Demystifying Synthetic Data in LLM Pre-training: A Systematic Study of Scaling Laws, Benefits, and Pitfalls

Feiyang Kang[1,2,*], Newsha Ardalani[1], Michael Kuchnik[1], Youssef Emad[1], Mostafa Elhoushi[3,*], Shubhabrata Sengupta[4,*], Shang-Wen Li[1], Ramya Raghavendra[1], Ruoxi Jia[2], Carole-Jean Wu[1]

[1]FAIR at Meta, [2]Virginia Tech, [3]Cerebras Systems, [4]Independent consultant

[*]Work done at Meta. **Correspondence:** Feiyang Kang <fyk@vt.edu>, Newsha Ardalani <new@meta.com>

Abstract

Training data plays a crucial role in Large Language Models (LLM) scaling, yet high quality data is of limited supply. Synthetic data techniques offer a potential path toward sidestepping these limitations.We conduct a large-scale empirical investigation (>1000 LLMs with >100k GPU hours) using a unified protocol and scaling laws, comparing natural web data, diverse synthetic types (rephrased text, generated textbooks), and mixtures of natural and synthetic data. Specifically, we found pre-training on rephrased synthetic data *alone* is not faster than pre-training on natural web texts; while pre-training on 1/3 rephrased synthetic data mixed with 2/3 natural web texts can speed up 5-10x (to reach the same validation loss) at larger data budgets. Pre-training on textbook-style synthetic data *alone* results in notably higher loss on many downstream domains especially at small data budgets. "Good" ratios of synthetic data in training data mixtures depend on the model size and data budget, empirically converging to ~30% for rephrased synthetic data. Larger generator models do not necessarily yield better pre-training data than ~8B-param models. These results contribute mixed evidence on "model collapse" during large-scale single-round (n=1) model training on synthetic data–training on rephrased synthetic data shows no degradation in performance in foreseeable scales whereas training on mixtures of textbook-style pure-generated synthetic data shows patterns predicted by "model collapse". Our work demystifies synthetic data in pre-training, validates its conditional benefits, and offers practical guidance.

1 Introduction

The remarkable advancements in Large Language Models (LLMs) are closely tied to the scale and, critically, the quality of their training data. As computational demands for training state-of-the-art models escalate and the finite nature of high-quality natural text becomes increasingly apparent (Villalobos et al., 2024), significant interest has turned towards *synthetic data* (Ben Allal et al., 2024; Eldan and Li, 2023; Patel et al., 2024; Chen et al., 2024; Long et al., 2024; Thrush et al., 2024; Havrilla et al., 2024; Maini et al., 2024; Li et al., 2023b; Abdin et al., 2024; Javaheripi et al., 2023; Cheng et al., 2024; Gu et al., 2023). Defined as text generated by pre-existing models or automated pipelines, synthetic data presents a compelling potential avenue for augmenting—or perhaps eventually replacing—traditional human-generated corpora during the foundational pre-training phase.

While the utility of synthetic data is increasingly established in post-training stages like instruction-tuning and alignment (Taori et al., 2023; Li et al., 2023a; Ge et al., 2024)—where objectives are targeted and natural data may be scarce—its role and effects during the crucial pre-training phase remain largely uncharacterized and poorly understood (Liu et al., 2024b). This knowledge gap represents a significant barrier to optimizing LLM development pipelines and motivates fundamental questions:

(RQ1) Can synthetic data effectively enhance LLM pre-training performance at large data scales and under what conditions?

(RQ2) How do different types and generation methodologies for synthetic data influence pre-training dynamics and scaling behavior?

(RQ3) What principles guide the effective deployment of synthetic data in pre-training, including "good" mixture ratios, the impact of generator model capabilities, and the statistics of the training corpus?

Despite the straightforward nature of these questions, clear answers remain elusive. This **ambiguity** stems from several factors. Firstly, the landscape is marked by *inconsistent empirical findings and considerable methodological heterogeneity* (Long et al., 2024; Liu et al., 2024b). Proposed

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 10751–10770
November 4-9, 2025

approaches often rely on bespoke setups, obscuring direct comparability and generalizability. Even for simple open-sourced methods, Yang et al. (2024) reports models trained on synthetic data from Maini et al. (2024) saturates early on in continued pre-training without much performance gain. Secondly, synthetic data generation involves *complex tradeoffs between targeted quality enhancement and broad distributional diversity*. Recent studies present a contradiction regarding synthetic data: while some argue it improves training data quality at the expense of diversity (Havrilla et al., 2024), others suggest that diversity itself is a key predictor of model performance (Chen et al., 2024). Thirdly, *theoretical concerns persist*, notably "model collapse" from recursive training (Dohmatob et al., 2024b,a), even if catastrophic failures are not yet widespread. This confluence of **potential benefits, inconsistent evidence, methodological variance, and theoretical risks** underscores a critical need for systematic investigation.

To address this critical gap and provide empirically grounded answers , we undertake a systematic, large-scale investigation into the role and effective use of synthetic data in LLM pre-training. Our study involves training over 1000 LLM variants (up to 3B parameters) on datasets comprising up to 200B tokens, utilizing over 100,000 GPU hours, enabling evaluation on the effect of model size and data regimes in its scaling laws.

Our principal findings reveal that:

1. Strategically incorporating specific synthetic data types can **significantly accelerate pre-training convergence**. Compared to pre-training on natural web texts, training on 1/3 rephrased synthetic data mixed with 2/3 natural web texts can speed up 5-10x (to reach the same validation loss) at larger data budgets.
2. However, the impact is **highly dependent on the synthetic data's type and characteristics**: Pre-training on rephrased synthetic data alone is not faster than pre-training on natural web texts; whereas pre-training on textbook-style synthetic data alone results in notably higher validation loss.
3. "Good" ratios of synthetic data in training data mixtures are nuanced, varying with data type, target model scale, and budget, converging to $\sim$ 30% for rephrased synthetic data. Counter-intuitively, **larger or more capable generator models do not necessarily yield superior synthetic data** than $\sim$8B-param models for pre-training downstream models.
4. We interpret the results with a focus on low-level statistics. Some unigrams that are frequent in test datasets but rare or absent in training datasets result in higher evaluation loss, whereas no single training set offers complete coverage. CommonCrawl has wider unigram coverage and the lowest KL-divergence to test datasets; however, it did not yield superior performance, suggesting **"good" training data mixtures depend on factors beyond simple similarity and pointing to more complex diversity-quality trade-offs.**

2 Related Work

Our research intersects with several key areas in LLM development, particularly concerning the generation and use of synthetic data for pre-training, data mixture strategies, the application of scaling laws, and concerns around model collapse.

Synthetic Data in LLM Pre-training The utility of synthetic data is well-recognized in targeted later stages of training, such as instruction tuning (Taori et al., 2023), alignment (Li et al., 2023a; Ge et al., 2024), and increasingly for enhancing reasoning capabilities (Muennighoff et al., 2025). Meta (2025) detail a dedicated "mid-training" stage using synthetic reasoning data, occurring after initial pre-training and prior to subsequent post-training with reinforcement learning (RL). In contrast, synthetic data's role in foundational pre-training for general capabilities is less established and characterized by varied approaches. The Phi series (Li et al., 2023b; Javaheripi et al., 2023) pioneered the use of "textbook-style" synthetic data for pre-training production-grade models. Abdin et al. (2024), discussing later Phi models (e.g., Phi-4), argue this approach particularly boosts reasoning with large training budgets where natural web text offers diminishing returns, while also acknowledging potential downsides like limited factual grounding and increased hallucination risks. Other foundational pre-training explorations include Eldan and Li (2023)'s story generation for smaller models, rephrasing existing texts (Maini et al., 2024), and employing diverse prompts for generation (Chen et al., 2024; Patel et al., 2024; Gu et al., 2023). Despite these explorations (see survey by Havrilla et al. (2024)), the landscape is characterized by methodological

heterogeneity and sometimes conflicting outcomes (e.g., Long et al. (2024) and Liu et al. (2024b) on diversity and quality; Yang et al. (2024) on saturation with rephrased data from Maini et al. (2024)). Our study differentiates itself by systematically evaluating multiple distinct synthetic data generation paradigms (rephrased web text, generated "textbooks") and their mixtures with natural data under a unified pre-training protocol and rigorous scaling law analysis across substantial data and model scales, aiming to clarify these ambiguities.

Data Curation, Mixing Strategies, and Scaling Laws Meticulous data curation and strategic mixing of diverse natural data sources are established as critical for LLM pre-training (Touvron et al., 2023; Raffel et al., 2020; Penedo et al., 2024; Xie et al., 2023; Ye et al., 2024; Liu et al., 2024a). However, the systematic integration and scaling behavior of *synthetic* data with natural corpora remain comparatively underexplored, despite promising initial findings suggesting benefits from such mixtures (Maini et al., 2024; Javaheripi et al., 2023). Seminal scaling laws describe predictable relationships between LLM performance and factors like model size, dataset size, and compute (Kaplan et al., 2020; Hoffmann et al., 2022) and have recently been extended to model natural data mixing strategies (Kang et al., 2024b). The scaling dynamics of pre-training specifically with synthetic data have been described as "mysterious" (Liu et al., 2024b). For instance, models trained on certain synthetic data types can exhibit early performance saturation (Yang et al., 2024), highlighting the need for a clearer understanding. Our work distinctively addresses these gaps by employing scaling law analysis as a primary *evaluative tool*. We systematically investigate optimal mixture ratios of different synthetic data types with a natural web text baseline, examining how these ratios and overall pre-training effectiveness interact with synthetic data characteristics and varying data budgets. This approach aims to demystify the role of synthetic data in scalable LLM pre-training and provide empirically grounded guidance for its effective integration.

