SYNTHETIC BOOTSTRAPPED PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Synthetic Bootstrapped Pretraining (SBP), a language model (LM) pretraining procedure that first learns a model of relations between documents from the pretraining dataset and then leverages it to synthesize a vast new corpus for joint training. While the standard pretraining teaches LMs to learn causal correlations among tokens within a single document, it is not designed to efficiently model the rich, learnable *inter-document* correlations that can potentially lead to better performance. We validate SBP by designing a compute-matched pretraining setup and pretrain a 3B-parameter model on up to 1T tokens from scratch. We find SBP consistently improves upon a strong repetition baseline and delivers a significant fraction of performance improvement attainable by an oracle upper bound with access to 20x more unique data. Qualitative analysis reveals that the synthesized documents go beyond mere paraphrases – SBP first abstracts a core concept from the seed material and then crafts a new narration. Besides strong empirical performance, SBP admits a natural Bayesian interpretation: the synthesizer implicitly learns to abstract the latent concepts shared between related documents.

1 INTRODUCTION

Pretraining on the diverse internet texts is now seen to be bottlenecked by the rapid depletion of high-quality text data [56]. This imminent "scaling wall" motivates us to utilize existing data more effectively. Re-examining the conceptual foundation of pretraining, its success originates from the rich causal correlation among tokens *within* a document. However, this is not the only source of correlation pretraining dataset contains: a code document implementing the attention mechanism is derived from the arXiv preprint of the transformer paper; The book of Harry Potter is structurally similar to the screenplay of its movie production. Such connections suggest a weaker form of *inter-document* correlation derived from an underlying joint distribution of pretraining documents. We hypothesize that this additional signal, which is missed by the standard pretraining, can be captured by synthetic data, presenting an underexplored avenue for improving performance.

To leverage this opportunity, we introduce Synthetic Bootstrapped Pretraining (SBP), a LM pretraining procedure that operates in three steps (Figure 1). First, SBP identifies semantically similar document pairs (d_1, d_2), such as the transformer paper and its code implementation, from the pretraining dataset. Second, SBP models the conditional probability of d_2 given d_1, creating a "data synthesizer" that can synthesize a new, related document given a seed document. Finally, SBP applies the trained conditional synthesizer to the pretraining corpus itself, creating a vast text corpus that encodes the rich inter-document correlations that were previously missed (§2). By training a data synthesizer from the pretraining dataset itself, SBP avoids the pitfall of "bootstrapping" model performance using an external, readily available teacher LM, demonstrating a clean setup where the source of improvement stems from better utilization of the same pretraining corpus.

To test our hypothesis, we design a compute-matched, data-constrained experimental framework under which we pretrain a 3B-parameter model on up to 1T tokens from scratch [30, 64], demonstrating the potential applicability of SBP for advancing frontier LMs. We compare SBP's performance against two crucial references: a strong repetition baseline, which represents the standard approach in data-constrained settings, and an oracle upper bound, which has access to an unlimited pool of unique internet data (§3). Our results show that SBP consistently surpasses the strong repetition baseline across different pretraining scales and closes a significant portion of the performance gap to the oracle with 20x additional unique data access (§4.1).

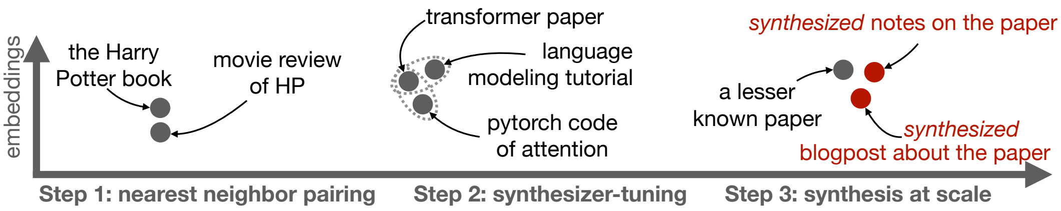

embeddings
the Harry Potter book
movie review of HP
transformer paper
language modeling tutorial
pytorch code of attention
synthesized notes on the paper
a lesser known paper
synthesized blogpost about the paper
Step 1: nearest neighbor pairing
Step 2: synthesizer-tuning
Step 3: synthesis at scale

Figure 1: Data synthesis illustration of Synthetic Bootstrapped Pretraining (SBP): It first identifies semantically similar documents (**Step 1**) and then trains a conditional model that generates one element of the pair from the other (**Step 2**). Finally, SBP applies the conditional model to the pretraining corpus itself to synthesize a new, vast corpus for joint training (**Step 3**).

