Training Trajectories of Language Models Across Scales

Mengzhou Xia[1], Mikel Artetxe[2], Chunting Zhou[2], Xi Victoria Lin[2],
Ramakanth Pasunuru[2], Danqi Chen[1], Luke Zettlemoyer[2], Ves Stoyanov[2]
[1]Princeton University [2]Meta AI
mengzhou@princeton.edu

Abstract

Scaling up language models has led to unprecedented performance gains, but little is understood about how the training dynamics change as models get larger. How do language models of different sizes learn during pre-training? Why do larger language models demonstrate more desirable behaviors? In this paper, we analyze the intermediate training checkpoints of differently sized OPT models (Zhang et al., 2022)—from 125M to 175B parameters—on next-token prediction, sequence-level generation and downstream tasks. We find that 1) at a given perplexity and independent of model sizes, a similar subset of training tokens see the most significant reduction in loss, with the rest stagnating or showing double-descent behavior (Nakkiran et al., 2020); 2) early in training, all models learn to reduce the perplexity of grammatical sequences that contain hallucinations, with small models halting at this suboptimal distribution and larger ones eventually learning to assign these sequences lower probabilities; and 3) perplexity is a strong predictor of in-context learning performance on 74 multiple-choice tasks from BIG-Bench, and this holds independently of the model size. Together, these results show that perplexity is more predictive of model behaviors than model size or training computation.[1]

1 Introduction

Scaling up language models has been shown to improve language modeling perplexity (Kaplan et al., 2020; Hernandez et al., 2022) as well as zero- or few-shot end task accuracies (Brown et al., 2020; Rae et al., 2021; Chowdhery et al., 2022; Zhang et al., 2022). However, relatively little is understood about why or how this happens. How do the training dynamics differ as models get larger? What do language models of different sizes learn during pre-training in terms of both generating texts and solving end tasks?

We attempt to make progress to answer these questions by studying the training trajectories of differently-sized OPT models (Zhang et al., 2022) through analyzing their intermediate checkpoints. In contrast to prior work, which studies the trajectories of small models with up to 300M parameters (Liu et al., 2021; Choshen et al., 2022; Blevins et al., 2022) or focuses on the language modeling objective alone (Kaplan et al., 2020; Hernandez et al., 2021, 2022), we are the first to comprehensively study the training trajectories of large-scale autoregressive language models with up to 175B parameters across a wide range of settings.

Repeatedly across training and different model scales, we analyze three aspects of model performance: (i) next-token prediction on subsets of tokens (ii) sequence-level generation and (iii) downstream task performance. We use perplexity, which is closely tied to language model evaluation, as the major metric throughout the study.

For **next-token prediction** (§3), we study the trajectory by categorizing each token's prediction as *stagnated*, *upward* or *downward* according to its perplexity trend as training progresses. We find each category comprising a significant number of tokens: while a significant number of tokens' perplexity stagnate, a subset of tokens with an increasing perplexity in smaller models exhibit a double-descent trend (Nakkiran et al., 2020) where perplexity increases and then decreases in larger models. These behaviors primarily emerge at a similar validation perplexity across model scales.

For **sequence-level generation** (§4), we study the distribution shift at a document level (50-500 tokens) by decoding sequences that small/large models favor more than the other. Human texts present expected scaling patterns in that they are best modeled by larger (or longer trained) models. However, to our surprise, large models are better at modeling

[1] Code is publicly available at https://github.com/xiamengzhou/training_trajectory_analysis.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13711–13738
July 9-14, 2023 ©2023 Association for Computational Linguistics

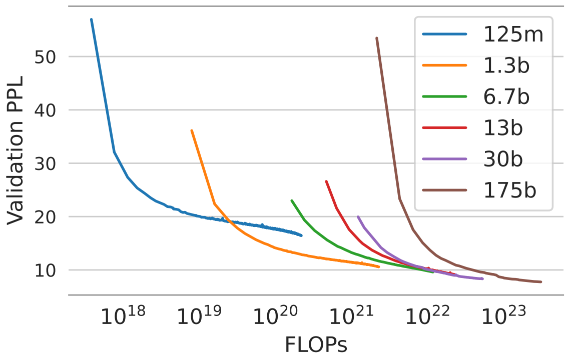


Figure 1: Validation perplexity (PPL) of OPT models against training FLOPs. Our work suggests that models with comparable perplexity levels during training exhibit similar predictions, regardless of their scales.

less human-like texts which contain synthetic noise and factually incorrect prompts. We propose an approach to decoding texts that small models favor more than large models from an interpolated distribution induced by combining signals from both models and find them grammatical but hallucinating.[2] All models go through a stage during training where the perplexity for such texts decreases; small models halt at this suboptimal distribution, while larger models escape it by eventually increasing the perplexity of these unnatural texts.

We further connect language modeling perplexity to **downstream tasks** (§5). By evaluating more than 70 multiple-choice tasks in BIG-Bench (Srivastava et al., 2022), we find that language modeling perplexity correlates well with few-shot in-context learning performance along the trajectory, regardless of model sizes. The gradual divergence of likelihood between correct and incorrect options leads to improvements in in-context learning.

Our work presents a comprehensive study of training trajectories of language models trained with similar procedures, e.g., OPT. We conclude that language models learn the same phenomena in the same order across different model sizes. The overall model perplexity is a composite measure of which language phenomena have been learned.

2 Experimental Settings

Models. Unless otherwise indicated, all of our experiments use OPT (Zhang et al., 2022), a collection of open-source autoregressive language models. OPT models serve as a good fit for this study due to their controlled pre-training procedures across all model sizes. In particular, all the models share the same tokenization and are trained on the same training data, covering a total of 300B tokens (180B unique). Note that different-sized models differ in batch sizes and total number of steps.[3] We collect intermediate checkpoints from the authors and perform evaluations of these checkpoints across six different sizes: 125M, 1.3B, 6.7B, 13B, 30B, and 175B.

Validation perplexity. Throughout this paper, we use *Validation Perplexity (Valid PPL)* to refer to the autoregressive language modeling perplexity measured on the entire validation set. We use the original OPT validation set, a held-out subset of the training corpus that covers a wide range of domains, such as books, news, and subtitles. We plot the trajectory of validation perplexity in Figure 1, which follows a similar power-law pattern observed in previous scaling work (Kaplan et al., 2020; Hoffmann et al., 2022).

Methodology. We aim to understand how models of different sizes behave throughout training as a function of computing (FLOPs)[4] and validation perplexity. Throughout the paper, we use different measurements to characterize model behavior and plot them against these two metrics.

3 Next-Token Prediction

Autoregressive language models are trained to predict the next token given a context. Figure 1 shows that validation perplexity, aggregated over all positions, gradually declines as training progresses. However, it is not clear if all token instances evolve similarly to the aggregated measurement. In this section, we study the trajectory of next-token predictions, dividing them into three categories—stagnated, upward trend, or downward trend—to understand how language models gradually learn new language phenomena.

3.1 Methodology

We evaluate intermediate checkpoints on a subset of validation data.[5] For each context-token pair (c, t), we obtain a series of perplexities $\text{PPL}_{m_1}(t \mid$

[2] Concurrent to our work, Li et al. (2022) propose a similar contrastive decoding approach for a different application. Refer to Appendix C.2 for more details.

[3] See Appendix A for more details of model checkpoints.

[4] We estimate the number of FLOPs of language models following Chowdhery et al. (2022).

[5] More dataset details are in Appendix B.1.

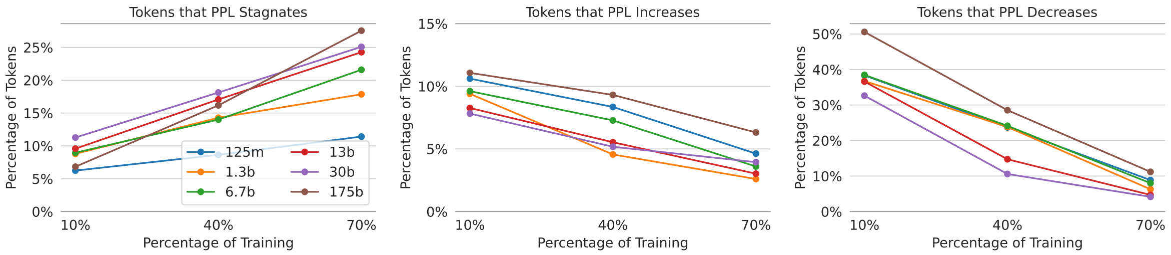


Figure 2: Percentage of predictions where perplexity stagnates (left), follows an upward trend (middle) and an downward trend (right). X-axis denotes that the trend is estimated after $p\%$ percentage of training.

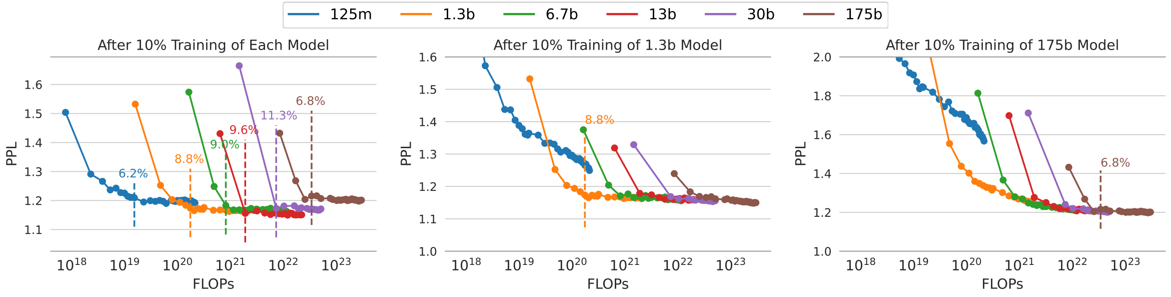


Figure 3: Perplexity of stagnated tokens. Left: different models are evaluated on different subsets of tokens selected after 10% of training of individual models (all showing a stagnated trend after 10%). Middle/right: all models are evaluated on the *same set of tokens*, selected after 10% of training of the 1.3B model and the 175B model respectively. The number next to the dashed line denotes the percentage of the selected tokens out of all tokens. Stagnated tokens selected by a smaller model (1.3B) are stagnated in larger models. Stagnated tokens selected by a larger model (175B) present a downward trend in perplexity in smaller models.

$c), \mathrm{PPL}_{m_2}(t \mid c), \ldots, \mathrm{PPL}_{m_n}(t \mid c)$ for checkpoints $m_1, m_2, \ldots, m_n$. We use linear regression to estimate the slope of a normalized series to roughly capture its trend. Starting from any intermediate checkpoint after $p\%$ of training (assuming that it is the j-th checkpoint) to the end checkpoint m_n, $\forall i \in [j, n]$, we fit the following function to learn the parameters α and β for each series:

$$\frac{\mathrm{PPL}_{m_i}(t \mid c)}{\mathrm{PPL}_{m_j}(t \mid c)} = \alpha + \beta \cdot (i - j). \qquad (1)$$

Note that different starting points might result in different trend estimations. We categorize the trends as follows based on β and its significance:

Upward trend. If $\beta > 0$ and its p-value is $<$ 0.05, we consider that the series follows an upward trend (*forgetting*).