Model Collapse and Generational Degradation The prospect of training models predominantly on model-generated data has spurred theoretical investigations into "model collapse" or "generational degradation," where recursive training might lead to a decline in model quality due to reduced diversity or amplified biases (Shumailov et al., 2023; Dohmatob et al., 2024b,a). While these risks are highlighted in theoretical analyses and simulations, large-scale empirical evidence from practical pre-training scenarios, especially those still incorporating significant natural data, remains limited. Our study contributes direct empirical insights on "model collapse" during large-scale single-round (n=1) training on synthetic data by pre-training models on substantial datasets with varying proportions and types of synthetic data.

3 Synthetic Data Generation Methods

We investigate two distinct paradigms for generating synthetic data: web rephrasing and textbook-style pure synthetic data. These paradigms represent different philosophies for augmenting or replacing natural text in pre-training.

3.1 Web Rephrasing

Inspired by techniques such as WRAP (Maini et al., 2024), web rephrasing leverages a pre-trained Language Model (LM) to refine existing web documents into a potentially more valuable pre-training resource. In our study, we implemented web rephrasing by sampling documents from the CommonCrawl dataset. A pre-trained generator LM was prompted to rewrite these documents. Drawing inspiration from variants explored in Maini et al. (2024), we generated two distinct styles intended to probe different potential benefits for pre-training:

- *High-Quality (HQ) Rephrasing:* Prompts instructed the generator model to rewrite the source text into clear, coherent, well-structured English, mimicking the style often found in high-quality sources like Wikipedia. This targets improving general text quality for foundational pre-training, akin to an aggressive data filtering or quality enhancement step. HQ rephrasing aims to increase the effective density and quality of information already present within the web corpus, aligning with the broader goal of improving data efficiency for pre-training.
- *Question-Answering (QA) Rephrasing:* Prompts instructed the generator model to restructure the source text's information into a conversational question-answering format. This explores incorporating instruction-following or dialogue-like structures directly into the pre-training phase, potentially accelerating the development of alignment capabilities. This QA rephrasing approach relates to the growing interest in 'instruction pre-

training' (Cheng et al., 2024), where downstream objectives like instruction-following or dialogue capabilities are incorporated early via synthetic data formatting.

3.2 Synthetic Textbooks (TXBK)

This paradigm is driven by the hypothesis that dense, high-quality, educational content might be more compute-efficient for instilling certain capabilities (e.g., reasoning, coding, factual recall) compared to diffuse web text. The goal is to generate entirely novel content that mimics the structure, style, and information density of textbooks or high-quality educational materials. For our experiments, we generated novel "textbook-style" documents. The generation process was seeded using keywords *randomly sampled* from CommonCrawl to provide diverse starting points for various topics. A pre-trained generator LM (e.g., Mistral-7B) was then prompted using structured instructions to produce text resembling textbook chapters or tutorials on the seeded topic. These prompts explicitly encouraged the generation of clear explanations, definitions, illustrative examples (including code snippets with explanations where relevant), and potentially associated exercises or reasoning steps. Throughout the generation process, an emphasis was placed on striving for factual accuracy, coherence, and a clear pedagogical structure.

4 Empirical Results

4.1 Experimental Setup

We conduct large-scale pre-training experiments comparing models trained on: (1) a natural web corpus baseline, (2) purely synthetic datasets generated using our distinct Web Rephrasing and Synthetic Textbook paradigms (see Section 3 for generation methodologies), and (3) various mixtures of natural and synthetic data. Approximately 600 LLM variants, with sizes up to 3 billion parameters, were trained on datasets of up to 200 billion tokens. This effort consumed over 70,000 GPU hours on NVIDIA A100 80G hardware.

4.1.1 Datasets

Natural Data Baseline: Our natural data consists of English text sourced from unfiltered CommonCrawl (CC) dumps, processed via the RedPajama-v2 pipeline (Weber et al., 2024).

Synthetic Data: All synthetic datasets were generated using a Mistral-Instruct-7b-v0.1 model (Jiang et al., 2023), with input documents for rephrasing or seeding sampled from our unfiltered CC baseline. Standard generation sampling parameters and light heuristic post-filtering were applied. Generation details, prompt templates, and sample generations are provided in Appendix B.3.

The following synthetic types were produced:

- *Web Rephrasing (Maini et al. (2024)-like):* Generated by rephrasing CC documents using prompts optimized from Maini et al. (2024) to produce longer texts in two styles: **High-Quality (HQ)** and **Question-Answering (QA)**.
- *Synthetic Textbooks ((Li et al., 2023b)-like):* Novel multi-chapter **"textbooks" (TXBK)** generated from CC-derived outlines, employing varied prompts targeting different audiences to encourage diversity. Each chapter averaged ∼450 tokens and often included exercises.

Training Data Mixtures: For each synthetic data type (HQ, QA, or TXBK), we prepared datasets by concatenating and shuffling source data under these conditions: 100% Natural (Unfiltered CC baseline); 100% Synthetic (consisting entirely of one synthetic type: HQ, QA, or Textbook); 67% Synthetic / 33% Natural; 33% Synthetic / 67% Natural. For each experimental point (defined by model size and data budget), models were trained on these different mixtures, enabling direct comparison. $5+$ model variants were trained per condition for robust scaling law analysis.

4.1.2 Models, Training, and Evaluation

We use a standard decoder-only Transformer architecture based on Llama 3 (Grattafiori et al., 2024), with model sizes ranging logarithmically from 100M to 3B parameters. All models were trained from scratch using the Meta Lingua library (Videau et al., 2024) on PyTorch (Paszke, 2019). Following the line of research on scaling laws for LLMs (Kaplan et al., 2020), we define the size of all models being trained as their count of *non-embedding parameters*, which are *learnable parameters* in the model except for those associated with the input and output token embeddings. A consistent training regime was applied for fair comparison, including a cosine learning rate schedule (10% warmup), a context length of 4096 tokens, an effective batch size of 1M tokens, and the Llama 3 pre-trained TikToken tokenizer (128k vocabulary) (Grattafiori et al., 2024). The primary performance measure is per-token average perplexity

(cross-entropy loss) calculated on a held-out diverse set of 14 non-code/math English text domains from The Pile (Gao et al., 2020) and the Wikitext-103 dataset (Merity et al., 2016), evaluated at the final checkpoint. Complete details are provided in Appendix A.2.

4.1.3 Data Scaling

For a fixed model size (1B parameters), data scaling is modeled as: $\hat{\mathcal{L}}(D) = \frac{B}{D^{\beta}} + E$, where $\mathcal{L}$ is validation loss, D is training data budget, and B, β, E are fitted coefficients.

We trained 1B-parameter models on various data mixtures with data budgets from 1B up to 200B tokens. The scaling formula was fitted using data points up to 100B tokens; predictions were then validated on runs trained with 200B tokens. Six data mixtures (CommonCrawl, 33% HQ + 67% CC, 33% QA + 67% CC, Textbook (TXBK), 67% TXBK + 33% CC, and 33% TXBK + 67% CC) were trained to 200B tokens for this validation, as our HQ and QA synthetic datasets were limited to 100B tokens each. The fit demonstrated high precision, as shown in Fig. 6 (deferred to Appendices), achieving a low Relative Mean Absolute Error (RMABE) of 0.41% when predicting for 200B tokens.

With reasonably reliable fits validated, we extrapolated data scaling (fitted up to 100B tokens) to predict loss for larger data budgets (up to 8T tokens), covering training regimes of state-of-the-art LLMs (Meta, 2025; DeepSeek-AI, 2024). Key findings are presented in Fig. 1 and summarized below:

1. **Pure synthetic data is not superior to CommonCrawl (CC):** Training solely on HQ or QA synthetic data does not significantly outperform training only on CC. Training only on TXBK performs notably worse than trainng on CC.
2. **Mixtures outperform pure synthetic types:** Mixing any synthetic data type with CC substantially improves performance over using that synthetic type alone.
3. **Rephrased data mixtures show low sensitivity to ratio (33% vs. 67% synthetic):** For HQ and QA, both 33% and 67% synthetic mixtures with CC yield similar performance.
4. **Textbook mixtures favor less synthetic data:** For TXBK, a 33% synthetic mixture significantly outperforms a 67% mixture. The 33% TXBK mixture surpasses pure CC performance after ∼20B tokens, while the 67% TXBK mixture underperforms pure CC.

