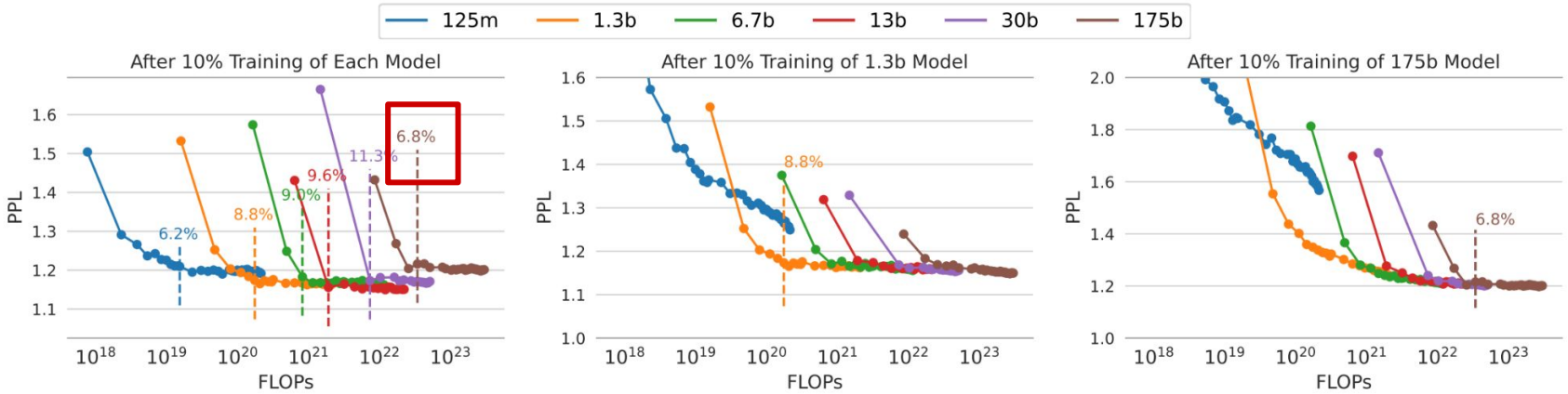
- Yes, these tokens are truly stagnated in training
- These 8.8% tokens eventually stagnated in larger models but not in smaller models

Perplexity of stagnated tokens



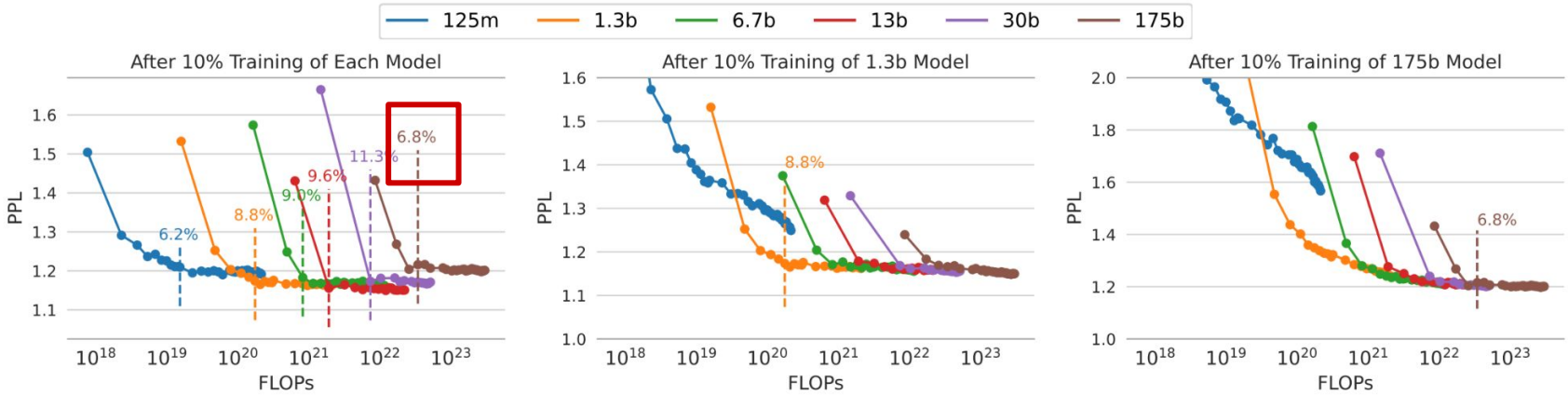
- Yes, these tokens are truly stagnated in training
- These 8.8% tokens eventually stagnated in larger models but not in smaller models

Perplexity of stagnated tokens



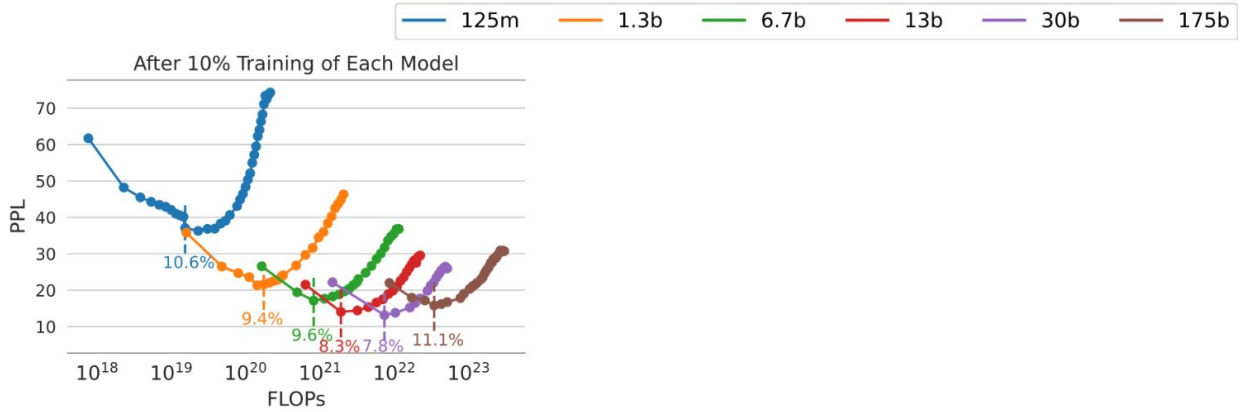
- Yes, these tokens are truly stagnated in training
- These 8.8% tokens eventually stagnated in larger models but not in smaller models
- These 6.8% tokens only stagnate in 175B model but not in smaller models