Besides strong benchmark performances, qualitative analysis of the synthesized documents reveals that they went beyond mere paraphrases of the real documents (§4.2). We postulate that the SBP synthesizer first abstracts latent concepts from the real document and then synthesizes a new document that expands upon the abstracted concept, incorporating diverse genres and content. We formalize this intuition through a Bayesian hierarchical concept model, where documents are related through shared concepts. From this perspective, we argue that the synthesizer implicitly learns a posterior likelihood model that abstracts latent concepts from the document – a mechanism not present in the standard LM pretraining (§5).

In summary, our contributions are threefold:

- **New pretraining framework:** We propose the Synthetic Bootstrapped Pretraining (SBP) algorithm that explicitly models inter-document correlations missed by standard pretraining practice and encodes those correlations into training via synthetic data.
- **Large-scale empirical validation:** We design a compute-matched pretraining setup that enables rigorous measurement of LM self-improvement and empirically validate SBP on a 3B-parameter model trained on up to 1T tokens from scratch.
- **Principled statistical interpretation:** We offer a natural Bayesian interpretation of SBP as implicitly learning a posterior for the latent concepts in a text document and concretize the intuition via qualitative analysis of synthesized documents.

In the remainder of the paper, we will first define the data-constrained pretraining problem we address and introduce the SBP technique we propose in §2. Then, we present the compute-matched experiment setup in §3 and results in §4. Finally, we conclude with a Bayesian interpretation of SBP that sheds light on the origin of the improved performance in §5.

1.1 Related work

Before we proceed, we review related work that highlights our contribution in three broad areas of research: LM pretraining, synthetic data for LM, and retrieval-augmented LM.

LM pretraining. The concept of pretraining, closest to its modern form, originates from a series of works including ELMo [42], ULMFiT [23], BERT [13], that propose to pretrain a neural network via an unsupervised objective and subsequently finetune for a wide range of downstream tasks. The GPT-series [43, 44, 7, 37] cemented the practice of using next-token prediction as the pretraining objective and applying it to large-scale crawled webpages as opposed to task-specific datasets (e.g., English-to-French translation). In recent years, the size of the pretraining corpora has grown rapidly, driven by the availability of massive web-crawled datasets, leading to a successful stream of dataset and pretrained model artifact: BERT [13, 33], GPT-2 WebText [44], CommonCrawl [11], CCNet [59], T5 C4 [46], the Pile [15], Gopher Massive Text [45], Llamda series [55, 14], RefinedWeb [41], Dolma [50], DCLM-baseline [30], NemotronCC [51], etc. While pretraining has been tremendously successful, the rapid depletion of available internet text motivates us to shift our focus from acquiring more data to using the existing data more effectively.

Synthetic data. A natural way to overcome the limitations of scarce high-quality web data is to pretrain [16, 1, 2, 3, 54] or continually pretrain [61, 47, 63, 36] LMs on synthetic data. Existing approaches to data synthesis rely on distillation from a powerful "teacher" LM that generates compressed knowledge representation for the "student" LM to learn [22]. These teacher models must first undergo a human alignment process, which requires extensive human annotations and preference data [38]. Synthetic data from the teacher LM hints at a limited scaling trend: whilst the synthesized data from the teacher LM can be as impressive [12] as 7x more effective than real data, the performance improvement quickly converges to that of the teacher LM [8]. We instead consider the scenario where the sole source of world knowledge comes from a fixed set of pretraining documents (e.g., the internet) and algorithmically learn a data synthesizer with minimal human intervention (e.g., generative teacher models or human writing prompts). Therefore, our experiment setup simulates a situation where the LMs can self-boost their pretraining capability by refining their understanding of the fixed collection of pretraining documents.