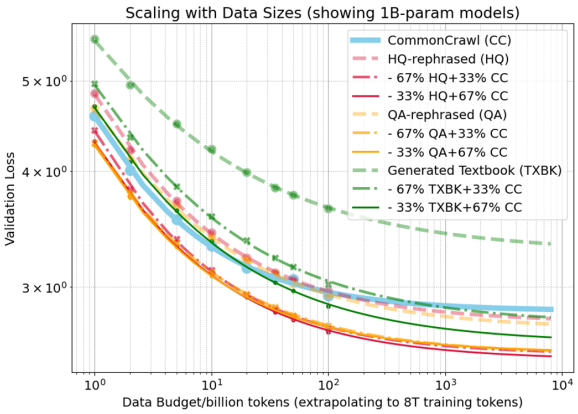


Figure 1: Extrapolated data scaling performance for 1B-parameter models across various data mixtures. (Fitted coefficients can be found in Table 4 in Appendices.)

4.1.4 Model Scaling

For a fixed data budget (50B tokens), model scaling is modeled as: $\hat{\mathcal{L}}(N) = \frac{A}{N^{\alpha}} + E$, where N is model parameter size, and A, α, E are fitted coefficients.

We trained models from 100M to 3B parameters for 50B tokens on all 10 data mixtures. The formula was fitted using models up to 2B parameters and validated on 3B-parameter models. This fit also proved highly precise (Fig. 7, deferred to Appendices), with an RMABE of 0.30% for 3B-parameter model predictions. Extrapolating model scaling (fitted with models up to 3B parameters) to predict performance for larger models (up to 200B parameters) on a 50B token budget (Fig. 2) revealed several differences from data scaling patterns:

1. Pure synthetic data remains non-advantageous over CC; notably, models trained on **pure rephrased synthetic data will underperform those trained on CC at larger models.**
2. For rephrased data mixtures, sensitivity to the mix ratio changes: while a 67% synthetic mix was marginally better for larger *data budgets* (data scaling), it becomes marginally *disadvantageous* for larger *model sizes* (model scaling) compared to a 33% mix.
3. For TXBK mixtures, 33% synthetic consistently outperforms 67%. The advantage of 33% TXBK over pure CC **appears to diminish with larger models**, a trend not observed in data scaling.

Overall, these model scaling results suggest synthetic data appears comparably less favorable for pre-training larger LMs relative to its utility

in data scaling scenarios. Despite outperforming training on CC, larger models are not as tolerant to a higher ratio synthetic data as larger data budgets. This observation aligns with practices where synthetic data is effective for smaller LMs or specific pre-training phases, but less predominantly used for the largest models.

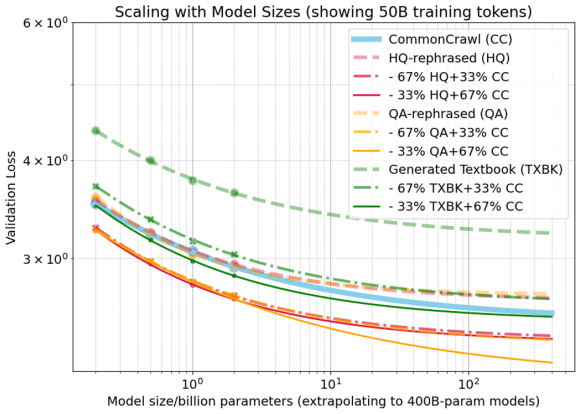


Figure 2: Extrapolated model scaling performance for training on 50B tokens across various data mixtures. (Fitted coefficients are provided in Table 5.)

4.1.5 Compute Scaling and Irreducible Loss

We also fit joint scaling laws incorporating both model size (N) and data budget (D) using data from all ∼700 training runs (details in Appendix A): $\hat{\mathcal{L}}(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E$. An example loss landscape for CC data is shown in Fig. 8. The coefficient E represents the *irreducible loss*—the theoretical minimum loss achievable with infinitely large models and data.

Estimations of E for each data mixture (Fig. 3) indicate their ultimate potential. Notably, **any mixture involving synthetic data, or pure synthetic data (except pure QA), is projected to achieve a lower irreducible loss than training only on CommonCrawl.** This empirically challenges theoretical concerns about "model collapse" in single-round training, which predict any synthetic data inclusion would ultimately degrade performance (Dohmatob et al., 2024a). Among the studied mixtures, **33% HQ rephrased data + 67% CC shows the lowest projected irreducible loss.** Conversely, pure QA rephrased data exhibits a high irreducible loss, second only to pure CommonCrawl.

5 Additional Studies: A Broader View

Beyond the primary scaling law analysis, we conduct targeted experiments to deepen our understanding of specific factors influencing the effective use of synthetic data in pre-training.

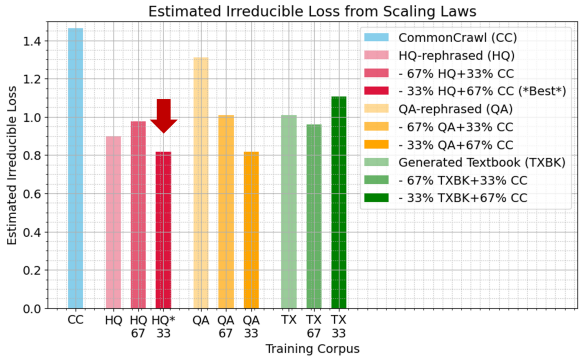


Figure 3: Estimated irreducible loss (E) for different data mixtures. Lower values are better.

5.1 "Good" Synthetic Data Mixture Ratios

Motivation Our main scaling law analysis tested limited discrete mixture ratios (0%, 33%, 67%, 100%) of synthetic and natural data. To identify "good" ratios with finer granularity, we performed a fine-grained grid search, motivated by indications that optimal mixtures vary with synthetic data type, model scale, and data budget.

Methodology We trained approximately 400 additional LLMs (200M to 1B parameters) on data budgets from 1B to 50B tokens. For each synthetic data type (HQ, QA, TXBK) and each (model size, data budget) configuration, we varied the percentage of synthetic data mixed with CommonCrawl across ten exponentially spaced points: 0%, 0.5%, 1%, 2%, 5%, 10%, 15%, 20%, 50%, and 100%. The "good" ratio was defined as the mixture yielding the lowest validation loss on the evaluation sets.

Findings Figure 4 visualizes the results. Best-found ratios are all below 50% appear to converge ∼30%. Key observations include:

- **HQ Rephrased Data:** The optimal mixture is consistently ∼**30% HQ synthetic data** with 70% CommonCrawl across tested scales. This 30% mixture generally outperformed the 50% mixture suggested by Maini et al. (2024) in our setups.
- **QA Rephrased Data:** The preferred ratio of QA data tends to **decrease** with increasing model/data sizes, shifting from ∼50% for smaller configurations towards 30% for larger ones.
- **Textbook (TXBK) Data:** Benefits are most apparent at larger scales. Optimal ratios are often minimal (<5%) for smaller configurations, in-

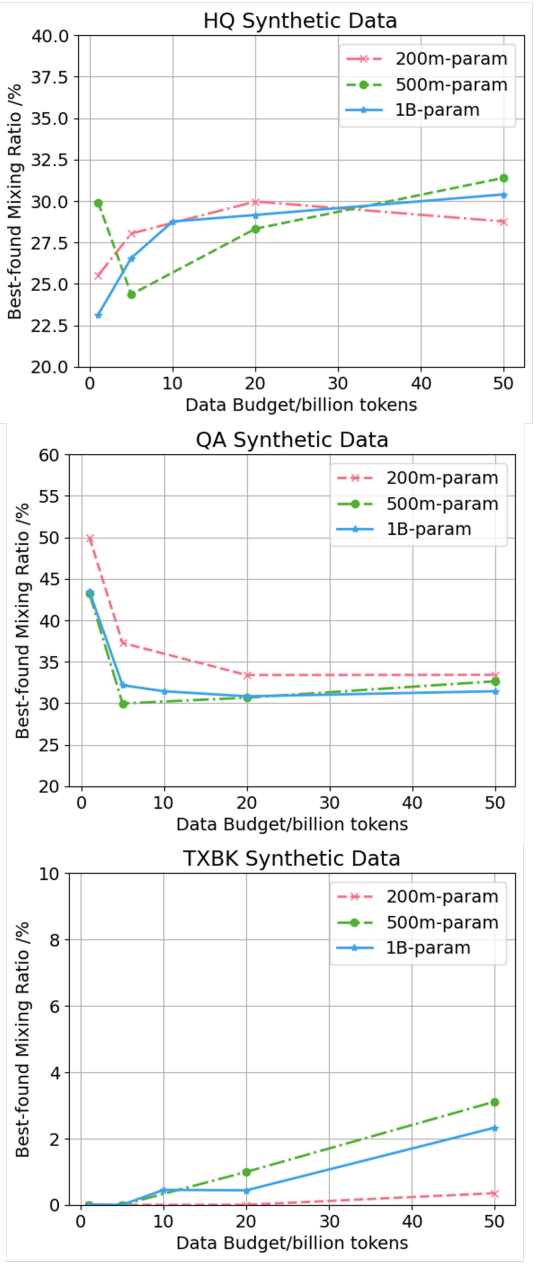


Figure 4: Best-found mixture ratios (percentage of synthetic data with CommonCrawl) from grid search for HQ, QA, and TXBK data types across different model sizes and data budgets. Best-found ratios are all below 50% appear to converge $\sim$ 30%.

creasing with scale but generally remaining below those for rephrased data.