Perplexity of stagnated tokens



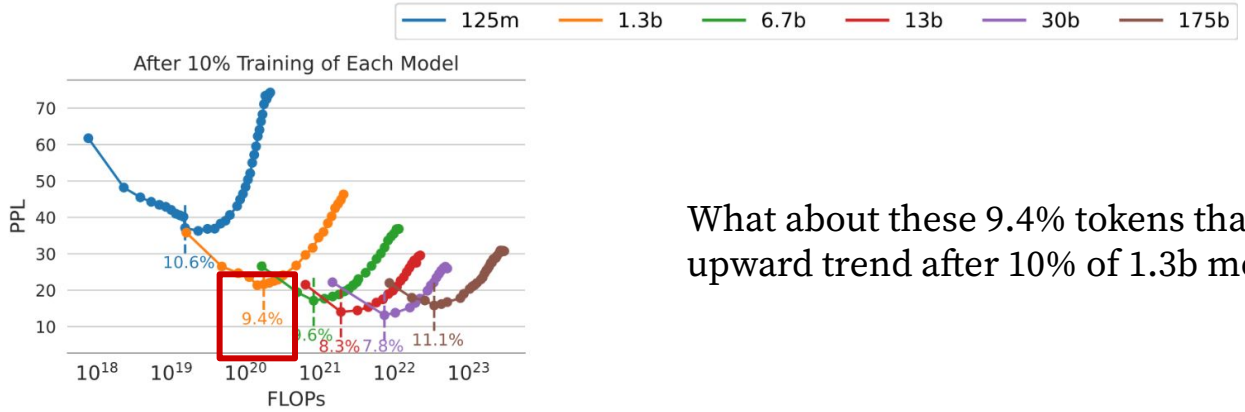
- Yes, these tokens are truly stagnated in training
- These 8.8% tokens eventually stagnated in larger models but not in smaller models
- These 6.8% tokens only stagnate in 175B model but not in smaller models
- The perplexity of these tokens do not align with FLOPs well

Perplexity of tokens with an upward trend



- Similarly, these tokens do present an upward trend

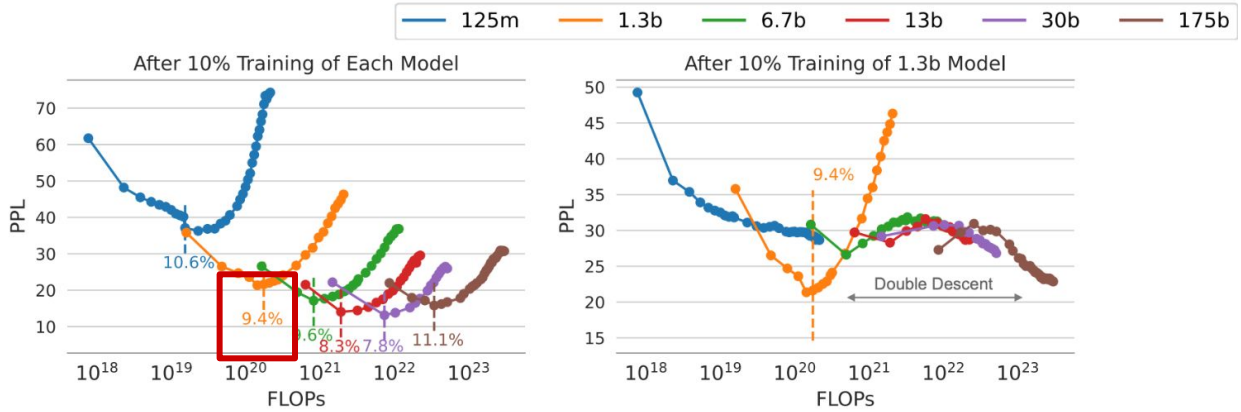
Perplexity of tokens with an upward trend



What about these 9.4% tokens that shows an upward trend after 10% of 1.3b model's training?