Downward trend. If $\beta < -0$ and its p-value is $<$ 0.05, we consider that the series follows a downward trend (*still learning*).

Stagnated trend. If a series does not follow an upward or downward trend, and the start and end values fall in a restricted interval, that is, $0.95 \leq \mathrm{PPL}_{m_j}/\mathrm{PPL}_{\mathrm{AVG}} \leq 1.05$ and $0.95 \leq \mathrm{PPL}_{m_n}/\mathrm{PPL}_{\mathrm{avg}} \leq 1.05$, where $\mathrm{PPL}_{\mathrm{avg}} = \exp(\frac{1}{n-j+1}\sum_i \log \mathrm{PPL}_{m_i})$, we consider the series to be stagnated (*already learned*).

We design the criteria to roughly capture the trend of the perplexity series of each next-token prediction. Under these criteria, a stagnated series from an earlier checkpoint would continue to stagnate, and a series that follows an upward or downward trend earlier might turn stagnated afterwards. The criteria do not necessarily cover all the series—wavy series with a large variance do not fall within any category and are eliminated. For the rest of the section, for simplicity, we use *tokens* to refer to context-token pairs.

3.2 Analysis

Percentage of tokens. We show the percentage of tokens that follow each trend in Figure 2. Overall, the percentage of stagnated tokens increases and the percentage of the other two types of tokens decreases, indicating that more tokens get to be *learned* and fewer tokens are still learning or, more

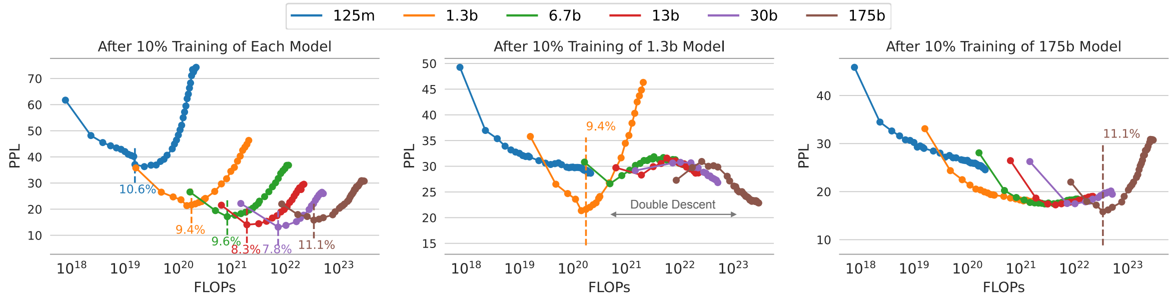


Figure 4: Perplexity of upward-trend tokens. Left: different models are evaluated on different subsets of tokens selected after 10% of training of individual models (all showing a downward-then-upward trend). Middle/right: all models are evaluated on the *same set of tokens*, selected after 10% of training of the 1.3B model and the 175B model respectively. The number next to the dashed line denotes the percentage of the selected tokens out of all tokens. Tokens selected by a smaller model (1.3B) present a double descent-like trend in larger models. Tokens selected by a larger model (175B) present a downward trend in the smaller models.

surprisingly, forgetting as training progresses. [6]

Stagnated tokens. We select stagnated tokens starting from 10% of training for a particular model and analyze the trajectory of these same tokens in other models. As shown in Figure 3 (middle), we observe that stagnated tokens after 10% of training in a small model (1.3B) also stagnate in larger models. However, the stagnated tokens selected by a large model (175B) still show a downward trend in smaller models. This suggests that larger models' stagnated tokens are roughly a superset of smaller models. On manual inspection, stagnated tokens are primarily non-content words such as prepositions, determiners, and punctuations.

Upward trend tokens. Similarly, we present the perplexity of upward trend tokens in Figure 4. The leftmost figure shows that such a phenomemon exists for all the models. For tokens that present an upward trend after 10% training of a small model (1.3B), we observe a stepwise double descent (Nakkiran et al., 2020) trend in larger models' trajectories, where the perplexity first increases and then decreases. We are the first to observe this phenomenon during language model training, and it suggests that larger models, with more computation and a larger capacity, first overfit to this subset of tokens and further generalize better for them. For the tokens identified after 20% training of the largest model (175B), the upward trend appears only at the end of training for the 13B and 30B models. We find it hard to characterize these tokens considering their contexts,[7] but the synergy across model sizes strongly suggests that consistent types of learning are triggered at particular computation levels for models across scales. [8]

Summary. In conclusion, large models first replicate small models' behavior on the same subset of tokens, and further unlock exclusive phenomena when fueled with more computation. In Appendix B.5, we find that trajectories of differently-sized models largely overlap when plotting against validation perplexity, indicating that they make similar predictions at a similar perplexity.[9]

4 Sequence-Level Generation

In this section, we extend the analysis from token-level predictions to entire sequences, up to 50-500 tokens. Larger language models consistently obtain a better perplexity in modeling human texts such as Wikipedia, with the perplexity decreasing as the model size and training computation increases (Figure 1). Autoregressive language models are probabilistic models of sequences that can generate strings of text. If larger models assign a higher probability to virtually all human-authored texts, what sequences do smaller models favor? We aim to first characterize these sequences and further analyze learning behavior on them to understand how models of different sizes evolve into their final distributions. In what follows, we first show that it is difficult to manually design such sequences, as large models can also favor corrupted or factually incorrect texts (§4.1). We then devise a decoding algorithm to automatically generate sequences fa-

[6] Only around 60% tokens are captured by our criteria and please find more details on other tokens in Appendix B.2.

[7] More details are in Appendix B.3.

[8] We explore the upward trends with different starting points and model scales in Appendix B.4.

[9] Please find more discussions in Appendix B.5.

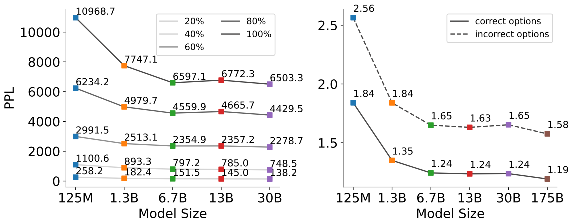


Figure 5: Scaling trends for corrupted datasets (p% random tokens) and options in multiple choice tasks. The perplexity on corrupted texts and incorrect options decrease as model size increases, even for sequences consisting of completely random tokens ($p = 100$).

vored by smaller models (§4.2), and conclude with an analysis of such sequences (§4.3).

4.1 Manual Design

Corrupted datasets. We hypothesize that injecting noise into human texts might reverse the scaling trend (i.e., perplexity on corrupted texts might increase as model size increases). To test this hypothesis, we replace 20%, 40%, 60%, 80%, and 100% of the subwords in each sequence with random subwords. We evaluate corrupted datasets on the *final* model checkpoints and report the perplexity in Figure 5 (left). Contrary to our hypothesis, downward trends largely retain across all noise levels, even when the entire sequence consists of random tokens (100%). This can be explained by the copy-and-complete interpretation for in-context learning described in Olsson et al. (2022): larger models fare better at making predictions to follow the context distribution than smaller models, even when the context is pure noise.[10]

Incorrect options of multiple-choice tasks. We next hypothesize that the perplexity of incorrect options for multiple-choice tasks might present an inverse scaling trend, as they are generally factually wrong. We present the perplexity of correct and incorrect options of 74 multiple-choice tasks from the BIG-Bench dataset in Figure 5.[11] However, we find that the perplexity of correct and incorrect options decreases as the size of the model increases.[12]

In summary, our initial attempt failed—we are not able to manually construct texts that are more probable in smaller models than larger models.

4.2 Methodology

To continue our search for such texts, we next devise a decoding approach that combines signals from two models and generates texts based on the interpolation of their distributions:

$$p'_i = \lambda_1 \cdot p_s(x_i|x_{<i}) + \lambda_2 \cdot p_l(x_i|x_{<i}); \qquad (2)$$

where p_s and p_l are the next-token distributions from the small and large models, respectively, and $\lambda_1, \lambda_2 \in [-1, 1]$. A set of λ_1 and λ_2 denotes a specific configuration. When $\lambda_1 = 0, \lambda_2 = 1$, it is simply decoding with the large model; when $\lambda_1 = 1, \lambda_2 = -1$, the decoding process favors the small model's prediction and suppresses the large model's prediction. This is the configuration that decodes sequences that small models have a lower perplexity on than large models.

We further remove tokens that have a negative score, and renormalize the distribution p'_i to ensure that the sum of the probabilities of all tokens is 1:

$$p(x_i|x_{<i}) = \frac{\mathbb{1}(p'_i > 0) \cdot p'_i}{\sum \mathbb{1}(p'_i > 0) \cdot p'_i}. \qquad (3)$$

Generation process. We decode sequences with two models, 125M and 30B, using different configurations of λ_1 and λ_2. We take the first 5 tokens of a subset of validation documents as prompts and generate 50 tokens conditioned on them.[13] We try greedy search and nucleus sampling (Holtzman et al., 2019) for decoding and evaluate the texts decoded from each configuration as follows: 1) we measure the text perplexity at final checkpoints of different-sized models to understand its scaling trend; 2) we measure the text perplexity at all intermediate checkpoints to understand how the perplexity evolves as training progresses.

4.3 Analysis

Inverse scaling. As shown in Figure 6 (row 1), we confirm that the perplexity of texts generated with the $p_s - p_t$ configuration presents an inverse scaling trend—perplexity increases as model size increases (column 1, 5). Other configurations either only show a modest upward trend (p_s), or a normal downward trend (p_l and $p_l - p_s$). Even though models of intermediate sizes (1.3B, 6.7B, 13B) are not involved in decoding, the scaling trend holds systematically across all model sizes. To further verify

[10] Please find details on corrupted datasets in Appendix C.1.

[11] Details on task selection are in Appendix D.1.

[12] To clarify, we are not discussing task accuracy here, but the scaling trend of correct and incorrect options. Find examples of correct and incorrect prompts in Table 8.

[13] We also generate longer sequences up to 100 and 500 words and the conclusions hold similarly. More discussions can be found in Appendix C.5.