These findings refine our scaling law observations, emphasizing the sensitivity of effective synthetic data deployment to its type and the training regime.

5.2 Impact of Generator Model Capability

Motivation It is often assumed that larger, more capable generator models produce higher-quality synthetic data, leading to better downstream performance. We empirically tested this hypothesis.

Methodology We used Llama-3 models of varying scales (3B, 8B, and 70B parameters) as generators to recreate subsets of our HQ (High-Quality) and QA (Question-Answering) rephrased datasets. Generation prompts and source CommonCrawl documents were kept consistent with our original pipeline, which utilized Mistral-7B-Instruct as the generator. A fixed 1B-parameter downstream model, with the same architecture as in previous experiments, was then trained for 5 billion tokens. For each generator (Llama3-3B/8B/70B), we evaluated the synthetic data produced by training the downstream model on mixtures with CommonCrawl. The percentage of synthetic data in these mixtures was varied across eight exponentially spaced points: 0.5%, 1%, 2%, 5%, 10%, 15%, and 20%. Approximately 200 models were trained for this ablation study to compare the efficacy of synthetic data generated by models of different capabilities.

Findings The results, illustrated by trends similar to those shown in Figure 5 (which would now represent these detailed mixture evaluations), challenge the "bigger is always better" intuition for generator models and reveal a nuanced relationship:

- A certain level of generator capability appears beneficial: synthetic data from Llama-3-8B generators consistently outperformed data from Llama-3-3B generators. **This finding suggests a baseline capability is necessary** and contrasts with suggestions from Maini et al. (2024) that rephrasing costs could be significantly reduced by using smaller generator LMs without a loss in downstream performance.
- However, **increasing generator size further to Llama-3-70B did not yield superior synthetic data** for pre-training compared to data from the Llama-3-8B generator, when assessed by the trained model's validation loss.
- In specific instances, the Llama-3-70B generator proved less effective. **For HQ rephrased data, synthetic data generated by Llama-3-70B models led to consistently worse** evaluation results than data from Llama-3-8B models. For QA rephrased data, the 70B generator's output resulted in comparable performance than that from the 8B generator.

This suggests that factors beyond sheer generator scale—such as instruction following fidelity at different scales, the diversity of generated outputs, or potential introduction of stylistic artifacts—play a

crucial role in determining the utility of synthetic data for pre-training. Simply employing the largest available generator may not be the most effective or efficient strategy.

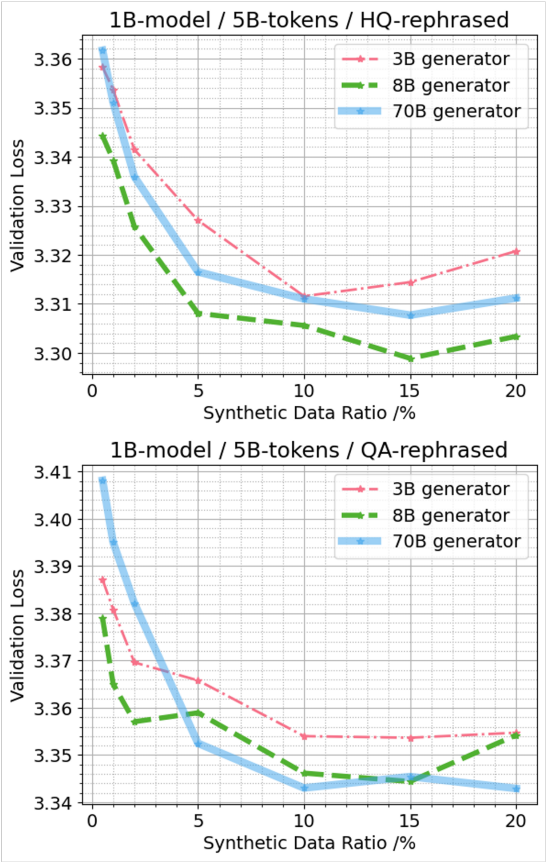


Figure 5: Generator model capability ablation. Compares validation loss of 1B-parameter models trained for 5B tokens using mixtures of HQ/QA rephrased data from Llama3-3B/8B/70B generators with CommonCrawl. The percentage of synthetic data in these mixtures was varied across seven exponentially spaced points from 0.5% to 20%.

5.3 Interpretation via Low-level Statistics

The impact of synthetic data on pre-training efficiency, particularly how "good" mixing ratios vary with synthetic data type, budget, and model size, necessitates investigation into underlying mechanisms. While synthetic data may improve "data quality" (e.g., coherence, reduced noise) at the cost of diversity (Long et al., 2024), the generation process reflects the generator LM's output distribution, potentially shrinking the support of distribution from natural expressions or introducing artifacts like model collapse (Dohmatob et al., 2024a).

We investigate via low-level statistical analysis:

(Q1) Does synthetic data exhibit reduced lexical diversity (i.e., a "shrunk support") compared to natural web text?

(Q2) Can improvements in test performance with synthetic data be attributed to smaller training-test distributional distance?

(Q3) Are optimal mixing ratios due to minimized distributional distance or a more complex diversity-quality trade-off?

Inspired by Magnusson et al. (2024)'s finding that a small fraction of high-frequency strings contributes significantly to LM loss, we conduct unigram frequency analysis across training and test corpora.

Full results and analyses are provided in Appendix 5.3. We summarize the key findings:

- **Vocabulary Mismatches and High-Loss Tokens:** Unigrams frequent in test sets but rare or absent in some training sets (e.g., "\n\n", " hvor" (Danish), " dön" (Turkish), etc.) cause higher evaluation loss. This issue is pervasive; no single training set offers complete coverage.
- **Synthetic Data and Unigram Distribution:** Synthetic data slightly shrinks unigram distribution compared to broad web corpora like CommonCrawl—yet CommonCrawl's wider coverage did not yield superior performance.
- **Distributional Distance to Test Sets:** KL-divergence on unigram distributions did not show synthetic data closer to test distributions; CommonCrawl appears closest to test datasets.

Preliminary conclusions are:

1. **Inherent Limitations of Single Data Sources:** All data sources, including CommonCrawl, have distributional gaps causing high evaluation losses on underrepresented tokens. This favors mixed corpora with broad lexical coverage with reasonable frequencies, helping explain why mixed-source corpora often outperform single-source ones.
2. **Beyond Distributional Matching for "Good" Mixtures:** Models often train best with significant synthetic data proportions (e.g., ≈30%) even if it does not minimize unigram distributional distance to test sets. This suggests factors beyond simple similarity, pointing to more complex diversity-quality trade-offs.

6 Conclusion

This large-scale empirical investigation (over 1000 LLM variants) demonstrates that synthetic data in foundational pre-training presents a nuanced trade-off. Strategically mixing specific synthetic types (e.g., $\sim$ 30% high-quality rephrased text

with natural data) can significantly accelerate pre-training convergence up to 5-10x and potentially achieve lower irreducible loss than natural data alone. These results contribute mixed evidence on "model collapse" during large-scale single-round (n=1) model training on synthetic data–training on rephrased synthetic data shows no degradation in performance in foreseeable scales whereas training on mixtures of textbook-style pure-generated synthetic data shows patterns predicted by "model collapse". However, effectiveness is also conditional on generation method, mixture strategy, and generator models. Larger generator models did not guarantee superior pre-training data. Downstream model performance cannot be simply explained by training data's diversity or similarity to test corpora, but pointing to more complex diversity-quality tradeoffs.

Our findings underscore that synthetic data requires careful, empirically-informed deployment, rather than being a universal solution to data constraints. Essential **next steps** involve developing more targeted synthetic data generation techniques and dynamic mixing strategies. Rigorous evaluation of their long-term impacts on diverse model capabilities (reasoning, robustness, alignment) at frontier scales is crucial, alongside pinpointing key beneficial characteristics of generator models beyond sheer size.

7 Discussions

On "Model Collapse" for Large-Scale Single-Round (n=1) Model Training on Synthetic Data. This paper contributes new evidence on large-scale single-round (n=1) model training on synthetic data, rejecting certain conjectures from prior research on "model collapse" and helping refine their range of application.

The notion, "model collapse", was formalized by Shumailov et al. (2023), characterizing the effect of iterative training on self-generated (or mixed) data. Subsequent works such as Dohmatob et al. (2024b) studies the effect for n-fold iterative synthetic training, where the main results show training on synthetic data even for n=1 (one-round) leads to significant flare-up in test perplexity compared to training on the original data. Further, Dohmatob et al. (2024a) shows that even the smallest fraction of synthetic data (e.g., as little as 1% of the total training dataset) can still lead to model collapse while training 124M-parameter GPT-2 small on BabyStories. Based on theoretical derivations, the authors conjecture that larger "models may mitigate the collapse, although they do not entirely prevent it." With strong conclusions on the important topic, the theoretical analysis is based on stylized models (e.g., regression models) and the language modeling experiments are simplistic (e.g., fine-tuning for one task). There remains a significant gap between these forecasts from "model collapse" and the advancement in generation/training on synthetic data.