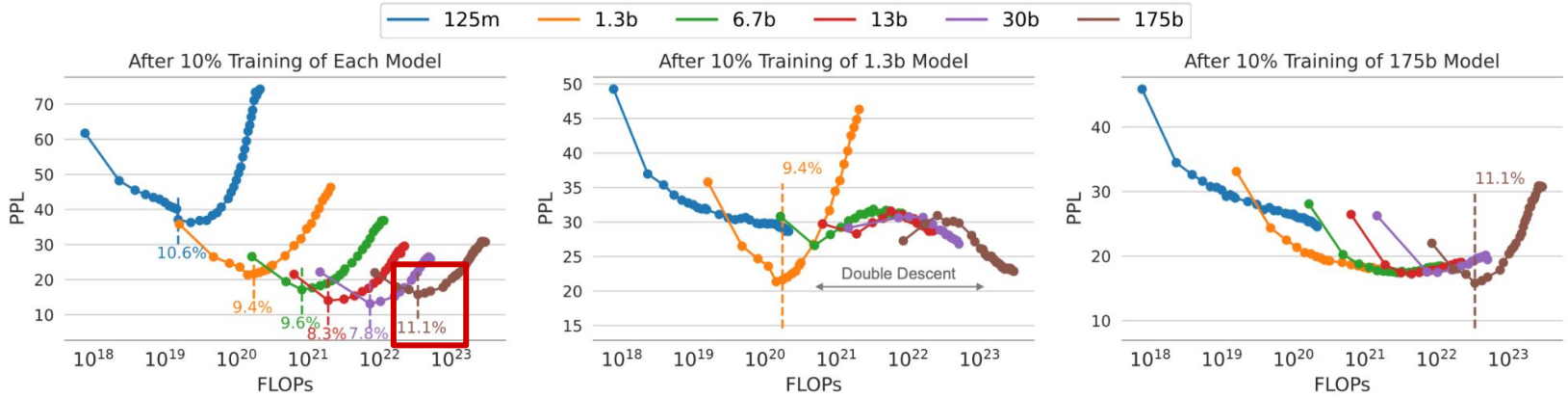
- Similarly, these tokens do present an upward trend

Perplexity of tokens with an upward trend



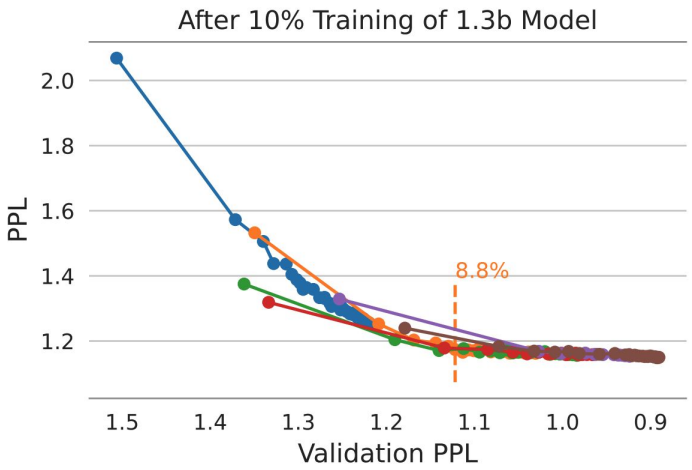
- Similarly, these tokens do present an upward trend
- It shows a downward trend in a smaller model, and a double-descent trend in larger models

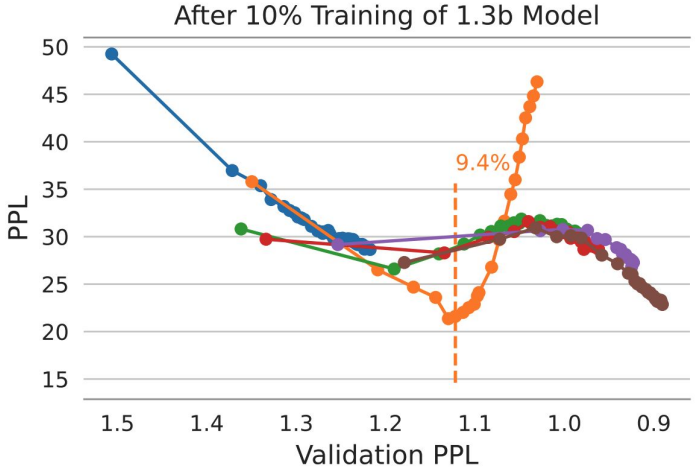
Perplexity of tokens with an upward trend



- Similarly, these tokens do present an upward trend
- It shows a downward trend in a smaller model, and a double-descent trend in larger models
- It shows a downward or downward/upward trend in smaller models

When plot against validation perplexity … (x: log scale)

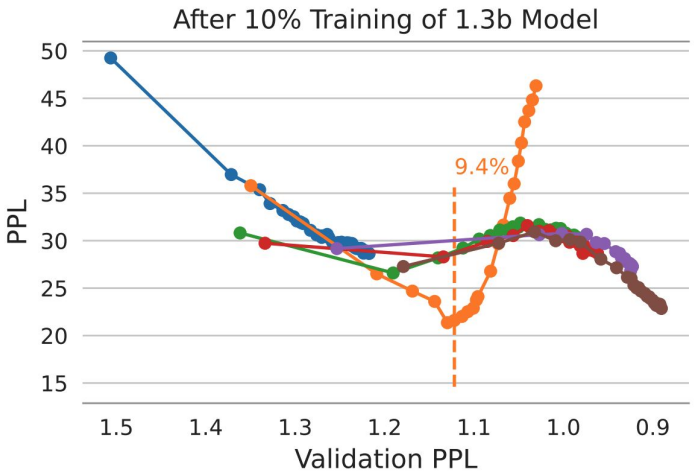


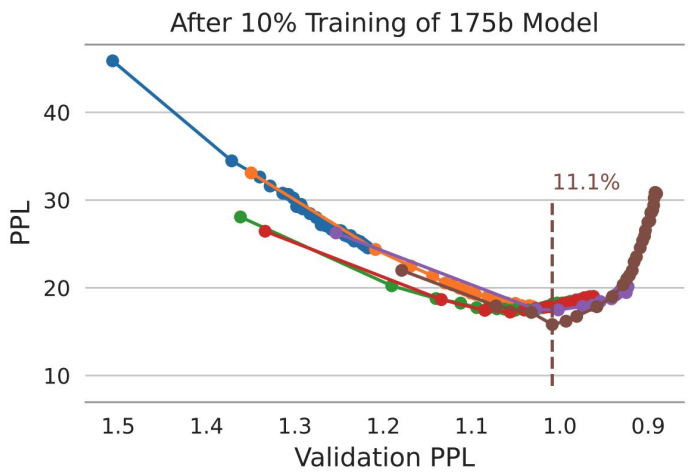


It aligns with validation PPL in a size agnostic way!