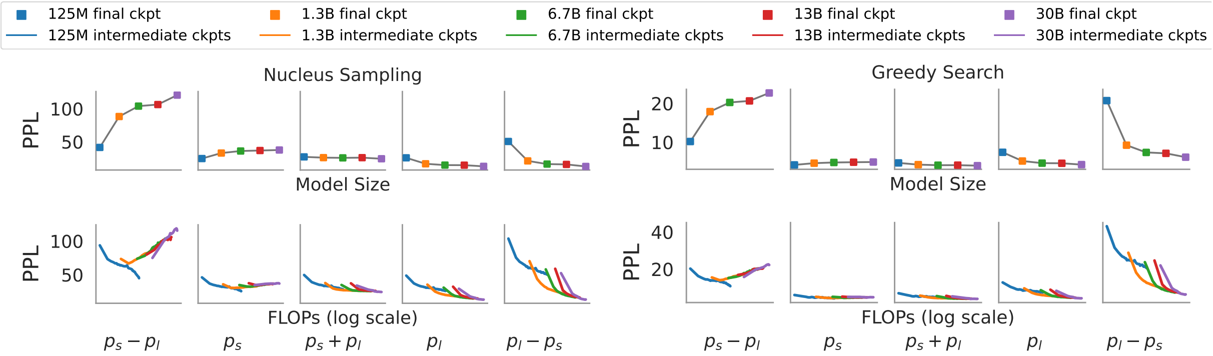


Figure 6: Perplexity of texts (generated with $\lambda_1 p_s + \lambda_2 p_l$) evaluated with differently-sized final model checkpoints (first row) and perplexity trajectory evaluated over intermediate checkpoints against FLOPs (second row). Each column denotes one configuration with different λ_1 and λ_2. Note that all the texts are generated by combining signals only from 125M and 30B models, but are evaluated over all the model scales.

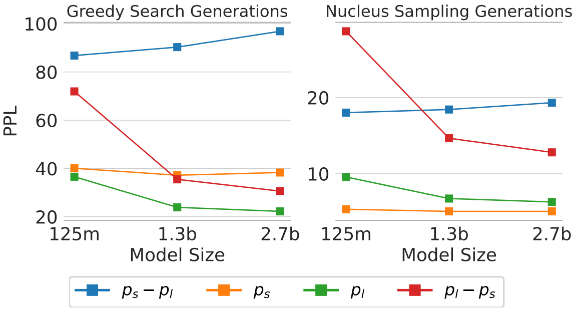


Figure 7: Evaluations using GPT Neo models on texts generated with OPT 125M and OPT 30B models. The perplexity follows a similar trend as OPT, suggesting a systematic distribution shift between model sizes.

the universality of the phenomenon in other families of language models, we evaluate the generated texts with final GPT Neo checkpoints (Black et al., 2021), which were trained on the Pile dataset (Gao et al., 2020). As shown in Figure 7, the perplexity trend aligns with OPT models. This confirms that the texts generated with our approach are not a result of model or data artifacts, but embody universal properties exhibiting a similar scaling trend in other model families.

Perplexity trajectory of generated sequences. In the second row of Figure 6, we present the perplexity trajectory of texts generated with different configurations. We observe that texts generated based on $p_s - p_l$ and, to a less extent, p_s, largely differ from the other configurations: 125M checkpoints present a downward trend, while other checkpoints present an upward trend. This might suggest that differently-sized models optimize in different directions for phenomena specific to these texts. However, taking a closer look, we observe that the 1.3B model also shows a downward trend at the beginning, which turns upward afterwards. This indicates that all models improve the perplexity of these texts at first but, with more training FLOPs, larger models shift away from this specific distribution where the 125M model stalls. In Appendix C.7, we further show that perplexity of the sequences decoded by contrasting the two models ($p_s - p_l$ and $p_l - p_s$) are less aligned with validation perplexity as other configurations.

Generated examples. Table 1 presents examples generated with different configurations. We find that the generations from $p_s - p_l$ are grammatically correct and carry actual meanings both for greedy search and nucleus sampling, but manifest other issues: 1) they entail highly-unlikely semantic usages such as *Fortunately, it wasn't all that great*—an ending word with a negative sentiment should be more prevalent; 2) the nucleus sampling examples, despite being fluent and consistent, hardly ground to real world scenarios. This suggests that small models are highly capable linguistically, and learning at scale primarily focuses on acquiring other types of knowledge.[14]

5 Downstream Tasks

In this section, we examine the trajectory of downstream tasks, evaluated on few-shot in-context learning (ICL).

5.1 Task Selection and Evaluation

BIG-Bench (Srivastava et al., 2022) is a large collection of tasks for evaluating language models. We evaluate intermediate checkpoints on its subset

[14] We present more generated examples and have a more detailed discussion on generation quality in Appendix C.3.

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
p_s	that bad. I was in the middle of a long day of work and I was in the middle of a long day of work. I was in the middle of a long day of work. I was in the middle of a long day	that bad. Not because the weather wasn't bad, but because of how many people didn't move their car around. For those who did, I wanted to say thanks to everyone else who still had a tire change on. That doesn't change
$p_s + p_l$	bad. I was able to get a few things done, and I was able to get a few things done. I was able to get a few things done, and I was able to get a few things done. I was able to	cold and we didn't have to set up a heated bed so we wouldn't freeze off in the middle of the night. It was a nice fall day and I had just finished wrapping up the color scheme on the wall. I still haven
p_l	bad. I got to spend some time with my family, and I got to see my friends. I got to see my friends, and I got to see my family. I got to see my family, and I got to see my	gloom, glum, and doom. One nice thing was the gift of snow for a few minutes this afternoon. It was fun to watch it pile up on the porch, watch the kids watch it pile up, and then run out and scatter
$p_l - p_s$	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

Table 1: Examples generated with greedy decoding and nucleus sampling under different configurations. The prompt is *Fortunately, the day wasn't all*.

of 74 multiple-choice tasks.[15] BIG-Bench comes with predefined templates with a unified QA format for in-context learning, which mitigates the extra complexity of prompt design.[16]

We focus on the 2-shot setting. Following Srivastava et al. (2022), we randomly select two in-context learning examples (excluding the evaluation example itself) for each test instance and pick the candidate for each evaluation example that has the highest probability normalized over its length. We use the average 2-shot accuracy of downstream tasks as a proxy for in-context learning capability.

5.2 Trajectory of ICL Performance

ICL vs. valid PPL. From Figure 8 (leftmost), it is evident that the downstream task performance strongly correlates with validation perplexity across all model sizes. The curves of different model sizes significantly overlap, indicating that when a small model and a large model are trained to the same perplexity level, they achieve comparable downstream task performance.

ICL vs. other metrics. it is evident that plotting task accuracy against various metrics yields distinct patterns. Notably, when subjected to an equal amount of training FLOPs, the performance of smaller models consistently surpasses that of larger models, with the exception of the 125M model. This observation implies that larger models possess untapped potential for improvement, especially when provided with more training FLOPs or data (Hoffmann et al., 2022; Touvron et al., 2023). Conversely, the remaining two plots indicate that larger models consistently outperform smaller ones when trained with the same number of training tokens and training steps.

5.3 Linearity vs. Breakthroughness Tasks

We select 12 tasks that present a linearity scaling pattern and 6 tasks that present a breakthroughness scaling pattern,[17] and plot the perplexity of the correct and incorrect options for each group of tasks against validation perplexity in Figure 9.

The performance of breakthroughness tasks increases tremendously as the validation perplexity drops below 8. The perplexity gap between the correct and incorrect options also starts to expand at this point for the 30B and 175B models. In contrast,

[15] Mode details on task selection are in Appendix D.1.

[16] Examples of prompts are in Appendix D.2.

[17] Breakthroughness here similar to the emergent dehavior defined in Wei et al. (2022). Details on how we select linearity and breakthroughness tasks are in Appendix D.3.

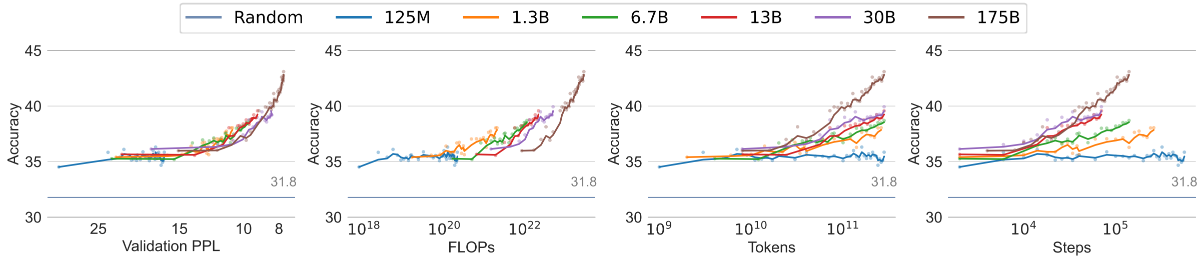


Figure 8: The 2-shot performance trajectory of 74 BIG-Bench tasks. The performance is measured by the average accuracy on the default set and plotted against validation perplexity, training FLOPs, training tokens and number of training steps. The task accuracy aligns with validation perplexity across different model sizes.

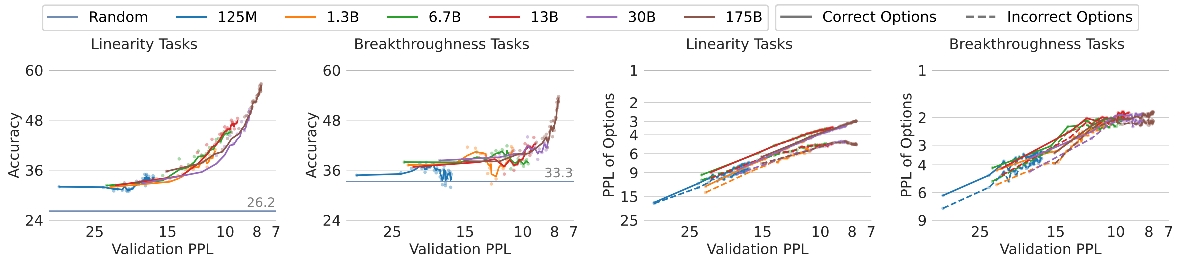


Figure 9: Trajectory of 2-shot in-context learning performance (left two) and option perplexity (right two) of 12 linearity and 6 breakthroughness tasks against validation perplexity. The perplexity divergence of correct and incorrect options drives the performance improvement.

the accuracy of linearity tasks gradually increases. The perplexity of correct and incorrect options first decrease as validation perplexity decreases, and it is only at the end of the curve that the perplexity of correct and incorrect options starts to diverge. This suggests that improvements in downstream accuracy are not generally driven by the model learning to assign a lower probability to incorrect candidates, but rather driven by the perplexity divergence of correct and incorrect options.

5.4 Breakthroughness Tasks Learn Smoothly on Trajectory

In Appendix D.4, we provide a detailed analysis of task accuracy in relation to perplexity and FLOPs for individual linearity and breakthroughness tasks. The corresponding plots can be found in Figure 17 and Figure 18. As expected, these plots exhibit a significantly larger variance, showcasing substantial fluctuations in task performance during the training process. However, we still observe a notable alignment between task accuracy and validation perplexity across different model scales. Notably, the breakthroughness tasks, which demonstrate sudden performance improvements at the final checkpoints, display a smooth and continuous growth trend along the training trajectory. This observation reinforces the findings of a recent study conducted by Schaeffer et al. (2023), where they discovered that modifying downstream task metrics results in gradual changes in performance rather than abrupt and unexpected shifts as model scale increases. These results suggest that when examining task performance at a finer level, either through continuous metrics or continuous model checkpoints, task performance largely exhibits a smooth growth pattern in tandem with validation perplexity. Nevertheless, as suggested by Ganguli et al. (2022), accurately predicting the learning curve of a specific task still remains challenging.