This work brings more clarity to this evolving topic. In this work, we found that for one-round (n=1) model training on synthetic data:

- when using rephrased synthetic data in pre-training contemporary LMs, we do not see patterns of degradation in performance in foreseeable scales, and pre-training on rephrased synthetic data mixed with natural data can lead to significant speed-up in reducing validation loss.
- training on mixtures of textbook-style pure-generated synthetic data did lead to notably higher loss on downstream domains, especially at small data budgets. This is largely consistent with the patterns and predictions reported in the "model collapse" papers.

This shows that in large-scale LM pre-training, training on synthetic data for one-round does not necessarily degrade validation performance, confining the extrapolation of theoretical results from "model collapse" papers.

- Despite the shrinking support on n-gram distributions, with the right type of synthetic data and a mixing ratio with natural data, the benefits for including synthetic data could outweigh the issues from "model collapse" and deliver substantial benefits. This adds counter evidence to the conjecture that including synthetic data would always lead to worse model performance.
- However, most benefits observed are from rephrased synthetic data, whereas textbook-style synthetic data often leads to performance degradation even when mixed with a large proportion of natural data. Empirical results on textbook-style synthetic data show patterns characterized in "model collapse", suggesting the generalizability of theoretical results in "model collapse" may depend on the nature of synthetic data.

Together, results and findings contributed in this work reject certain claims from "model collapse" and help refine their range of application.

Limitations

Our study, while extensive, has limitations influencing the scope and generalizability of findings:

- **Limited Scope of Synthetic Data:** We analyzed three specific synthetic data types (HQ/QA rephrased, TXBK). Findings may not directly apply to other generation methods (e.g., synthetic code, dialogues) or prompting strategies.
- **Evaluation Focus:** Analysis heavily relied on perplexity/loss for scaling. It lacked in-depth human evaluations for nuanced capabilities or safety, and assessment on highly specialized tasks. Additional evaluations on NLP benchmarks would be a desirable addition as loss is not our final goal.
- **Temporal Effects:** We examined a single pre-training stage. Potential long-term effects, subtle degradation, or multi-generational dynamics ("model collapse") were not investigated.
- **Scale Constraints:** Experiments reached 3B parameters and 200B tokens. Scaling trends observed regarding synthetic data utility at larger model sizes require validation at frontier model scales (>100B parameters, >10T tokens).
- **Impact of Tokenizers:** Studies in Section 5.3 show that different training datasets have different coverage of tokens, where certain tokens rare in the training data may be associated with a higher loss in evaluation. Though not significant enough to affect the main results in this paper (such as the "good" mixing ratios), the impact of tokenizer may become more visible in finer-grained analysis on validation loss.

Ethical Considerations

The generation and use of synthetic data in LLM pre-training warrant careful ethical reflection:

- **Bias Propagation:** Synthetic data risks inheriting and amplifying biases from generator models. Auditing generators and generated data for fairness is crucial but was outside this study's scope.
- **Factual Accuracy:** Generated content can include inaccuracies (hallucinations). Large-scale use could embed misinformation in models, necessitating robust quality control.
- **Data Diversity:** Over-reliance on potentially homogeneous synthetic data could reduce model robustness and diversity compared to training on varied real-world text.
- **Transparency & Reproducibility:** We mitigate some concerns by committing to open-sourcing our full recipe to facilitate reproducibility and further community research.

Acknowledgment

Ruoxi Jia and the ReDS lab acknowledge support through grants from the Amazon-Virginia Tech Initiative for Efficient and Robust Machine Learning, and the National Science Foundation under grants IIS-2312794, IIS-2313130, OAC-2239622.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. 2024. On the diversity of synthetic data and its impact on training large language models. *arXiv preprint arXiv:2410.15226*.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. *arXiv preprint arXiv:2406.14491*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. 2024a. Strong model collapse. *arXiv preprint arXiv:2410.04840*.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. 2024b. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. *arXiv preprint arXiv:2305.09137*.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, and 1 others. 2024. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *arXiv preprint arXiv:2412.02980*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Feiyang Kang, Hoang Anh Just, Anit Kumar Sahu, and Ruoxi Jia. 2023. Performance scaling via optimal transport: Enabling data selection from partially revealed sources. *Advances in Neural Information Processing Systems*, 36:61341–61363.

Feiyang Kang, Hoang Anh Just, Yifan Sun, Himanshu Jahagirdar, Yuanzhi Zhang, Rongxing Du, Anit Kumar Sahu, and Ruoxi Jia. 2024a. Get more for less: Principled data selection for warming up fine-tuning in llms. *arXiv preprint arXiv:2405.02774*.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. 2024b. Autoscale: Automatic prediction of compute-optimal data composition for training llms. *arXiv preprint arXiv:2407.20177*.

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*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023a. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024a. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and 1 others. 2024b. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503*.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Walsh, Yanai Elazar, Kyle Lo, and 1 others. 2024. Paloma: A benchmark for evaluating language model fit. *Advances in Neural Information Processing Systems*, 37:64338–64376.

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

AI Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai.meta.com/blog/llama-4-multimodal-intelligence/*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

A Paszke. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Ajay Patel, Colin Raffel, and Chris Callison-Burch. 2024. Datadreamer: A tool for synthetic data generation and reproducible llm workflows. *arXiv preprint arXiv:2402.10379*.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. 2024. Improving pretraining data using perplexity correlations. *arXiv preprint arXiv:2409.05816*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mathurin Videau, Badr Youbi Idrissi, Daniel Haziza, Luca Wehrstedt, Jade Copet, Olivier Teytaud, and David Lopez-Paz. 2024. Meta Lingua: A minimal PyTorch LLM training library.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, and 1 others. 2024. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37:116462–116492.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. 2024. Synthetic continued pretraining. *arXiv preprint arXiv:2409.07431*.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*.

A Experimental Setup and Implementation Details

A.1 Datasets

A.1.1 Natural Data Baseline

Our natural data baseline consists of English text sourced from *unfiltered CommonCrawl (CC) dumps processed via the RedPajama-v2 pipeline* (Weber et al., 2024). This serves as our reference point representing widely used, large-scale web data.

A.1.2 Synthetic Data Generation

To ensure consistency, a single generator model, *Mistral-Instruct-7b-v0.1* (Jiang et al., 2023), was used for generating all synthetic datasets described below. Input documents for generation methods requiring a source were sampled from our unfiltered CommonCrawl baseline data. Standard sampling parameters (temperature=0.7, top-p=0.95) were generally used, unless otherwise specified (see Appendix B for more details on prompts and post-filtering).

Method A: Web Rephrasing (WRAP-like) Inspired by WRAP (Maini et al., 2024), we generated two stylistic variants by prompting Mistral-Instruct-7b-v0.1 to rewrite input CC documents (up to 2k tokens):

- *High-Quality (HQ) Rephrasing:* Used prompts optimized from the original WRAP work to produce longer (∼550 tokens avg.) and more coherent synthetic texts, mimicking high-quality, well-structured English (e.g., Wikipedia style). The prompt aimed for clarity, coherence, and quality improvement while preserving core information.
- *Question-Answering (QA) Rephrasing:* Used prompts optimized to restructure the input document's content into a conversational QA format (∼550 tokens avg.), embedding instruction-following patterns.

Method B: Synthetic Textbooks (Phi-like) Inspired by Phi (Li et al., 2023b) and related community efforts (e.g., Cosmopedia (Ben Allal et al., 2024)), we generated novel textbook-style content. This involved a two-step process:

1. An outline for a 10-chapter "book" was constructed based on keywords or themes extracted from input CC documents.
2. Based solely on the outline, each chapter was generated (∼450 tokens/chapter, ∼5k tokens/book), often including exercises and reference answers. We employed 4 prompt variations targeting different audiences (e.g., "grade school students", "college students", "domain experts", "general audience") to encourage diversity.

Light heuristic filtering was applied post-generation to remove clearly malformed outputs (details in Appendix B).

A.1.3 Data Mixtures

We created training datasets for various conditions by concatenating and shuffling the source data. For each synthetic data type (HQ, QA, Textbook), we prepared the following conditions relative to the unfiltered CC baseline:

- 100% Natural (Unfiltered CC)
- 100% Synthetic (consisting entirely of one synthetic type: HQ, QA, or Textbook)
- 67% Synthetic / 33% Natural
- 33% Synthetic / 67% Natural

For each experimental point (model size, data budget), models were trained on these different mixture ratios corresponding to one synthetic type, allowing for direct comparison against the 100% natural baseline. We typically trained 3-5 model variants per condition to enable robust scaling law analysis.

A.2 Models and Training Configuration

A.2.1 Model Architecture

We employ a standard decoder-only Transformer architecture based on *Llama 3 models* (Grattafiori et al., 2024). Key architectural features include SwiGLU activation functions, RMSNorm for layer normalization, and Rotary Position Embeddings (RoPE). We train models at multiple sizes, *ranging logarithmically from approximately 100 Million to 3 Billion parameters*, to facilitate scaling law analysis. In the research of scaling laws for large language models (LLMs), model sizes are counted as *non-embedding parameters*, which refers to all the *learnable parameters* in the model except for those associated with the input and output token embeddings (Kaplan et al., 2020). This work follows this setup and the size of all models being trained refers to their count of non-embedding parameters. Table 1 outlines the architectural parameters for various model sizes explored.