Except for 9.4% tokens in 1.3B model

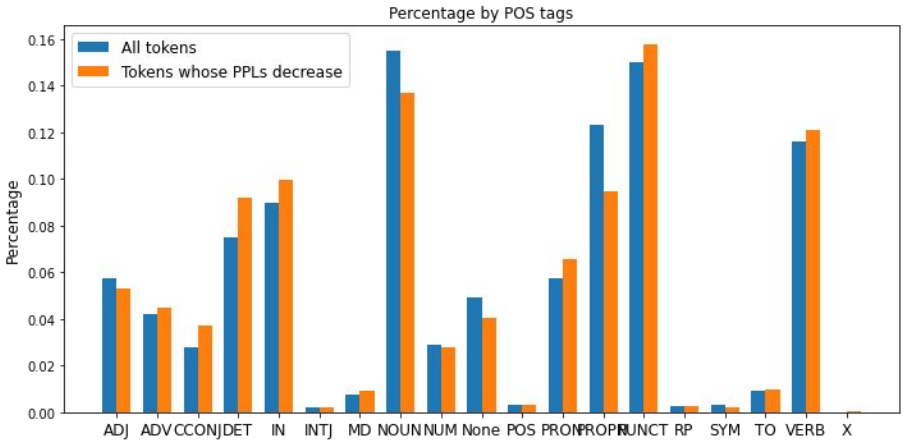
Perplexity Increase...



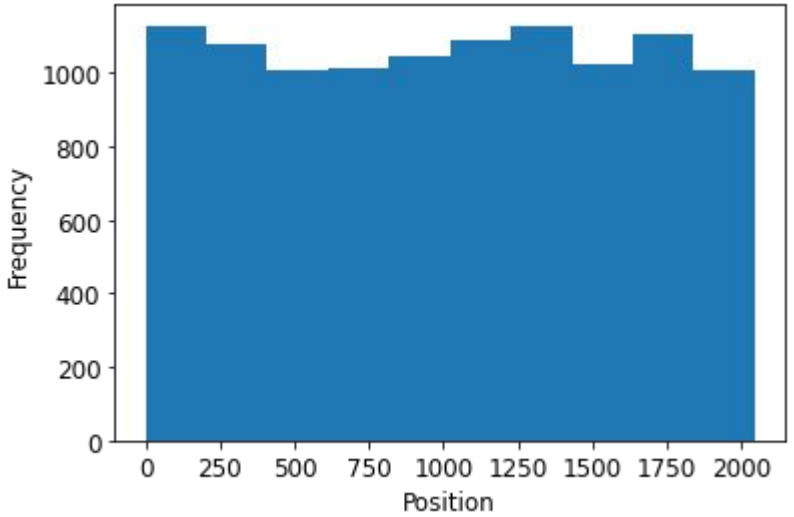


- These two sets of tokens do not overlap with each other and we didn't find any pattern
- It seems that perplexity increase happens for different sets of tokens throughout training
- Future work: Why does it happen? Is it really necessary for it to happen?

Further dissecting what these tokens are



They are not a particular type of POS.



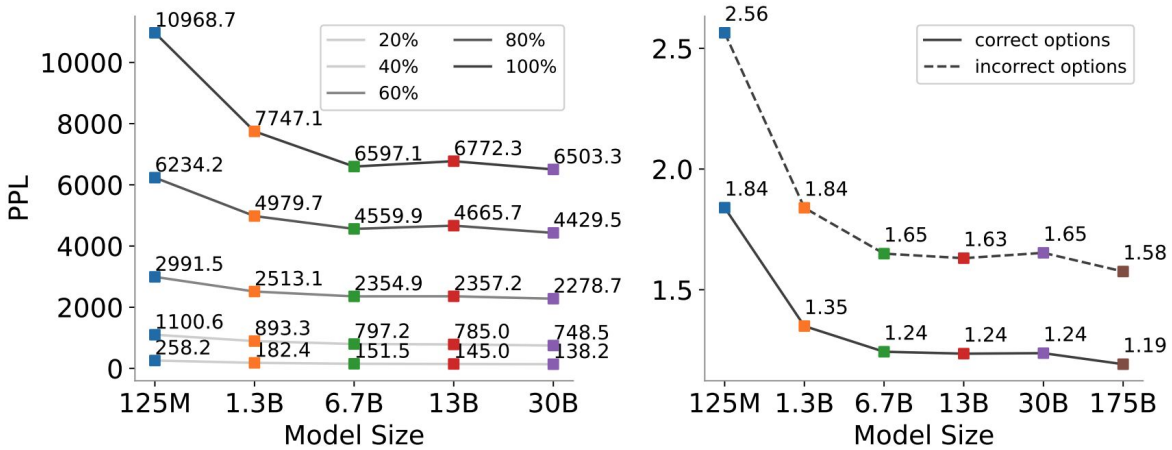
The occurrence is also agnostic of token positions.

It should a property of language (context, token), but beyond my comprehension.

Perplexity of Generated Sequences

Inverse scaling in language modeling

- A larger model has a lower perplexity on human texts, what do smaller models have lower perplexity on?
- Noise? In correct options in downstream tasks?