6 Related Work

Phase change. Olsson et al. (2022) study induction heads to understand the formation of in-context learning ability. The main finding is that there exists a critical phase chage (Power et al., 2022; Nanda and Lieberum, 2022) that forms the in-context learning ability. Our studies are in the same spirit as these work, but we did not discover any phase change for the phenomena we examined; all of them evolve steadily as training progresses.

(Inverse) scaling laws. Previous work studies scaling on downstream tasks (Wei et al., 2022; Srivastava et al., 2022), pre-training data (Hernandez et al., 2022), architectures (Tay et al., 2022a), biases (Tal et al., 2022), and other domains, such as vision tasks and neural machine translation (Alabdulmohsin et al., 2022). Our work studies different scaling behaviors over model trajectories.

Inverse scaling refers to a scaling behavior where increasing the model size leads to worse performance for a downstream task (Perez and McKenzie). Part of our work intends to understand the distributional shift from small models to large models for language modeling along training trajectories, which overlaps with the theme of inverse scaling.

Perplexity vs. downstream performance. Regarding the pre-training/fine-tuning paradigm, Wettig et al. (2022) and Tay et al. (2022a) find that a lower pre-training perplexity does not necessarily translate to better fine-tuning performance. For zero-shot inference, Saunshi et al. (2020) mathematically shows that doing well in language modeling benefits downstream tasks. On the contrary, Shin et al. (2022) claims the opposite relationship for in-context learning performance and perplexity when training language models with different corpora, but they only test four downstream tasks on a few model checkpoints. Our work extensively evaluates multiple domains and tasks on both language modeling and downstream tasks across checkpoints of different scales, which entails less variance.

Effective scaling Several prior studies have focused on effectively scaling models by examining limited compute settings (Geiping and Goldstein, 2022), exploring different objectives (Tay et al., 2022b; Artetxe et al., 2022b), and investigating different architecture and training setups (Scao et al., 2022b). This work specifically examines model scales under a unified setting, but the proposed techniques can be applied to other settings as well.

7 Conclusion

To summarize, our study demonstrates that validation perplexity is a reliable indicator of the behavior of OPT models, regardless of their sizes. Larger models, with increased computational power and capacity, exhibit behavior similar to that of smaller models while also unlocking new phenomena and capabilities as validation perplexity decreases further. However, there are certain exceptional cases where models behave differently, sometimes even in opposite directions, such as in the perplexity of texts generated by contrasting two models. This suggests that the underlying model distributions are not entirely identical at the same perplexity level.

The availability of a larger number of open-sourced model checkpoints, such as those provided by Biderman et al. (2023), offers opportunities for interpreting language model behaviors through the analysis of training trajectories. The techniques we propose can be extended to analyze language models trained using different resources and methodologies. Additionally, we leave open questions for future research, such as further exploring the phenomenon of double-descent more in-depth.

Limitations

We discuss the limitations of the work as follows:

- One major limitation of our work is that we analyze language models pre-trained with the same data, similar training procedures, and the same autoregressive language modeling objective. Our findings may support model families trained in this restricted setting. When comparing models trained with different corpora, such as Neo GPT NEO (Black et al., 2021) and BLOOM (Scao et al., 2022a), different architectures and objectives, such as retrieval-based language models (Khandelwal et al., 2020; Zhong et al., 2022; Borgeaud et al., 2021) and sparse models (Fedus et al., 2022; Artetxe et al., 2022a), the relationship between validation perplexity and downstream task performance could be more obscure.

- For downstream task evaluation, we only evaluate on multiple-choice tasks, where the evaluation protocol is the most similar to the pre-training objective. Evaluating on generation-based tasks is more messy and hard to scale up, and we will leave it as future work. Another risk is that as we always take aggregated measurements over tasks, it might conceal important patterns of individual tasks.

- We do not provide a concrete explanation for the double-descent behavior that consistently occurs during pre-training, nor do we know if it is an artifact of the data, the objective or the optimization process. We consider it an interesting phenomenon and will look more closely into it in future works.

Acknowledgement

We thank Sadhika Malladi for helping out with writing and having insightful discussions on the project with the authors. We thank Tianyu Gao for helping out running experiments on open-text generation in the Appendix. We also thank Stephen Roller, Srini Iyyer, Todor Mihaylov, Xiaochuang Han, and all members of the Princeton NLP group for helpful discussion and valuable feedback. This work was conducted when Mengzhou Xia was interning at Meta Platforms, Inc.

References

Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. 2022. Revisiting neural scaling laws in language and vision. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2022a. Efficient large scale language modeling with mixtures of experts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Mikel Artetxe, Jingfei Du, Naman Goyal, Luke Zettlemoyer, and Ves Stoyanov. 2022b. On the role of bidirectionality in language model pre-training. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono-and cross-lingual pretraining dynamics of multilingual language models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The grammar-learning trajectories of neural language models. In *Association for Computational Linguistics (ACL)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research (JMLR)*.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. 2022. Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Jonas Geiping and Tom Goldstein. 2022. Cramming: Training a language model on a single gpu in one day. *arXiv preprint arXiv:2212.14034*.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. 2022. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Association for Computational Linguistics (ACL)*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A Smith. 2021. Probing across time: What does roberta know and when? In *Findings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 820–842.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2020. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations (ICLR)*.

Neel Nanda and Tom Lieberum. 2022. A mechanistic interpretability analysis of grokking. *Alignment Forum*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Ethan Perez and Ian McKenzie. Inverse scaling prize: Round 1 winners.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems (NeurIPS)*.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*.

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. 2020. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations (ICLR)*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022a. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Bideman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. 2022b. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Yarden Tal, Inbal Magar, and Roy Schwartz. 2022. Fewer errors, but more stereotypes? the effect of model size on gender bias. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q Tran, Dani Yogatama, and Donald Metzler. 2022a. Scaling laws vs model architectures: How does inductive bias influence scaling? *arXiv preprint arXiv:2207.10551*.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022b. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations (ICLR)*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training language models with memory augmentation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

A Checkpoint Details

We present the checkpoint information in Table 2. OPT models of different sizes are trained with different batch sizes and end up training with different number of steps given the same amount of training tokens. We select early-stage checkpoints every 4K steps for evaluation, and enlarge the interval to 10K or 20K for late stage checkpoints. There are a few checkpoints missing/corrupted from the training process, e.g., 125M 180K, and we have to eliminate them our evaluation.

All OPT models are trained with 300B tokens, of which 180B tokens are unique. This training procedure means that OPTs are trained with repeated data, though training with non-repeating data consistently lead to better performance in language modeling and downstream tasks (Lee et al., 2022; Hernandez et al., 2022).

B Next-Token Predictions

B.1 Data Used in the Main Paper

We use the Gutenberg PG-19 (Rae et al., 2020) subset as the main dataset for analysis in the main paper. This validation subset contains 50 lines of texts, and we take the first 2048 tokens of each line for analysis, resulting in 102350 context-token pairs. We observe similar patterns when evaluated on other validation subsets such as Wikipedia and opensubtitles, and we omit the results for brevity.

B.2 Trajectory of Other Tokens

We set our criteria to be relatively strict to make sure that the perplexity trajectory of the selected tokens does present the trend (stagnated/upward/downward) we expect. We present the trajectory of the tokens that do not fall into any of the categories in Figure 10. We find that the trend of these tokens are not consistent across models. After 10% of training, the curves of 125M, 1.3B, 6.7B present a slight double-descent trend, and for the rest of the models, the curves present a downward/stagnated trend. After 40% of training, the curves of 125M present a slight double-descent trend towards the end, and the curves of other models present a downward/stagnated trend. This suggests that the rest of the tokens might contain a larger variance in their perplexity trajectories.

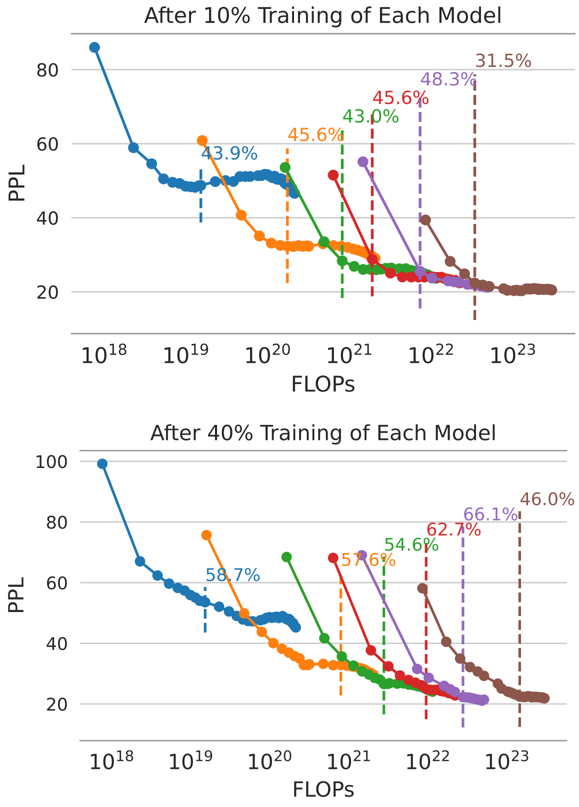


Figure 10: Perplexity of tokens that do not fall into any of the categories. Different models are evaluated on different subsets of tokens selected after 10% (up) and 40% (down) of training of individual models. The trends are not consistent across different model sizes.

B.3 Properties of Stagnated and Upward-Trend Tokens

We show an example paragraph in Table 3, where the stagnated tokens are in blue, upward-trend tokens are in red and downward-trend tokens are in green. It's easy to see that stagnated tokens are mostly connecting words, determiners, punctuation and continuation of words. However, we find it hard to characterize the tokens that present an upward-trend in perplexity simply based on token types. We made attempts to further decipher what language properties this subset might entail based on the part-of-speech tags and positions in sequences, and did not observe any obvious patterns when compared to all the tokens in the validation set. One thing we are sure is that the phenomenon of the upward trend in perplexity as well as the double-descent phenomenon on a certain subset of tokens systematically appears across all model sizes. Therefore, this subset of context-token pairs must embody certain intrinsic language properties, which might be beyond our comprehension so far.