Retrieval augmented LM. A natural class of methods that incorporates multiple documents together is retrieval augmented generation (RAG) [28, 29]. While originally introduced as a technique to be used at test-time for a domain-specific downstream task [5, 31], retrieval augmented approaches have been extended in scope: [26] and [60] implement RAG at pretraining scale and show improved test perplexity; [19] incorporates RAG at pretraining time by jointly training a retriever and the model itself for improved QA performance. [49] groups related documents into the same context window for improved long-context capability. In general, while the RAG-related approach enables the model to utilize rich inter-document correlations, it is fundamentally limited by the context window of the LM. In contrast, SBP encodes correlations into synthetic data that can be iteratively learned by the LM one document at a time. Prior to the advancement of embedding models that allow retrieving the entire document, [18] proposed retrieving neighboring pairs of sentences using Jaccard similarity and modeling the conditional distribution between them, similar to our conditional data synthesizer objective; however, they did not perform any pretraining experiments.

2 Our method

In this section, we introduce the data-constrained pretraining setup (§2.1) and then present the SBP procedure in three detailed steps (§2.2). We will present SBP as a general pretraining recipe by introducing a generic setup that includes a pretraining dataset, an LM architecture, and a collection of evaluation benchmarks. We defer the concrete compute-matched experiment design to §3.

2.1 Data-contrained pretraining setup

We consider a *data-constrained* setup where the goal is to train the best-performing LM given access to a fixed document collection $\mathcal{D}_{\text{pretrain}}$ (e.g., a snapshot of the entire internet). To establish a controlled experimental framework, we also choose a transformer architecture with parameters θ and a collection of held-out evaluation benchmarks Perf (e.g., perplexity, few-shot QA accuracy). Recall that a transformer takes in a sequence of tokens and outputs a sequence of conditional probabilities of each token given all previous tokens. Applying the chain rule for joint probability, we can use a transformer to calculate the probability $p_\theta(x)$ of observing a particular text input x, or the conditional probability $p_\theta(x|y)$ of one text x given another y.

Under such a setup defined by ($\mathcal{D}_{\text{pretrain}}$, p_θ, Perf), pretraining searches for the best-performing transformer weights by maximizing the sum of the log-likelihood of pretraining documents $\arg\max_\theta \sum_{d\in\mathcal{D}_{\text{pretrain}}} \log p_\theta(d)$, and then evaluates the performance through $\mathrm{Perf}(\theta)$. Statistically, this objective treats each document as an independent sample from a hypothetical distribution of all documents and attempts to learn this marginal distribution. However, this modeling assumption overlooks the structural similarities shared between natural language texts (e.g., Figure 1). We next present the SBP procedure that fills this gap.

2.2 Synthetic bootstrapped pretraining

At a high level, SBP finds related document pairs (d_1, d_2) from the pretraining dataset $\mathcal{D}_{\text{pretrain}}$ and trains a conditional synthesizer $p_\theta(d_2|d_1)$ using the same transformer architecture parametrized by θ. It then uses it to synthesize a large collection of documents $\mathcal{S}_{\text{pretrain}}$ to perform joint pretraining on $\{\mathcal{D}_{\text{pretrain}}, \mathcal{S}_{\text{pretrain}}\}$. The fact that SBP trains a data synthesizer from $\mathcal{D}_{\text{pretrain}}$ itself also distinguishes it from extensive existing work that relies on a readily available "teacher" LM.

Step 1: Nearest neighbor pairing. In preparation for training the conditional data synthesizer, SBP first curates pairs of related documents. To efficiently perform similarity search at pretraining scale, we adopt the Approximate Nearest Neighbor (ANN) methodology [34], which embeds each document as a quantized vector normalized to the unit sphere and then performs massively parallelizable linear algebraic operations. In our implementation of SBP, we use inner-product similarity, which we denote by $\langle d_1, d_2 \rangle$. Then, we select a subset of pairs whose similarity score exceeds a certain threshold α: $\mathcal{D}_{\text{ST}} = \{(d_1, d_2) \in \mathcal{D}_{\text{pretrain}} \times \mathcal{D}_{\text{pretrain}}, \text{ s.t. } \langle d_1, d_2 \rangle > \alpha\}$. We provide the implementation details of paired data curation in §A.2.