Both noisy data and incorrect options follow a normal scaling pattern.

We decode these sentences by contrasting two differently-sized models

$$p'_i = \lambda_1 \cdot p_s(x_i|x_{<i}) + \lambda_2 \cdot p_l(x_i|x_{<i})$$

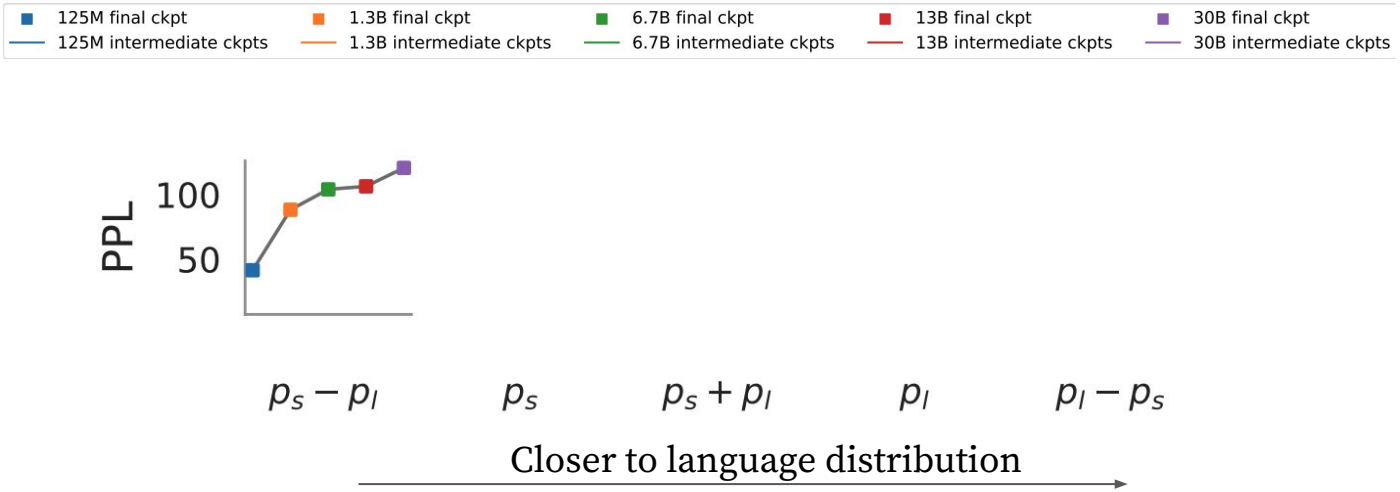
A small model's next token prediction

A large model's next token prediction

$\lambda_1 = 1, \lambda_2 = -1$ Maximizing small model's prob and minimizing large model's prob

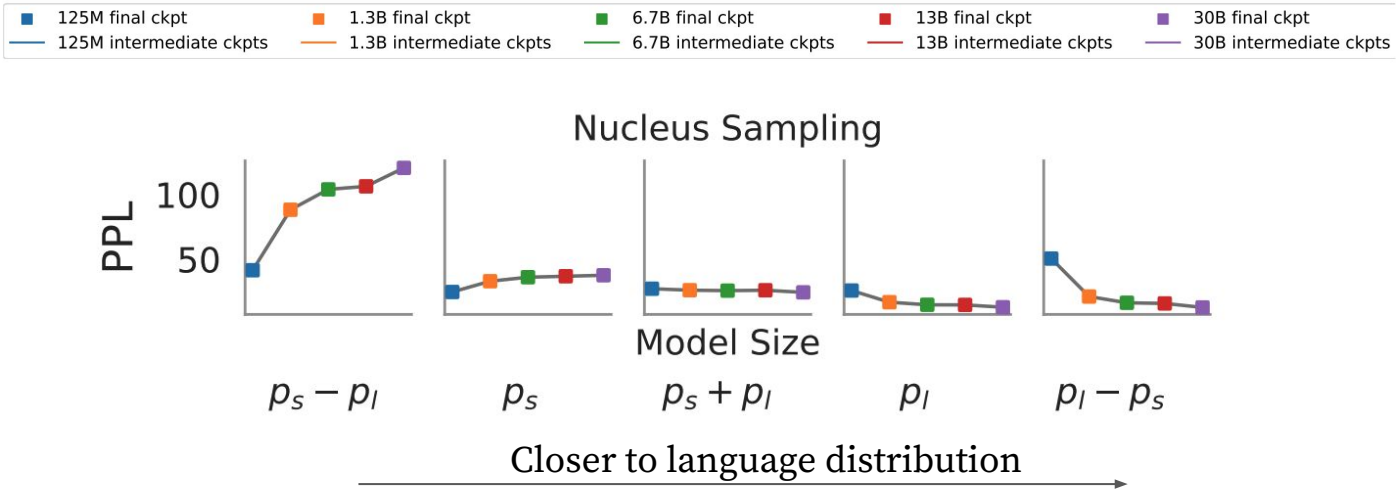
Similar to Li et al. 2022, but does not require any hyperparameter tuning.

Do the $p_s - p_l$ generations follow an inverse scaling trend?