# Params	LR	Batch Size	# Steps	# CKpt	CKpt Steps
125M	$6.0e-4$	0.5M	600K	36	2K, 6K, 10K, 14K, 18K, 22K, 26K, 30K, 34K, 38K, 40K, 60K, 80K, 100K, 120K, 140K, 160K, 200K, 220K, 240K, 260K, 280K, 300K, 320K, 340K, 360K, 380K, 400K, 420K, 440K, 460K, 480K, 500K, 520K, 540K, 560K
1.3B	$2.0e-4$	1M	300K	22	2K, 6K, 10K, 14K, 18K, 22K, 26K, 30K, 34K, 38K, 40K, 60K, 80K, 100K, 120K, 140K, 160K, 180K, 200K, 220K, 240K, 260K
6.7B	$1.2e-4$	2M	150K	21	2K, 6K, 10K, 14K, 18K, 22K, 26K, 30K, 34K, 38K, 40K, 50K, 60K, 70K, 80K, 90K, 100K, 110K, 120K, 130K, 140K
13B	$1.0e-4$	4M	75K	18	2K, 6K, 10K, 14K, 18K, 22K, 26K, 30K, 34K, 38K, 42K, 46K, 50K, 54K, 58K, 62K, 66K, 70K
30B	$1.0e-4$	4M	75K	18	2K, 6K, 10K, 14K, 18K, 22K, 26K, 30K, 34K, 38K, 42K, 46K, 50K, 54K, 58K, 62K, 66K, 70K
175B	$1.2e-4$	2M	150K	32	4K, 8K, 12K, 16K, 20K, 24K, 36K, 40K, 44K, 48K, 52K, 56K, 60K, 64K, 68K, 72K, 76K, 80K, 84K, 88K, 92K, 96K, 100K, 104K, 108K, 112K, 120K, 124K, 128K, 132K, 136K, 140K

Table 2: Checkpoint (Ckpt) information for OPT models. LR denotes learning rate. Note that we take these checkpoints for practical reasons and the distance between checkponts are not evenly spaced. But it should not affect the analysis.

It would be interesting to do an in-depth analysis in understanding why it happens during pre-training, and how it connects to natural language properties.

B.4 More Explorations on Upward Trends

In this section, we explore the subset of tokens that present an upward trend when selected by models of other sizes from the main paper (6.7B, 13B, 30B). We present the perplexity trajectory of these tokens in Figure 11. For the subset of tokens selected after 10% of training of the 6.7B model, the larger models' perplexity also increase but only the largest 175B model presents a double descent behavior where the perplexity declines further. When the tokens are selected after 40% of training of 6.7B, the trends remain similar but the change is mulch more mild. Overall, except the model that is used to select the tokens, the curves of other models present a similar trend, and we will show that these curves overlap with each other almost completely when plotting against validation perplexity in the next subsection. The consistent occurrence of double-descent behavior along the trajectory shows that it's a phenomenon happening universally across the entire autoregressive pre-training process.

B.5 Results against Validation Perplexity

In the main paper, we mostly plot measurements against FLOPs, in this section, we plot the perplexity trajectory of tokens that present different trends against **validation perplexity** in Figure 12. These figures present the same series of results as Figure 3 and Figure 4, except that the x-axis is validation perplexity. As mentioned in section 2, we use the aggregated perplexity of a number of subsets as the validation perplexity.

From Figure 12, we see that given a similar level of validation perplexity, for different subsets of tokens, the trajectories of models across sizes overlap well with each other, suggesting that the predictions for these tokens are similar across model scales at

After 10% training of 1.3B *model*	*After 10% training of* 175B *model*
Appropri ate ; pertaining to the subject . \n P ect oral . The bone which forms the main rib or support at the forward edge of a bird ’s wing . \n Pers istent . Keeping at it ; determination to proceed . \n Per pend icular . At right angles to a surface . This term is sometimes wrongly applied in referring to an object , particularly to an object which is vertical , meaning up and down . The blade of a square is perpend ie ular to the handle at all times , but the blade is vertical only when it points to the center of the earth . \n P ern icious . Bad ; not having good features or possessing wrong attributes . \n P end ulum . A bar or body suspended at a point and adapted to swing to and fro . \n Per pet ual . For all time ; un ending or unlimited time . \n P hen omen a . Some peculiar happening , or event , or object . \n P itch . In aviation this applies to the angle at which the blades of a prope ller are cut . If a prope ller is turned , and it moves forward ly in the exact path made by the angle , for one complete turn , the distance traveled by the prope ller ax ially indicates the pitch in feet . \n Pl acement . When an object is located at any particular point , so that it is operative the location is called the placement . \n Pl ane . A flat surface for supporting a flying machine in the air . Plane of movement per tains to the imaginary surface described by a moving body	Appropri ate ; pertaining to the subject . \n P ect oral . The bone which forms the main rib or support at the forward edge of a bird ’s wing . \n Pers istent . Keeping at it ; determination to proceed . \n Per pend icular . At right angles to a surface . This term is sometimes wrongly applied in referring to an object , particularly to an object which is vertical , meaning up and down . The blade of a square is perpend ie ular to the handle at all times , but the blade is vertical only when it points to the center of the earth . \n P ern icious . Bad ; not having good features or possessing wrong attributes . \n P end ulum . A bar or body suspended at a point and adapted to swing to and fro . \n Per pet ual . For all time ; un ending or unlimited time . \n P hen omen a . Some peculiar happening , or event , or object . \n P itch . In aviation this applies to the angle at which the blades of a prope ller are cut . If a prope ller is turned , and it moves forward ly in the exact path made by the angle , for one complete turn , the distance traveled by the prope ller ax ially indicates the pitch in feet . \n Pl acement . When an object is located at any particular point , so that it is operative the location is called the placement . \n Pl ane . A flat surface for supporting a flying machine in the air . Plane of movement per tains to the imaginary surface described by a moving body

Table 3: An example paragraph to demonstrate tokens that present a stagnating, upward or downward trend after 10% training of 1.3B and 175B models. Tokens that present an upward trend in perplexity are in Red; tokens that present a downward trend are in Green; stagnating tokens are in Blue. Black tokens do not present a clear trend.

a fixed level of validation perplexity. The only exception is the upward-trend tokens selected after 10 % training of 1.3B, where evaluating with 1.3B presents a clear upward trend as the validation perplexity increases, while the models larger than 1.3B present a overlapping double descent-like trend. This indicates that the underlying distribution of models at the same level of perplexity are largely similar but could differ in edge cases.

These results lays the foundation for downstream task evaluations, which heavily relies on the pretraining objective for evaluation.

C Sequence-Level Generation

C.1 Details of Corrupted Datasets

We corrupt texts from the opensubtitle subset of the validation set by replacing $p\%$ tokens (subwords) with randomly sampled tokens in the sequences. We cap the max length of a sequence to be 100, though changing max length values does not affect the conclusion. Although the perplexity on these corrupted sequences is extremely high, especially when the replacement rate is high, it is still much lower than a truely random model (the perplexity of a random model should be $|V|$ where V is the vocabulary), even for the fully corrupted dataset. It reflects that larger language models are better at exploiting random patterns to produce in-

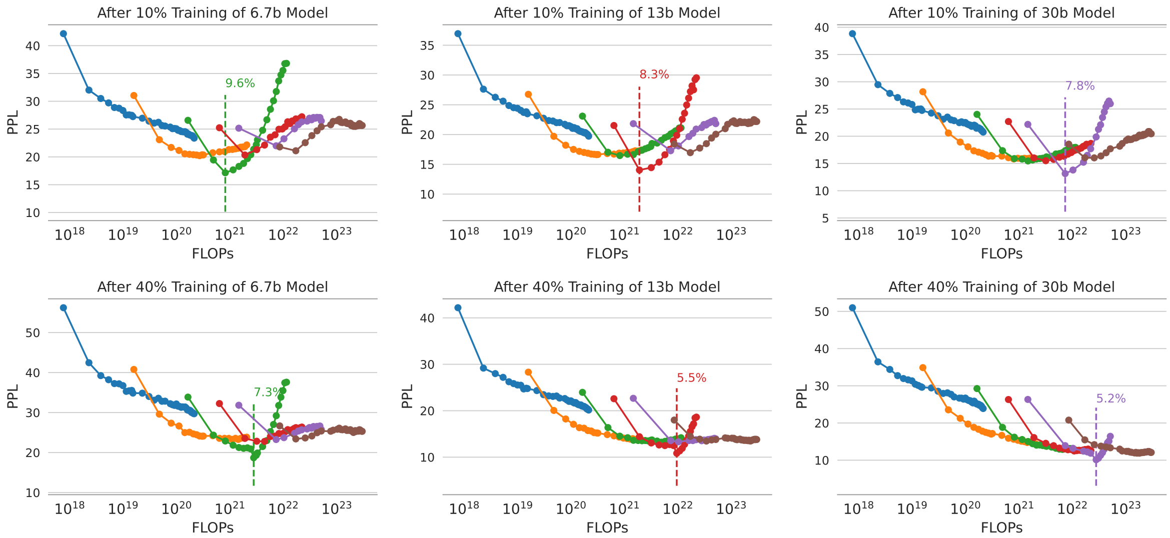


Figure 11: Perplexity of tokens that present an upward trend after 10% or 40% of training of the 6.7B, 13B and 30B models. For each figure, all the models are evaluated on the same subset of tokens.

distribution contents than smaller counterparts. We also tried other ways of corruption, such as deleting, inserting, repeating tokens/spans, and all these corruptions result in similar scaling trends.

C.2 Comparison to Li et al. (2022)

Our decoding approach is similar to the contrastive decoding method (CD) proposed in Li et al. (2022), though initially for completely different purposes. The difference between the two methods is in the subtraction space. The contrastive score in CD is defined by dividing the expert probability over amateur probability, which is equivalent to subtraction in the log probability space. Our approach operates subtraction in the probability space directly, ruling out unlikely options where the small model is much more confident than the large model directly. Due to this different design choice, our approach does not need to add the adaptive plausibility restriction, nor involve any additional hyperparameter. Subtraction in the probability space easily eliminates the false positive cases.

We initially propose the approach to decoding sequences that small models favor more than large models to understand the distributional shift across model scales, while contrastive decoding proposed in Li et al. (2022) is a general open-generation approach. Nonetheless, our approach could be an effective and lightweight alternative for open-ended generation without the need to adjust hyperparameters. In Appendix C.4, we show that our approach outperforms nucleus sampling on MAUVE scores.