Step 2: Synthesizer-tuning. SBP exploits the correlation between pairs of related documents by maximizing the conditional probability of d_2 given d_1: $\theta_{\text{ST}} = \arg\max_\theta \sum_{(d_1,d_2)\in\mathcal{D}_{\text{ST}}} \log p_\theta(d_2|d_1)$, which we obtain by summing over the log conditional probabilities corresponding to tokens from document d_2. We refer to this step as "synthesizer-tuning" as we are training a conditional probabilistic model that synthesizes a related d_2 from a given d_1. When performing synthesizer-tuning, we initialize p_θ at the transformer weights that has gone through normal pretraining. As a result, the model is equipped with the knowledge of individual documents at initialization, but not the conditional relation between them. Importantly, each document d_1 can be associated with multiple instances of d_2, encouraging the synthesizer to produce diverse, high-entropy outputs rather than deterministic synthesis.

Step 3: Data synthesis at scale. Finally, SBP synthesizes $\mathcal{S}_{\text{pretrain}}$ through a hierarchical sampling process: (I) First sample the seed document d_1 from $\mathcal{D}_{\text{pretrain}}$ uniformly at random; (II) Then sample synthesized document d_2 from $p_{\theta_{\text{ST}}}(\cdot|d_1)$. This process achieves synthetic data diversity utilizing two sources of variation: first through the variation of the seed documents d_1, which comes from the diversity of the pretraining document $\mathcal{D}_{\text{pretrain}}$ itself, and second through the entropy of the conditional distribution $p_{\theta_{\text{ST}}}(\cdot|d_1)$, which stems from the diverse inter-document correlations captured in $\mathcal{D}_{\text{ST}}$. While the procedure is empirically motivated, it actually admits a statistically principled Bayesian modeling of the distribution of natural language texts, which we explain in §5. For now, we focus on demonstrating the empirical effectiveness of SBP.

3 Experiment setup

In this section, we present the *compute-matched* experimental setup we designed to validate SBP against natural reference methods. Before diving into the details of this design, we briefly mention our choice of data, model, and evaluation: We curated a pretraining dataset by cleaning and filtering DCLM [30], implemented a 3B-parameter transformer architecture modified from Llama 3 [14], and selected nine commonly used benchmarks targeted at general world knowledge and commonsense reasoning (§A.1). Note that for MMLU [21], we find that accuracy-based evaluation yields non-smooth performance changes for small models. We therefore designed a perplexity-based MMLU to track smooth progress changes during training. Our largest experiment trains the 3B model on up to 1T total training tokens (§3.1), bringing validation at a scale relevant for frontier LM development.

3.1 Compute-matched comparsion

We use a *compute-matched* experimentation framework to rigorously compare SBP against two natural references: a repetition baseline where we repeat $\mathcal{D}_{\text{pretrain}}$ multiple times to utilize the available training compute and an oracle upper bound that enables the model to access as many unique documents as possible. Operationally, we control the training compute by controlling the total tokens seen during training, which is proportional to the training FLOPs given a fixed batch size and context window. We validate SBP across two different scales:

- **200B-scale**: In this setting, we cap the training compute to be 200B tokens and cap the data access at $\|\mathcal{D}_{\text{pretrain}}\| = 10\text{B}$ tokens.
- **1T-scale**: We also consider a larger scale closer to frontier model training, where we cap the training compute at 1T tokens and data access at $\|\mathcal{D}_{\text{pretrain}}\| = 50\text{B}$ tokens.

For each training scale, $\mathcal{D}_{\text{pretrain}}$ with different sizes is sampled uniformly at random from the 582M documents pool. Given the compute-controlled comparison scheme, we next introduce two reference methods against which we compare SBP.

Repetition baseline. Since the compute budget typically exceeds the total number of unique tokens $\|\mathcal{D}_{\text{pretrain}}\|$, a natural baseline to use the additional compute is to repeat $\mathcal{D}_{\text{pretrain}}$ over multiple epochs. By design, in both 200B-scale and 1T-scale, we repeat the pretraining dataset $\mathcal{D}_{\text{pretrain}}$ 20 times to exploit the available compute budget. In practice, when the pretraining dataset comes from a mixture of different sources, higher-quality documents can be seen as many as 30 times during

Table 1: Computed-matched comparison of Synthetic Bootstrapped Pretraining (SBP) and oracle performance gains over the repetition baseline. On average, SBP delivers roughly 43% of the performance improvement in QA accuracy attainable by an oracle with access to 20x more unique data.