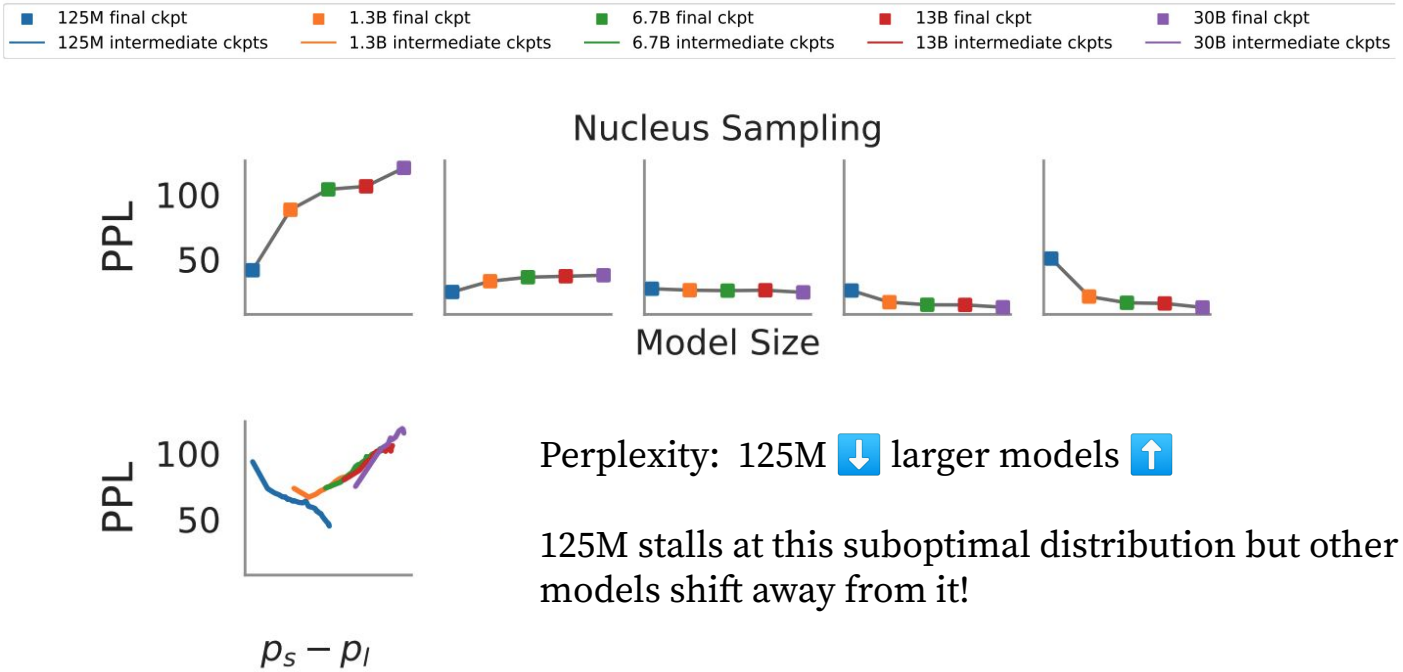
Yes, it does!

Do the $p_s - p_l$ generations follow an inverse scaling trend?



Other generations show a more flat trend or downward trend

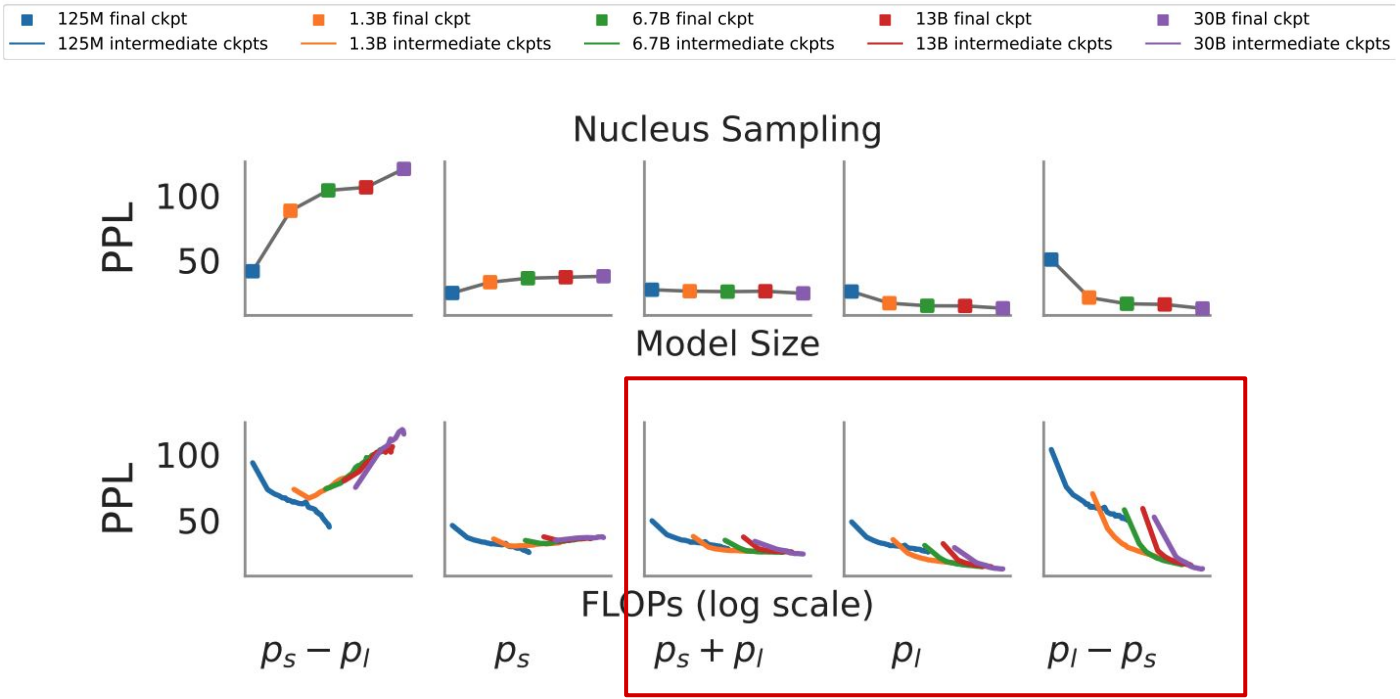
What are the trajectories like for these generations?