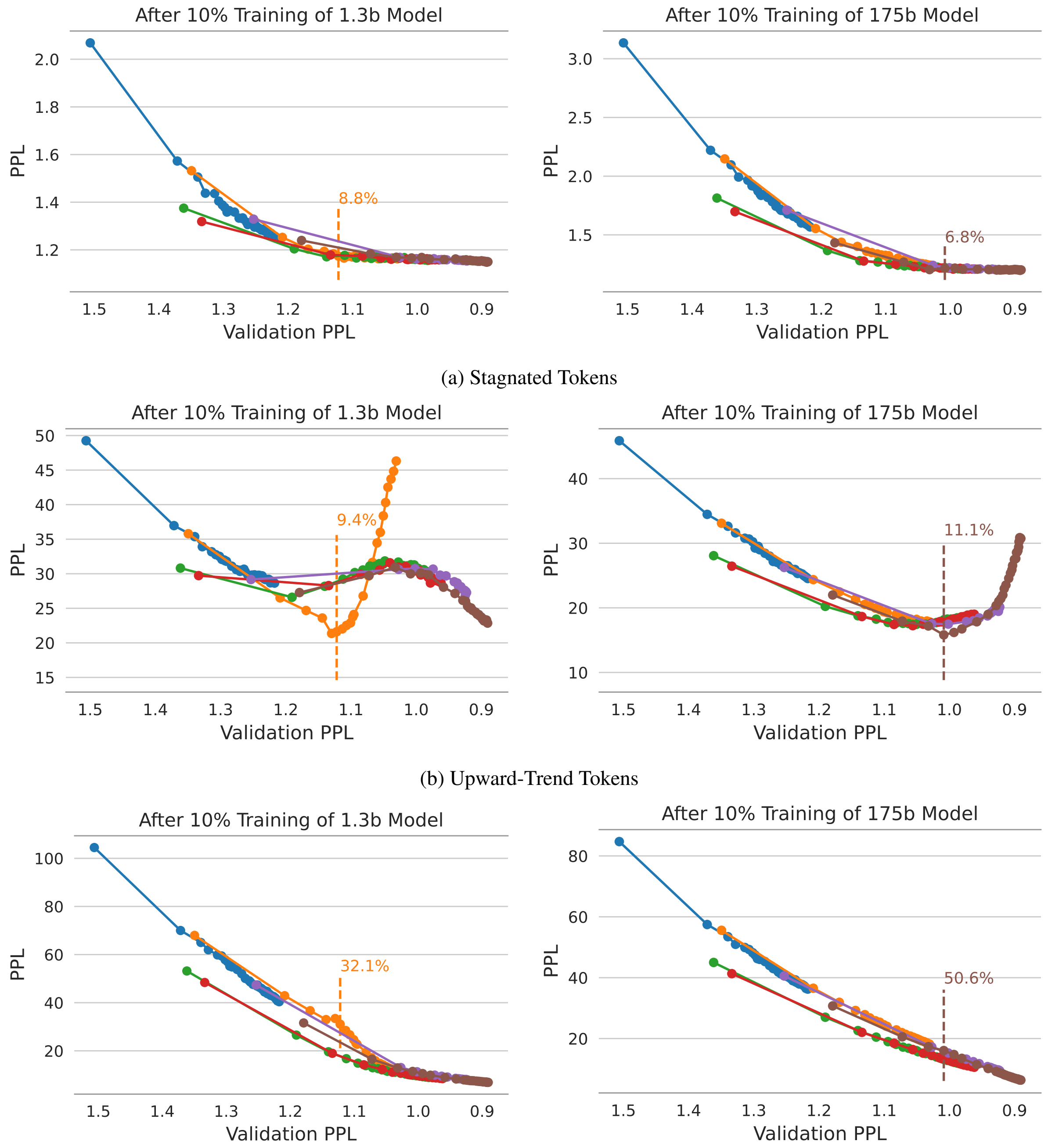
C.3 Generation Quality

To have a better understanding of the overall quality of the generated sequences, we evaluate these sequences decoded with each configuration in Figure 6 using MAUVE scores (Pillutla et al., 2021). We present the MAUVE scores in Figure 13 . Our generation protocol is slightly different from the standard open-ended generation practices in that we only use 5 tokens as prompts for generation, while usually at least 128 tokens are used (Krishna et al., 2022; Su and Collier, 2022; Li et al., 2022). Using fewer tokens as prompts leads to a higher generation diversity, and the generated distribution could be largely different from the ground-truth sentences. Therefore, we find that the MAUVE scores of our generated sequences are much lower than reported in open-ended generation literature.

Comparing the two decoding protocols, subtraction between two distributions ($p_s - p_l$ and $p_l - p_s$) leads to a better generation quality than summing the two ($p_s + p_l$) for greedy sampling, but vice versa for nucleus sampling. To verify the effectiveness of the approach, we compare it to nucleus sampling with standard open-generation protocols in Appendix C.4.

C.4 Open-ended Generation Evaluation

We follow the generation protocol in Krishna et al. (2022) for open-ended generation, where we generate sequences with a maximum length of 128 given contexts that have 256 tokens. We decode sequences based on either $p_l - p_s$ or p_l with greedy



(a) Stagnated Tokens

(b) Upward-Trend Tokens

(c) Downward-Trend Tokens

Figure 12: Perplexity of stagnated tokens, upward-trend tokens and downward-trend tokens against validation perplexity. Curves of different models largely overlap with each other, signifying that validation perplexity is a good indicator of model behaviors along the trajectory, e.g. the double descent-like phenomenon, agnostic to model sizes.

decoding or nucleus sampling ($p = 0.9$) and evaluate the quality of the generation with MAUVE scores.

We present the results in Table 4. Consistently, our approach to subtracting the probability from a small model from a large model outperforms nucleus sampling with one single model consistently, indicating that our approach has the potential to serve as an effective general decoding method for open-ended generation.

C.5 Generating Longer Sequences

We extend the study to generate longer sequences up to 100 and 500 tokens, and we present perplexity trajectories in Figure 14 and Figure 15, respectively. We find that the inverse scaling trend across model

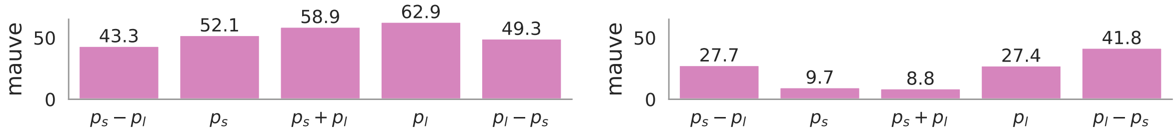


Figure 13: MAUVE scores (the higher, the better) on sequences with a maximum length of 50.

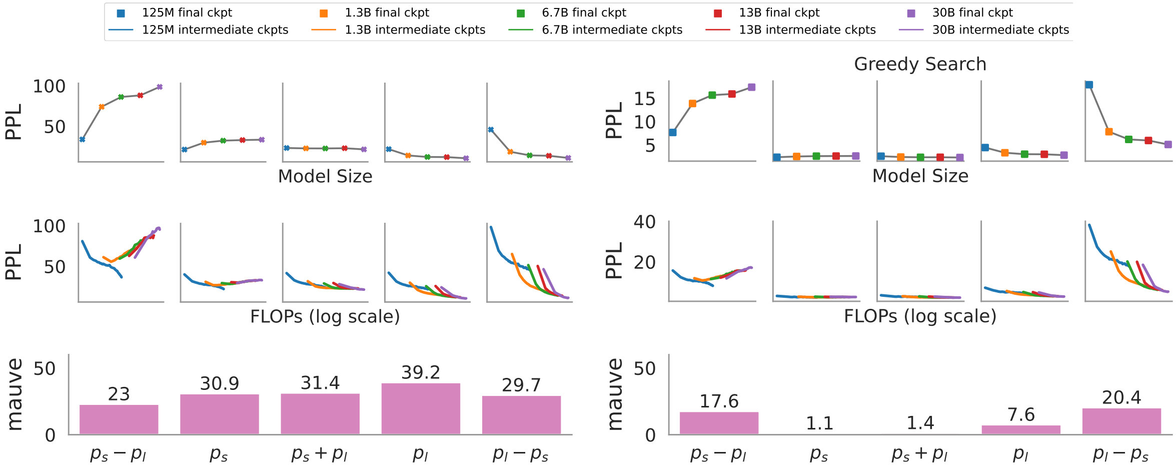


Figure 14: Greedy search and nucleus sampling results with generations of a length of 100.

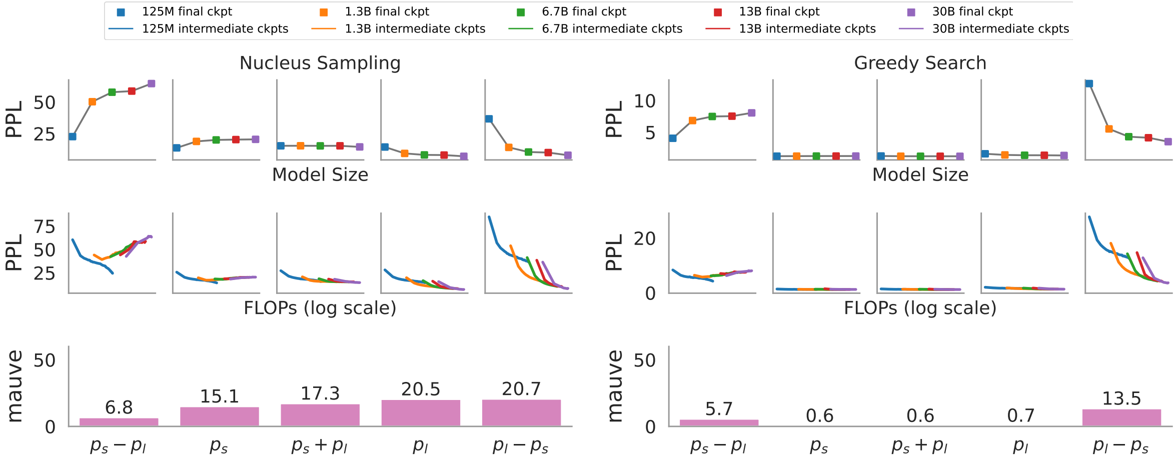


Figure 15: Greedy search and nucleus sampling results with generations of a length of 500.

sizes and the opposite perplexity trend between the 125M and 30B also hold for longer sequences. MAUVE scores on generated sequences of different lengths are largely consistent. The longer the decoded sequences are, the worse the overall quality.

C.6 Examples of Generated Sequences

We present more examples of generated sequences in Table 5 and Table 6. Similar to Table 1, we find that nucleus sampling with $p_l, p_l - p_s$ and greedy search with $p_l - p_s$ constantly generate high-quality sequences. Greedy decoding $p_s - p_l$ generates mediocre sequences that are largely grammatical and fluent, but less coherent and sometimes contain hallucinations.

C.7 Validation Perplexity vs. Perplexity of Generated Texts

We plot the perpelxity trajectory of generated texts against validation perplexity in Figure 16. The trajectories largely align well across model sizes for $p_s, p_s + p_l$ and p_l but diverge in the case of $p_l - p_s$ and $p_s - p_l$. This indicates that the underlying distributions of different-sized models given the same perplexity are similar but not exactly identical.