A.2.2 Training Hyperparameters

All models were trained from scratch (random initialization) using the *Meta Lingua library* (Videau

Table 1: Model Architecture for Different Parameter Sizes

Model Size	Dimension (d_{model})	Layers (n_{layers})	Heads (n_{heads})
100M	576	7	9
200M	832	10	13
500M	1280	16	20
1B	1792	22	28
2B	2240	28	35
3B	2624	32	41

et al., 2024) on PyTorch (Paszke, 2019) for efficient distributed training. We used the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$, a weight decay of 0.1. A cosine learning rate schedule was used with a linear warmup equivalent to 10% of the total training steps. A consistent context length of 4096 tokens with an effective batch size of 1M tokens was used across all training runs. The Llama 3 pretrained TikToken tokenizer with a 128k vocabulary size was used (Grattafiori et al., 2024). Specific global batch sizes and gradient accumulation strategies were maintained consistently for comparable experimental settings to ensure fair comparisons. The models were trained using the hyperparameters detailed in Table 2.

Table 2: Training Hyperparameters

Hyperparameter	Value
Optimizer	AdamW
Peak Learning Rate	3.0×10^{-4}
Warmup Steps	10%
Min. LR Ratio	1.0×10^{-6}
Gradient Clipping	1.0
Batch Size (per device)	4
Accumulation Steps	8
Sequence Length	4096
(Effective Batch Size)	(1M tokens)
Prefetch Size	1024
Add BOS token	True
Add EOS token	True
Model Data Type	bf16
Epochs	1
GPU Hardware	8x NVIDIA A100 80GB

Training time scales linearly with the training data size and scales nearly linearly with the model size (provided that it fits into GPU RAM without changing the batch size). Typically, we were able to train a 1B-parameter model for $\approx$ 15B tokens per day on a single node with 8x NVIDIA A100 80GB GPUs.

A.2.3 Evaluation Protocol

Model performance was evaluated using intrinsic metrics during training and a suite of downstream tasks post-training. Validation perplexity (log ppl equivalent to cross-entropy loss) was tracked during training. Final perplexity was calculated on a diverse set of *14 non-code/math English text domains from The Pile* (Gao et al., 2020) (specifically: NIH ExPorter, Pile-CC, Wikipedia (en), USPTO Backgrounds, PubMed Central, PubMed Abstracts, PhilPapers, OpenWebText2, OpenSubtitles, Gutenberg (PG-19), FreeLaw, BookCorpus2, Books3, ArXiv) and also on the *Wikitext-103* dataset (Merity et al., 2016). Per-token average perplexity across these domains serves as a key intrinsic performance measure. For model evaluation, the generator settings detailed in Table 3 were used.

Table 3: Hyperparameters for Perplexity Evaluation

Hyperparameter	Value
Max Tokens to Generate	2048
Generator Data Type (dtype)	bf16

B Synthetic Data Generation: Prompts, Samples, and Parameters

This section provides further details on the synthetic data generation process, including the prompts used, sample outputs, and specific generation/filtering parameters.

B.1 Prompt Templates for Synthetic Data Generation

For HQ Rephrasing, prompts were adapted from Maini et al. (2024) and modified to encourage longer, high-quality text. For QA Rephrasing, prompts were designed to convert informational text into a question-answer dialogue format, adapted from Maini et al. (2024) and modified to promote better format-following and conversion for complete information. For Synthetic Textbooks, prompts guided the generation of chapter content based on outlines, with variations for different target audiences (grade school, college, expert, general).

B.2 Sample Generations

This would ideally showcase the stylistic differences and typical output quality.

B.3 Generation Parameters and Post-Filtering

All synthetic data was generated using a Mistral-Instruct-7b-v0.1 model.

Sampling Parameters: Unless specified otherwise (e.g., for particular prompt explorations not detailed in the main paper), the following sampling parameters were used:

- Temperature: 0.7
- Top-p (nucleus sampling): 0.95

These parameters were chosen to balance creativity and coherence in the generated text.

Post-Filtering: Light heuristic post-filtering was applied to all generated synthetic datar, removing documents that were excessively short (e.g., less than 50 tokens) or excessively long relative to the target length for that generation type, if such outputs occurred despite prompt length guidance. The goal of this light filtering was to remove egregious generation errors without overly sanitizing the data or significantly altering its distribution.

C Supplementary Discussion on Related Work

This appendix provides supplementary details to the related work discussed in Section 2, offering further context on synthetic data applications, data curation practices, and model collapse theories.

C.1 Synthetic Data in Post-training

The use of synthetic data is particularly well-established and successful in post-training phases, primarily for aligning LLMs with human instructions and preferences. This success stems from the ability to generate large amounts of targeted data for specific, often narrow, objectives where human annotation is costly or slow. Key examples include:

Instruction Generation (Self-Instruct): Techniques like Self-Instruct (Wang et al., 2022) use an LLM to bootstrap instruction-following data (instruction, input, output tuples) from a small seed set, enabling effective instruction fine-tuning without extensive human labeling.

B.1.1 Prompt Template HQ Rephrasing

- **System Prompt:** Provide direct and detailed response to the instructions without adding additional notes.
- **[USER]:** For the following document, regardless of its original content or formatting, write a full article of the same content in high quality English language as in texts on Wikipedia: [xxxx]. Provide the rephrased article without any additional notes. Long article with full length and complete details. Rephrased article:

B.1.2 Prompt Template QA Rephrasing

- **System Prompt:** Provide direct and detailed response to the instructions without adding additional notes.
- **[USER]:** For the following document, regardless of its original content or formatting, convert it into a comprehensive list of question-answer pairs with multiple tags of "Question:" followed by "Answer:", where questions and answers cover complete information of the original document. Document: [xxxx]. Provide the converted question-answer pairs without any additional notes. Question-answer pairs with corresponding tags ("Question:", "Answer:"):

B.1.3 Prompt Template for Generating Textbook-style Synthetic Data: Step 1, Outline Generation

- **Step 1: generate an outline based on input text.**
- **System Prompt:** Provide direct and detailed response to the instructions without adding additional notes.
- **[USER] <4 versions>:** Imagine you are a prolific author tasked with writing a textbook. You are working on writing a textbook involving the knowledge and information of the following text. Text: [xxxx]\n Your task is to write an outline for the textbook. Your target audiences are <grade school students/college students/field experts/general public>. The textbook has 10 chapters in total plus title, introduction, and appendices. Textbook outline:

B.1.4 Prompt Template for Generating Textbook-style Synthetic Data: Step 2, Chapter Generation

- **Step 2: generate each section based on outline.**
- **System Prompt:** Provide a direct and detailed response to the instructions without adding additional notes.
- **[USER]:** Imagine you are a prolific author tasked with writing a textbook. You are working on writing a textbook with the following outline.\n Outline: [xxxx] \n Your task is to write Chapter x of the textbook. Your target audiences are grade school students. Include exercises at the end of the chapter to test the reader's knowledge of the chapter and then provide reference answers to each question.

B.2.1 Sample Generation for HQ Rephrasing

- **[Input text]:** Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989 Home

  / Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989

  Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989 More Views

  The B & B intercooler offers 100% more cooling area than the stock unit, and 35% more than any other aftermarket cooler. Due to this increased size, the cooler has greater thermal efficiency, which means cooler air to the engine. This will reduce the chance of detonation and allow for greater boost pressure for more power. Bolts in stock location with minor modifications. Choose from Silver Aluminum or Black Powder Coated (allow 2-weeks). APPLICATIONS

  PORSCHE 911 (930) 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989

- **HQ Rephrased:** The Porsche® 911 B&B Hi-Capacity Intercooler is a high-performance cooling system designed for the Porsche 911 (930) model from 1978 to 1989. This intercooler offers 100% more cooling area than the stock unit, and 35% more than any other aftermarket cooler. This increased size allows for greater thermal efficiency, which means cooler air to the engine. This, in turn, reduces the chance of detonation and allows for greater boost pressure, resulting in more power.

  The B&B intercooler is designed to bolt into the stock location with minimal modifications. It is available in two finishes: Silver Aluminum and Black Powder Coated. Customers should allow 2-weeks for delivery of the Black Powder Coated finish.

  The applications for the Porsche® 911 B&B Hi-Capacity Intercooler are the Porsche 911 (930) models from 1978 to 1989. This intercooler is designed to provide superior cooling performance, making it an ideal choice for those looking to improve the performance of their Porsche 911.

B.2.2 Sample Generation for QA Rephrasing

- **[Input text]:** Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989 Home

  / Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989

  Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989 More Views

  The B & B intercooler offers 100% more cooling area than the stock unit, and 35% more than any other aftermarket cooler. Due to this increased size, the cooler has greater thermal efficiency, which means cooler air to the engine. This will reduce the chance of detonation and allow for greater boost pressure for more power. Bolts in stock location with minor modifications. Choose from Silver Aluminum or Black Powder Coated (allow 2-weeks). APPLICATIONS

  PORSCHE 911 (930) 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989

- **QA Rephrased:** Question: What is the name of the product?