Perplexity: 125M ⬇️ larger models ⬆️

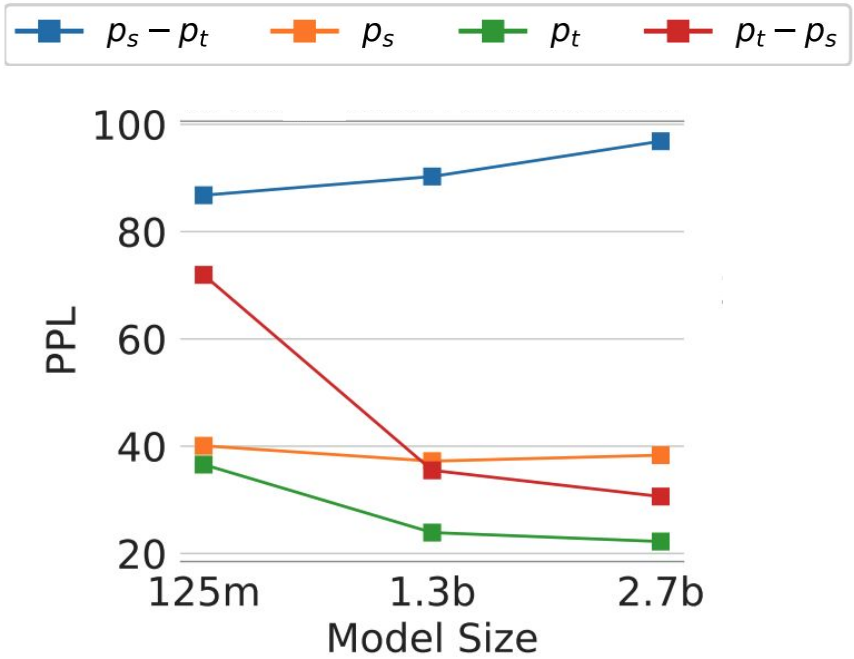
125M stalls at this suboptimal distribution but other models shift away from it!

What are the trajectories like for these generations?



Downward trend!

Similar trend with GPT-NEO



Distribution shift happens systematically!

What are the generated sequences like?

Dist.	Greedy Search	Nucleus Sampling
	Fortunately, the day wasn't all ...	***Fortunately, the day wasn't all ...***
$p_s - p_l$	that great. The sun was setting and the sun was falling. I went to bed and woke my husband, who was asleep in his bed, to find that I was still asleep in the middle of the night with him. He was still awake when we	that good when the computer said doom and gloom about me. Sure enough, because of our stubborn attempt at terrorizing him via cyberbackup (which relied heavily on computer traffic management (VCMD) to ensure my identity), I was able fix my old

The generations are grammatically correct, fluent, but contains hallucinations.

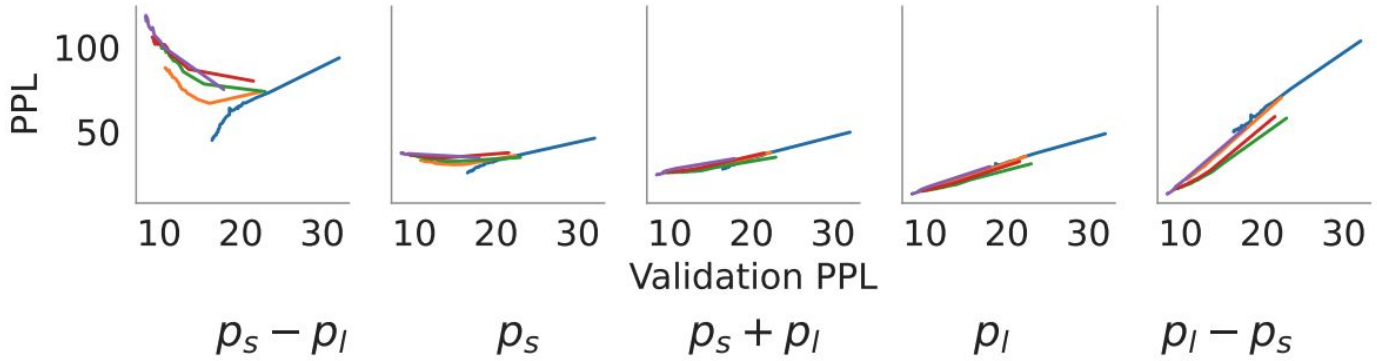
When p_s > p_t, the next tokens are grammatically correct but not correct in commonsense.

What are the generated sequences like?

Dist.	Greedy Search	Nucleus Sampling
$p_l - p_s$	bad news. The U.N.'s Intergovernmental Panel on Climate Change released a landmark study showing that we have 12 years to limit climate catastrophe. And a group of young activists filed a landmark climate lawsuit in federal district court, demanding that the government take	bad for Iowa fans. Tight end C. J. Fiedorowicz decided, for what has to be the millionth time now, to use Twitter as his own personal slogan board, and this time he decided to riff off the famous Bugs Bunny

Amazing generation quality when decoding with p_l-p_s, better than simply generate with p_l.