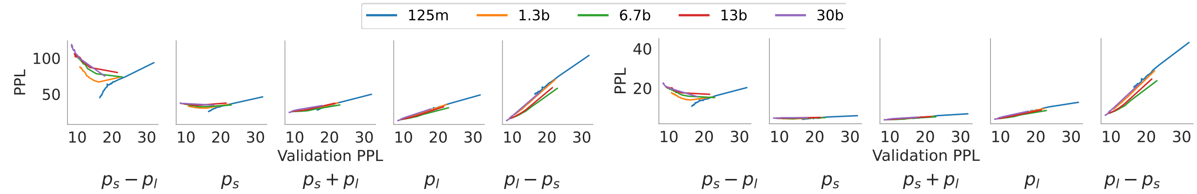


Figure 16: Validation perplexity vs. perplexity of generated texts. We find that models of different scales do not have the same perplexity on the generated texts when decoded with $p_s - p_l$ or $p_l - p_s$ given the same validation perplexity, but they largely align when decoded with other configurations.

	greedy	**nucleus**
350m	0.065	0.807
350m-125m	0.795	**0.852**
1.3b	0.164	0.877
1.3b-125m	0.851	**0.890**
1.3b-350m	0.888	0.886
2.7b	0.237	0.832
2.7b-125m	0.815	**0.851**
2.7b-350m	0.846	0.843

Table 4: MAUVE scores of generations following open-generation protocols. Nucleus sampling on an interpolated distribution ($p_l - p_s$) consistently outperforms decoding with a single model (p_l).

D Downstream Tasks

D.1 Task Selection and Evaluation

Out of comuputational considerations, we only evaluate multiple-choice tasks that have fewer than 1000 evaluation examples. The list of selected tasks is shown in Table 7. We report 2-shot in-context learning performance on the `default` set of each BIG-Bench dataset.

D.2 Prompts

We use fixed prompt formats from the BIG-Bench datasets. Optimizing the prompts might lead to extra margins in performance. Studying the relationship between prompt formats and downstream task performance along the trajectory is interesting, but we consider it out of the scope of this work. We present examples from four datasets in Table 8.

D.3 Linearity and Breakthroughness Tasks

Srivastava et al. (2022) identify tasks showing a linearity or breakthroughness pattern and (Wei et al., 2022) coin the term *emergent ability* for models showing breakthroughness patterns on certain tasks. Previous works mainly study scaling patterns of downstream tasks with final model checkpoints, and we extend this to training trajectories of models across scales. We largely follow Srivastava et al. (2022) to identify tasks with linearity and breakthroughness patterns – the former depicts the trend where the task performance scales with the model size reliably, and for the latter, the performance remains low until a critical model size.

We select 12 tasks that show a linearity pattern and 6 tasks that show a breakthroughness pattern based on the metrics proposed in (Srivastava et al., 2022). For each model size x_i and the corresponding performance y_i, the metrics are defined as

$$L = \frac{I(y)}{\sqrt{\frac{1}{n}\sum_i z_i^2}}; B = \frac{I(y)}{\sqrt{\text{Median}(\{z_i^2\})}}; \quad (4)$$

where $I(y) = \text{sign}(\arg\max_i y_i - \arg\max_i y_i) \cdot(\max_i y_i - \min_i y_i)$ is a measure to capture the overall improvement of performance when scaling up. We find that these two measures are not sufficient for identifying the scaling trends for linearity and breakthroughness, thus we also manually check the scaling pattern to verify. The linearity and breakthroughness tasks are lists in Table 9.

D.4 Trajectory of Each Task

We present the scaling curves (on the final model checkpoints) and training trajectories of each linearity and breakthroughness task in Figure 17 and Figure 18. The evaluation of each task presents a large variance across the training steps. Though the tasks might present a breakthroughness pattern on the scaling curves, their trajectory curves show that language models pick up the task gradually.

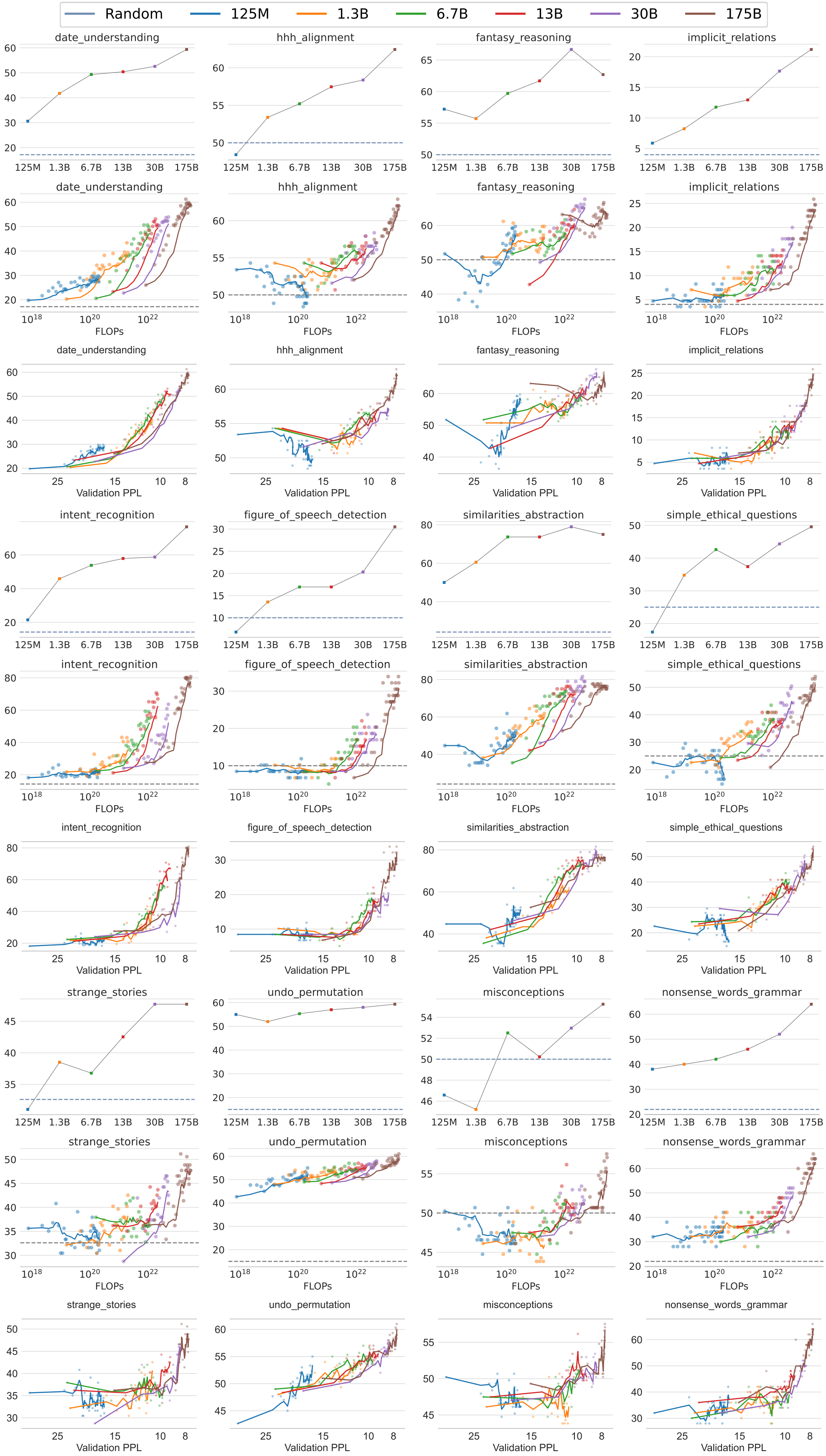


Figure 17: Scaling curves and trajectories of linearity tasks.

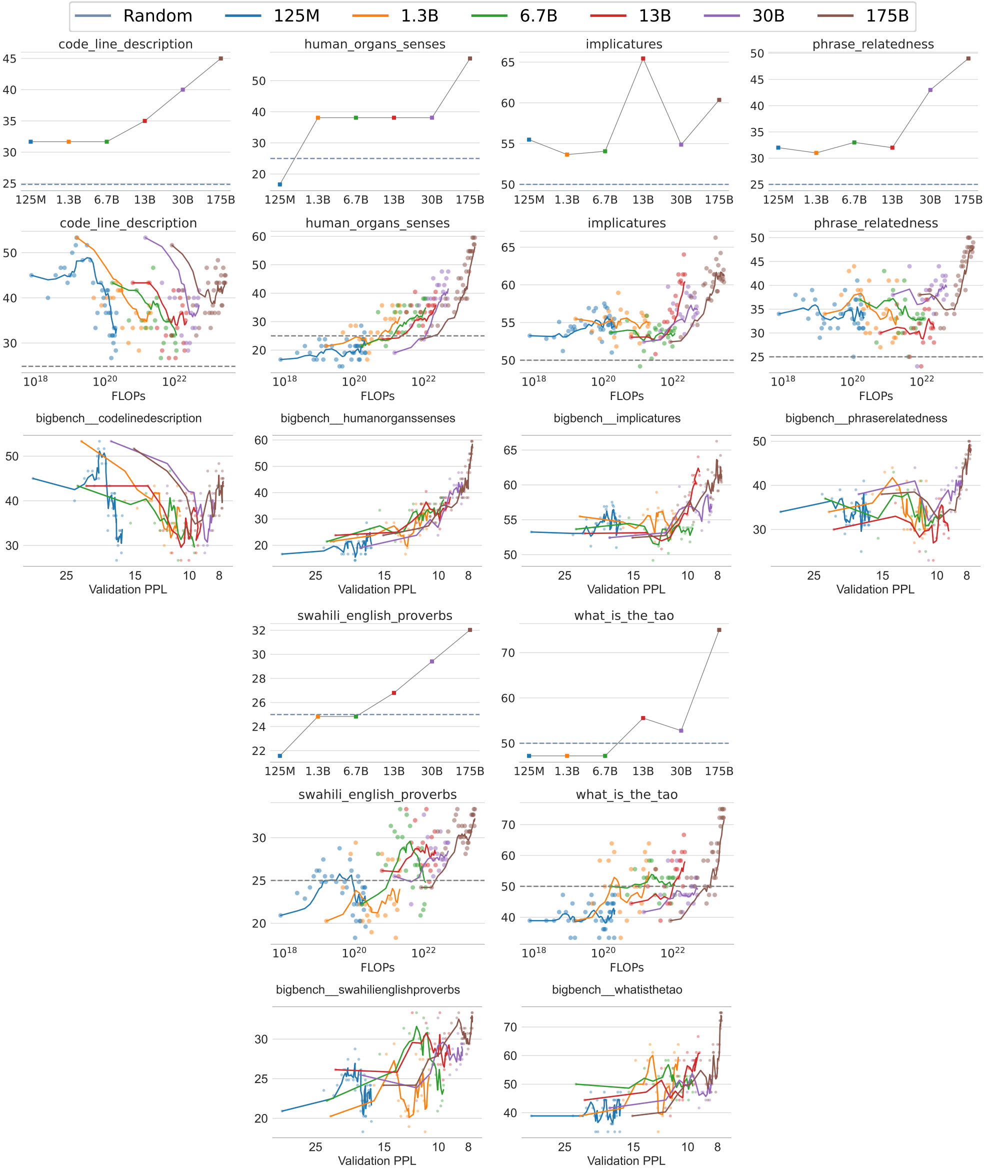


Figure 18: Scaling curves and trajectories of breakthroughness tasks.