  Answer: Porsche® 911 B&B Hi-Capacity Intercooler, 1978-1989

  Question: What is the cooling area of the stock unit?

  Answer: 100% more cooling area than the stock unit

  Question: What is the cooling area of the B&B intercooler compared to other aftermarket coolers?

  Answer: 35% more than any other aftermarket cooler

  Question: What is the purpose of the increased size of the cooler?

  Answer: Greater thermal efficiency, which means cooler air to the engine

  Question: What effect will this have on the engine?

  Answer: Reduced chance of detonation and allow for greater boost pressure for more power

  Question: Where will the bolts for the cooler be installed?

  Answer: Bolts in stock location with minor modifications

  Question: What are the two options for the color of the cooler?

  Answer: Silver Aluminum or Black Powder Coated

  Question: How long will it take for the cooler to arrive?

  Answer: 2-weeks for Black Powder Coated

  Question: What are the applications of the B&B intercooler?

  Answer: Porsche 911 (930) 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989.

B.2.3 Sample Generation for Textbook-style Synthetic Data: Step 1, Outline Generation

- **[Sample outline <general public>]:** Title: The Baltimore Orioles' Spring Training

  Introduction:

  * Brief overview of the Orioles' history and recent successes

  * Importance of spring training in preparing for the regular season

  * Purpose of the textbook: To provide readers with an in-depth look at the Orioles' spring training activities

  Chapter 1: Overview of the Orioles' Spring Training Facilities

  * Description of the team's two main spring training fields: Ed Smith Stadium and the Gulf Coast League

  * Explanation of the new artificial turf field at Ed Smith Stadium

  * Importance of having multiple fields for inclement weather and rehabbing injured players

  Chapter 2: Injury Updates and Rehabilitation

  * Overview of the Orioles' key injuries and their status during spring training

  ...

  Chapter 10: The Future of Orioles Baseball

  * Overview of the team's current roster and their potential for success in the upcoming season

  * Description of the team's prospects and their potential to develop into top Major League players

  * Explanation of the importance of continued investment in player development and team building for long-term success

  Conclusion:

  * Recap of the key takeaways from the Orioles' spring training

  * Final thoughts on the importance of spring training and the team's potential for success in the upcoming season

  Appendices:

  * List of Orioles' major league roster and key prospects

  * Glossary of baseball terms and concepts discussed in the textbook

  * Additional resources for further reading on baseball and the Orioles.

Preference Data for RLHF: Reinforcement Learning from Human Feedback (RLHF) requires data indicating preferences between model outputs. Synthetic data generation can augment or replace human effort in creating these preference pairs, facilitating large-scale alignment (Taori et al., 2023).

AI Feedback and Constitutional AI: Methods like Reinforcement Learning from AI Feedback (RLAIF) and Constitutional AI (Bai et al., 2022) utilize AI models themselves, guided by principles or rules, to provide feedback or generate preferred responses, further automating the alignment process.

C.2 Data Curation Practices

The quality and composition of pre-training data are paramount. Major dataset development efforts highlight common practices:

Large-Scale Corpora Examples:

- C4 (Colossal Clean Crawled Corpus) (Raffel et al., 2020): Derived from Common Crawl using heuristics like language filtering (retaining primarily English text), removing code/markup, filtering based on blocklists, and deduplication at the document level.

- The Pile (Gao et al., 2020): A diverse dataset combining 22 different high-quality sources, including academic papers (PubMed, arXiv), books (Books3), code (GitHub), web text (Pile-CC), and conversational data, with source-specific filtering.

- RefinedWeb (Penedo et al., 2024): Focused on rigorous filtering and aggressive fuzzy deduplication of web data from Common Crawl to create a high-quality, large-scale web corpus, arguing against heuristic domain mixing.

Domain Mixing: Research actively explores the impact of mixing data from different sources (Liu et al., 2024a; Xie et al., 2023; Kang et al., 2023). For example, including code data (Touvron et al., 2023) or synthetic reasoning data (Abdin et al., 2024) has been shown to improve reasoning, while the optimal ratio of web text, books, and other domains may vary depending on evaluation metrics and model scale (Ye et al., 2024; Kang et al., 2024b,a).

C.3 Model Collapse Mechanisms

The theoretical concern of model collapse (Shumailov et al., 2023; Dohmatob et al., 2024b,a) posits that training generative models on their own output can lead to degenerative feedback loops. Proposed mechanisms include:

Distributional Drift: The distribution of synthetically generated data may subtly differ from the true underlying data distribution. Iterative training

can amplify these differences, causing the model's learned distribution to drift further away.

Loss of Diversity: Models might over-represent common modes in the data they generate, leading to a gradual loss of information about less frequent phenomena or the tails of the distribution ("tail-forgetting").

Artifact Amplification: Flaws, biases, or stylistic quirks of the generator model may be replicated and amplified in subsequent generations trained on its output. Understanding the empirical conditions under which these theoretical risks manifest in large-scale LLM training is an ongoing research effort.

D Additional Experiment Results

We provide some additional results and analyses to experiments and studies in Sections 4 and 5.

D.1 Additional Results on Section 4.1.3

We trained 1B-parameter models on various data mixtures with data budgets from 1B up to 200B tokens. The scaling formula was fitted using data points up to 100B tokens; predictions were then validated on runs trained with 200B tokens. Six data mixtures (CommonCrawl, 33% HQ + 67% CC, 33% QA + 67% CC, Textbook (TXBK), 67% TXBK + 33% CC, and 33% TXBK + 67% CC) were trained to 200B tokens for this validation, as our HQ and QA synthetic datasets were limited to 100B tokens each. The fit demonstrated high precision, as shown in Fig. 6, achieving a low Relative Mean Absolute Error (RMABE) of 0.41% when predicting for 200B tokens.

D.2 Additional Results on Section 4.1.4

We trained models from 100M to 3B parameters for 50B tokens on all 10 data mixtures. The formula was fitted using models up to 2B parameters and validated on 3B-parameter models. This fit also proved highly precise (Fig. 7), with an RMABE of 0.30% for 3B-parameter model predictions.

D.3 Additional Results on Section 4.1.5

Figure 8 shows joint fitted scaling law predictions for models trained only on CommonCrawl (CC) with visualization for loss contours.

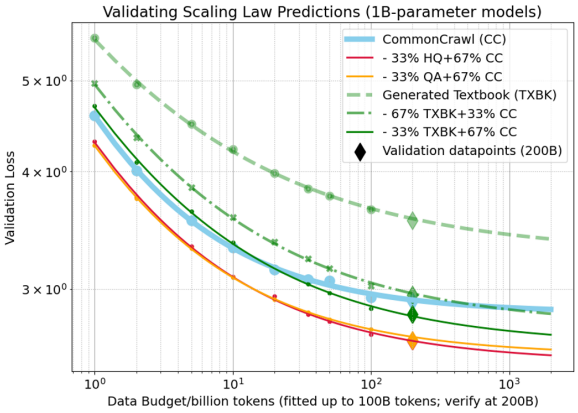


Figure 6: Validation of the data scaling formula. Predictions for 200B tokens (fitted using up to 100B tokens) achieve an RMABE of 0.41%. Solid dots display actual loss values while the fitted curves shows predicted loss. Validation datapoints are illustrated by diamond marks.

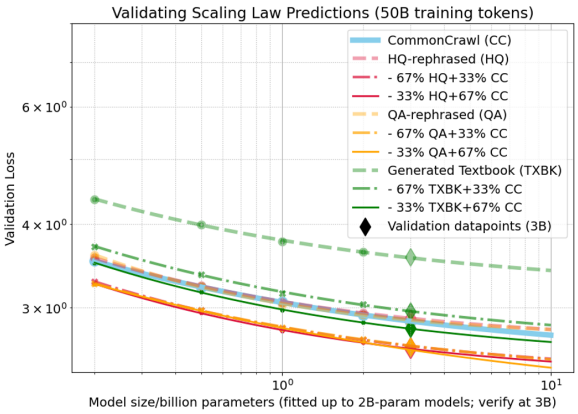


Figure 7: Validation of the model scaling formula. Predictions for 3B-parameter models (fitted using up to 2B-parameter models) achieve an RMABE of 0.30% on validation datapoints illustated with diamond marks.

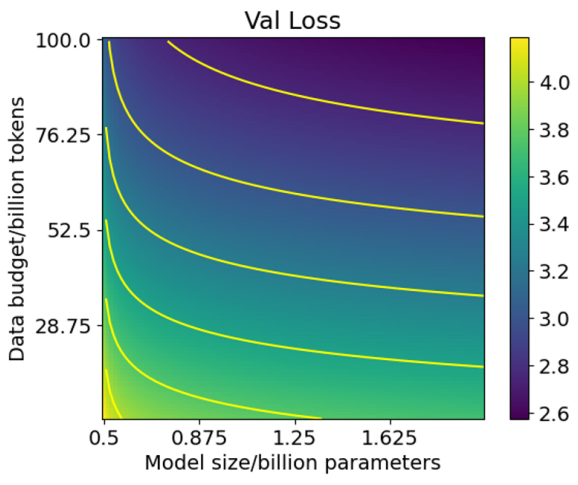


Figure 8: Joint fitted scaling law predictions for models trained only on CommonCrawl (CC). Yellow lines are loss contours.