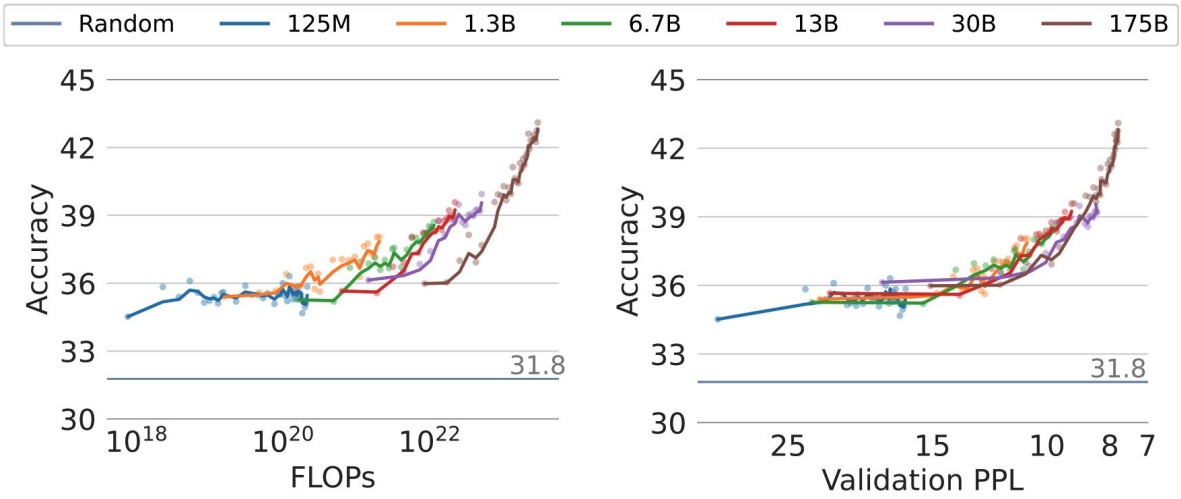
PPL of generated sequences vs. Validation PPL



It largely aligns with validation perplexity
except edge cases like p_s - p_l

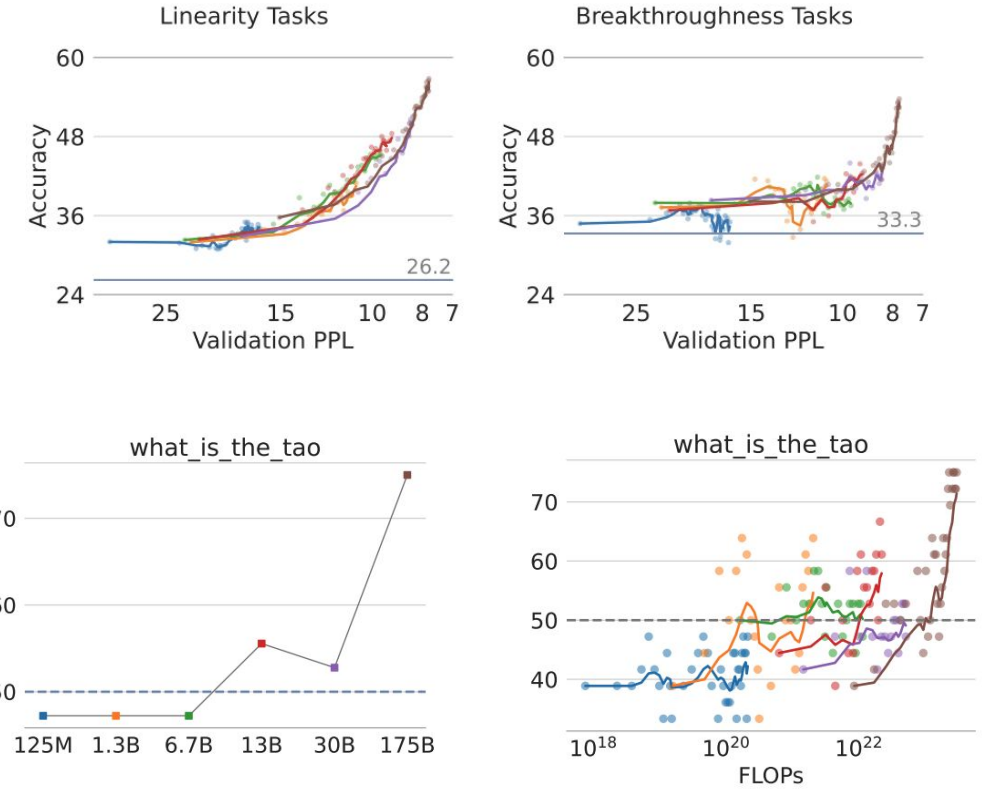
In-context Learning

ICL accuracy vs. FLOPs/Validation PPL



- ICL: 2-shot over 74 BigBench Tasks
- Accuracy aligns with Validation PPL regardless of model sizes

Emergent tasks are continuous on trajectories.



Conclusion

- When a large/small model achieve the same perplexity, their behavior/predictions are very similar, if not identical.
  - Perplexity of next token prediction of different trends
  - Perplexity of generated sequences
  - In-context learning
- It’s not the model size, or training flops that determine model behaviors, but the perplexity, and scaling up is a way to effectively reduce perplexity

Future Work

More model behaviors on trajectories

- More fine-grained analysis on specific tasks? COT?

Double Descent in Pre-training

- Why does it happen and does it have to happen?

Initialization for larger models

- Given two models of different sizes have the same perplexity, can we learn how to map the parameters of a small model to a large model for training efficiency?

Suboptimal distribution

- If we know the distribution of a small model is suboptimal, can we incorporate the information when training a large model to enhance training efficiency?