Dist.	Greedy Search	Nucleus Sampling
	A girl (Lisbeth Salander) has ...	***A girl (Lisbeth Salander) has ...***
$p_s - p_l$	just discovered a new way to get her hair done!\n\nLisba is a blonde beauty who has been living her dream for quite some time now.\n\nLisba was recently spotted wearing a red and white wig and a black and blue striped suit that she wore in an Instagram video. The blonde beauty is currently living in the UK with her husband, Matt, with the help of their daughter Lizz.\n\nLizz and Matt have been dating for a while but have been dating	left the grandpa home he's lived to preserve for her.\nLisbin and her daughter Dylan Joanna (Arista Welch-Collinson) do everything they can to protect their sister.\nBut unlike ever before their young girl fills it all with grief with every form of loss.\nAs she learns Dylan isn't here anymore and acknowledges he's changed his mind, Daley finds herself falling back on the same old rules again.\nYellen's been teaching the inane Lyle
p_s	a crush on a guy (Lisbeth Salander) and she's not sure what to do about it.\n\nShe's a girl who's been in love with a guy (Lisbeth Salander) for a while, but she's not sure what to do about it.\n\nShe's a girl who's been in love with a guy (Lisbeth Salander) for a while, but she's not sure what to do about it.\n\nShe	just discovered\nthat it's not an actual\nworm! Thanks to the Halloween\nMirror campaign, she was discovered in the\ngoldeneye-buxco-only prologue of the main\ngame, for her interest in science and fantasia.\n\nMalcolm Young: There are four bugs on here.We're working on them, though, because they're incredibly busy and we're both making what are\nexpected\nto
$p_s + p_l$	a crush on a guy (Alexander Skarsgard) and he's a bit of a jerk.\nI think you mean Lisbeth Salander.\n\nI think you mean Lisbeth Salander.\nI think you mean Lisbeth Salander.\nI think you mean Lisbeth Salander.\nI think you mean Lisbeth Salander.\nI think you mean Lisbeth Salander.\nI think you mean Lisbeth Salander.\nI think you	a feeling that there's something more to computer games than what we see in them. Turns out, there is something more to computer games. But it's not "aliens," "moles," or "spies." It's about a virtual world that we can all be part of if we so choose.\n\nWe believe in calling that world Virtual Reality.\n\nThe game is called THE PREMIER.
p_l	a crush on a guy (Alexander Skarsgard) who is a computer hacker.\nThe Girl with the Dragon Tattoo?\nYup.<s>	has been attacked at her apartment by three men. They rape her and beat her. She later finds herself at a center for the mentally disabled. There, she has... See full summary »\nThe beginning of a story. The first nymphomaniac to admit that she has a problem, Helga refuses to seek treatment because she sees it as a surrender to defeat. However, it's... See full summary »\nA young man has just killed a drunken girl in a subway station.
$p_l - p_s$	sex for the first and last times in The Girl with the Dragon Tattoo.\nI don't remember that in the book.\nIt's implied. She says something like "I've never done this before" when they're having sex for the first time. And when she's having sex with her boyfriend for the last, she says something along the same lines.</s>	her face ripped open in The Girl with the Dragon Tattoo. That one still disturbs me. The third movie not as much, but that scene was still disturbing</s>

Table 5: Generated examples with greedy decoding and nucleus sampling under different configurations. The prompt is *A girl (Lisbeth Salander) has*.

Dist.	Greedy Search	Nucleus Sampling
	Now in private practice together, ...	***Now in private practice together, ...***
$p_s - p_l$	I'm going through the same process. I've never had any issues.\nI've had the same issue too! I'm trying not get into any of the practices because it's so hard. But now I can't do anything because of it :(I'm hoping to do something with my time and money to get some help!\nThat's really sad! Hopefully I can get some help! I hope to get some advice from someone who knows how to help me out, and that they	I can confirm it works pretty perfectly on My"EBM Used by me if I ever need\nGreat news :)</s>
p_s	I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan of the "I'm a big fan	a firm working on management strategies for retailing for software, designing, and engineering complex healthcare facilities, and leading multi-channel providers in addition to providing a variety of consulting services. Experience in all stages of PR is critical to have.\n\nThis 3.3 year-term contract includes a wide range of consultant training, including training for email and email integration, and three-way calls for projects.\n\nAt Microsoft Healthcare, we have a broad selection of technical leadership and support teams for our healthcare
$p_s + p_l$	I have the pleasure of working with a number of clients who have been referred to me by my colleagues. I have been able to help them with their legal issues and I have been able to help them with their personal issues.\n\nI have been able to help them with their legal issues and I have been able to help them with their personal issues.\n\nI have been able to help them with their legal issues and I have been able to help them with their personal issues.\n\nI have	Father Harry Thomas, a faculty member at Canisius College, and Father Christopher Cooney, pastor at Holy Redeemer Church in Lancaster, are a good team. The two have collaborated on two traditional healing classes for children since the spring of 2016. Their latest effort, followed by Father John Clifford, pastor at Christ the King Church in Canisius, has taken the call of mercy to the study level. Beginning September 24, Christ the King Church, Canisius, will host "Pope
p_l	Dr. David and Dr. David are a husband and wife team of chiropractors who specialize in the treatment of back pain, neck pain, headaches, and other musculoskeletal problems. They are dedicated to providing the highest quality of care to their patients in a comfortable, friendly, and professional environment.\n\nDr. David is a graduate of the Palmer College of Chiropractic in Davenport, Iowa. He has been practicing in the greater San Diego area since 1995. He	Spencer and Field with many years of combined practice are passionate about delivering high quality health care to the people of Texas. "Our mission is to empower you and your family to reach your health and wellness goals through nutritional and lifestyle changes. We take a whole-family approach to care and believe that true health is created from the inside out. If you're ready to feel better, we want to be part of your journey"</s>
$p_l - p_s$	Drs. Michael J. Gazzaniga and David A. Eagleman have written a new book that explores what they believe are some fundamental mysteries of the human mind. In The Brain: The Story of You, they argue that the brain is not just the seat of our thoughts and emotions but also of who we are as people.\n\nIn this excerpt from the introduction, the authors explain why they wrote the book and what they hope readers take away.\nThe Brain: The...</s>	the pair focus their legal expertise on helping immigrant families and individuals resolve a wide range immigration matters, including deportation defense, asylum, naturalization (citizenship), removal defense, consular processing (visas), VAWA petitions (domestic violence) as well as deportation and removal proceedings, appeals and motions before immigration court, administrative motions in immigration court, removal orders and waivers of inadmissability. Both attorneys are admitted to the Maryland State Bar as well as the District of Columbia Court of appeals

Table 6: Generated examples with greedy decoding and nucleus sampling under different configurations. The prompt is *Now in private practice together,*.

anachronisms	analogical_similarity	analytic_entailment
authorship_verification	causal_judgment	cause_and_effect
code_line_description	common_morpheme	conceptual_combinations
crash_blossom	crass_ai	cryobiology_spanish
dark_humor_detection	date_understanding	disambiguation_qa
discourse_marker_prediction	emoji_movie	empirical_judgments
english_russian_proverbs	entailed_polarity	entailed_polarity_hindi
evaluating_information_essentiality	fantasy_reasoning	figure_of_speech_detection
hhh_alignment	hinglish_toxicity	human_organs_senses
identify_math_theorems	identify_odd_metaphor	implicatures
implicit_relations	intent_recognition	international_phonetic_alphabet_nli
irony_identification	kannada	key_value_maps
known_unknowns	logical_args	logical_sequence
mathematical_induction	metaphor_boolean	metaphor_understanding
misconceptions	misconceptions_russian	moral_permissibility
movie_recommendation	nonsense_words_grammar	odd_one_out
penguins_in_a_table	periodic_elements	persian_idioms
phrase_relatedness	physical_intuition	physics
presuppositions_as_nli	riddle_sense	ruin_names
salient_translation_error_detection	sentence_ambiguity	similarities_abstraction
simple_arithmetic_json_multiple_choice	simple_ethical_questions	snarks
social_support	sports_understanding	strange_stories
suicide_risk	swahili_english_proverbs	symbol_interpretation
understanding_fables	undo_permutation	unit_interpretation
what_is_the_tao	which_wiki_edit	

Table 7: The list of multiple-choice tasks we use from BIG-Bench. Clicking the name of a task will direct you to the task's GitHub page.

date_understanding

Q: Yesterday, Jan 21, 2011, Jane ate 2 pizzas and 5 wings. What is the date tomorrow in MM/DD/YYYY?
A: 01/23/2011

Q: It is 4/19/1969 today. What is the date yesterday in MM/DD/YYYY?
A: 04/18/1969

Q: Yesterday was April 30, 2021. What is the date today in MM/DD/YYYY?
A:

Options: `05/01/2021,02/23/2021,03/11/2021,05/09/2021,06/12/2021`

nonsense_words_grammar

Q: How many things does the following sentence describe? The balforator, heddleilwilder and the sminniging crolostat operate superbly and without interrtulation.
A: 3

Q: How is the quijerinnedescribed in the next sentence? The umulophanitc quijerinne eriofrols the dusty grass.
A: umulophanitc

Q: Which word in the following sentence is a verb? The grilshaws bolheavened whincely.
A:

Options: `The, grilshaws, bolheavened, whincely`

entailed_polarity

Given a fact, answer the following question with a yes or a no.
Fact: Ed grew to like Mary. Q: Did Ed like Mary?
A: yes

Given a fact, answer the following question with a yes or a no.
Fact: They did not condescend to go. Q: Did they go?
A: no

Given a fact, answer the following question with a yes or a no.
Fact: The report was admitted to be incorrect. Q: Was the report incorrect?
A:

Options: `yes, no`

sentence_ambiguity

Claim: Delhi is not the only Hindi-speakingstate in India.
True or False? True

Claim: The population of the second-largest country in the world in 2021 exceeds the population of the third, fourth, and fifth largest countries combined.
True or False? True

Claim: Pescatarians almost never consume vegetarian food.
True or False?

Options: `True, False`

Table 8: Examples of prompts and answer options for four BIG-Bench multiple-choice tasks.

Linearity Tasks		
date_understanding	fantasy_reasoning	figure_of_speech_detection
hhh_alignment	implicit_relations	intent_recognition
misconceptions	similarities_abstraction	simple_ethical_questions
strange_stories	undo_permutation	nonsense_words_grammar
Breakthroughness Tasks		
code_line_description	human_organs_senses	phrase_relatedness
swahili_english_proverbs	what_is_the_tao	implicatures

Table 9: The list of linearity and breakthroughness tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 8

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
section abstract and section 1

☑ A4. Have you used AI writing assistants when working on this paper?
I used copilot to generate image captions and complete sentences throughout the paper, but all the generated texts have been heavily edited.

B ☑ Did you use or create scientific artifacts?

section 2

☑ B1. Did you cite the creators of artifacts you used?
section 2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We use internal data from the organization.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
section 2

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we use consists of a collection of open-sourced language modeling datasets, though the split is used internally, the contents should be largely observed by other researchers.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 2

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 2

C ☑ Did you run computational experiments?

section 3, 4, 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 2 and Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 3, 4, 5

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.