Table 4: Fitted coefficients for Figure 1, Data Scaling.

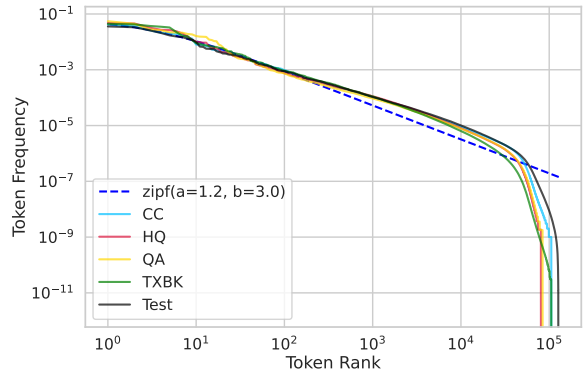
Fitted Coefficients:	CommonCrawl (CC)	HQ-rephrased (HQ)	67% HQ+ 33% CC	33% HQ+ 67% CC	QA-rephrased (QA)	67% QA+ 33% CC	33% QA+ 67% CC	Generated Textbook (TXBK)	67% TXBK+ 33% CC	33% TXBK+ 67% CC
B	1.75424788	2.08762692	1.87764133	1.78983191	1.95783297	1.73878781	1.71111225	2.25602694	2.2130317	2.06636438
β	0.55550749	0.48937729	0.51033911	0.48672269	0.45444664	0.48231484	0.50447156	0.38402667	0.42750747	0.44700694
E	2.82671678	2.74887338	2.53811788	2.50490252	2.70446033	2.53989634	2.54857649	3.26734143	2.73600177	2.6117112

Table 5: Fitted coefficients for Figure 2, Model Scaling.

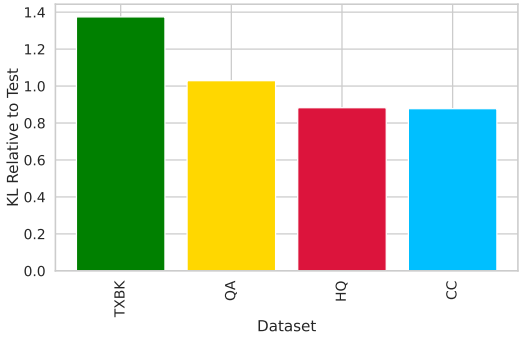
Fitted Coefficients:	CommonCrawl (CC)	HQ-rephrased (HQ)	67% HQ+ 33% CC	33% HQ+ 67% CC	QA-rephrased (QA)	67% QA+ 33% CC	33% QA+ 67% CC	Generated Textbook (TXBK)	67% TXBK+ 33% CC	33% TXBK+ 67% CC
A	0.56088365	0.41624816	0.44475769	0.43919198	0.35099494	0.49017589	0.69735727	0.60660315	0.55037357	0.48812362
α	0.37639592	0.48890806	0.45415438	0.4587784	0.59004325	0.40797778	0.31327323	0.41653062	0.42453172	0.45212858
E	2.491062	2.6459769	2.35668164	2.3364956	2.68957867	2.31629037	2.09815169	3.17632391	2.61596734	2.49067997

D.4 Additional Results on Section 5.3

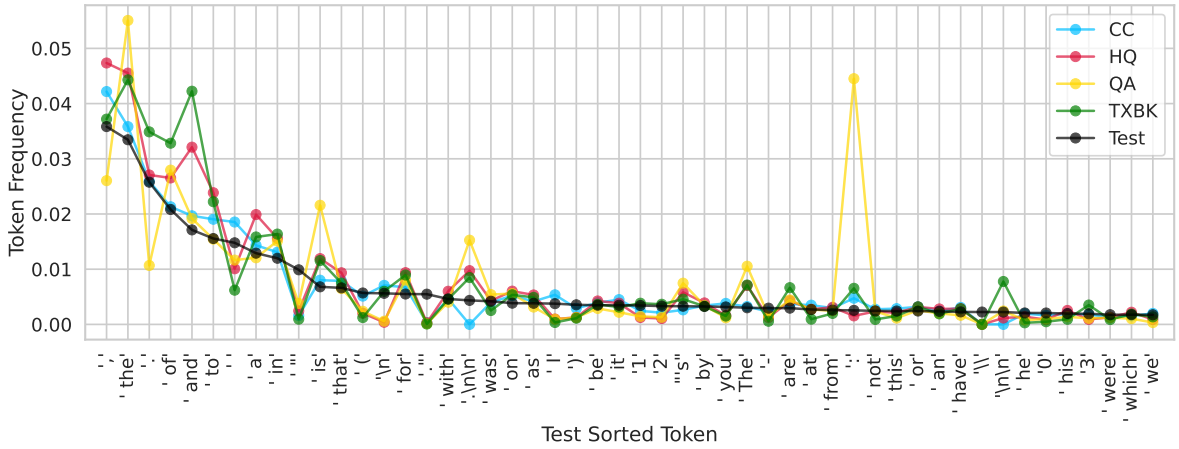
Figure 9 shows analyses of token distributions across datasets and methods. Figure 10 visualizes per-token loss and rolling avergage in evaluation for models trained on respective corpus.



(a) Unigram analysis with Zipf function fitting for unigram (token) frequencies of different training data corpora. CC appears to have wider and slight more uniform coverage of tokens than other training corpora, whereas the test corpora have wider coverage of tokens than training corpora.



(b) KL-divergence beween unigram distributions of the test dataset and each training corpus. CC appears to have the smallest KL-divergence from test data, suggesting the highest distribution similarities, but does not yield high dowmstream model performance.



(c) All estimated token frequencies for each training and test corpus, sorted by the frequency of test tokens. Some methods have relatively lower representation of certain tokens (e.g., CC's ". \n\n") and others increase certain token frequencies (e.g., QA's ":").

Figure 9: Analysis of token distributions across datasets and methods.

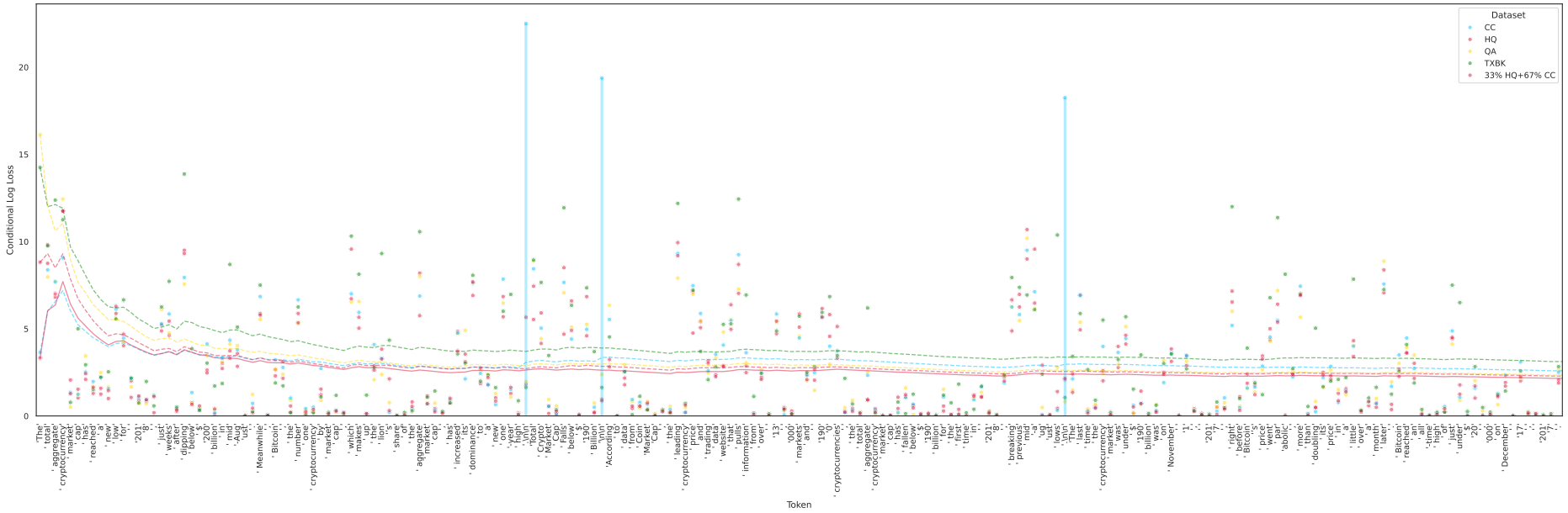


Figure 10: Visualization of per-token loss (dots) and rolling average (curves) in evaluation for 1B-parameter models trained for 100B tokens on respective corpus. The model trained on CC, which has a low frequency for the token ". \n\n", shows high loss in evaluation when encountering this token, though the rolling average remains stable.