	200B-scale			**1T-scale**		
Benchmark	**Baseline**	**SBP**	**Oracle**	**Baseline**	**SBP**	**Oracle**
			Perplexity on held-out data ↓			
OpenWebText2	5.74	-0.53	-1.02	4.51	-0.02	-0.12
LAMBADA	6.87	-0.85	-1.86	4.33	-0.03	-0.22
Five-shot MMLU	3.83	-0.36	-0.51	3.17	-0.06	-0.05
			QA accuracy ↑			
ARC-Challenge (0-shot)	35.32	+1.28	+2.82	42.66	+1.62	+3.84
ARC-Easy (0-shot)	68.94	+2.65	+4.29	75.63	+0.42	+2.11
SciQ (0-shot)	90.50	+1.00	+2.40	93.20	+0.80	+0.50
Winogrande (0-shot)	60.14	+1.90	+5.53	65.19	+1.42	+2.92
TriviaQA (1-shot)	22.51	+3.36	+7.37	36.07	+0.25	+0.59
WebQS (1-shot)	8.56	+3.74	+10.83	19.34	+0.54	+0.44
Average QA accuracy	**47.66**	**+2.32**	**+5.54**	**55.35**	**+0.84**	**+1.73**

pretraining, while lower-quality texts may appear only once. [35] systematically evaluates the repetition baseline as a proposal to scale LMs under data constraints and finds that repeating $\mathcal{D}_{\text{pretrain}}$ up to 4 times yields nearly no performance degradation compared with having access to unlimited fresh data, but after around 40 times, repetition yields rapidly diminishing returns. Therefore, our choice of 20 times repetition with compute-matched comparison strikes a reasonable balance between efficient experimental execution and exhausting all possible performance gains from a fixed $\mathcal{D}_{\text{pretrain}}$ via repetition.

Oracle upper bound. Besides showing improvement against the repetition baseline, we also evaluate an oracle upper bound with unlimited data access. The motivation behind this is to contextualize the numerical improvement delivered by SBP. As we shall see in the next section, because different benchmarks respond differently to data size changes, SBP can deliver an improvement as large as 3.74% on some benchmarks but only 0.14% on others (Table 1). Also, as performance on LM benchmarks tend to scale logarithmically [39, 25] against data improvement, the numerical difference quickly caps out as we move from the 200B scale to the 1T-scale. By introducing this oracle upper bound, we can contrast the SBP improvement against this "oracle" improvement.

Training recipe. For both the repetition baseline and oracle upper bound at both 200B-scale and 1T-scale, we use a batch size of 2,048 and a context window of 4,096, resulting in a throughput of 8M tokens per step. We apply a cosine learning rate scale with a 5% warmup to a peak learning rate of 1e-2, followed by subsequent decay to 5e-5 towards the end. Under this setup, pretraining costs 11K v5p-TPU hours at 200B-scale and 59K v5p-TPU hours at 1T-scale. For a clean comparison, we adhere to this hyperparameter throughout the paper, including the SBP experiment presented next.

4 Experiment Results

We perform SBP experiments under the compute-matched framework outlined in §3 at two levels of training compute budget: 200B-scale and 1T-scale. After joint training on real and synthetic data $\{\mathcal{D}_{\text{pretrain}}, \mathcal{S}_{\text{pretrain}}\}$, we find SBP consistently improves upon the repetition baseline throughout both scales (Table 1). In this section, we focus on presenting the performance of SBP and evaluating the quality of the synthesized pretraining data. We defer the implementation details of SBP to §A.2.

4.1 Main benchmark performance

At the 200B-scale, we start with the source dataset of $\|\mathcal{D}_{\text{pretrain}}\| =$10B and curate a SBP dataset of $\|\mathcal{S}_{\text{pretrain}}\| =$75B tokens (detailed ablation in §A.3). We perform joint training on $\{\mathcal{D}_{\text{pretrain}}, \mathcal{S}_{\text{pretrain}}\}$

with the principle that we do not repeat any synthetic documents during training. This means that out of a 200B token training budget, we spent 37.5% of it on the 75B synthetic tokens from $\mathcal{S}_{\text{pretrain}}$ without any repetition, and the remaining 62.5% on the real dataset $\mathcal{D}_{\text{pretrain}}$ repeated 12.5 times. As shown in Table 1, SBP consistently decreases test loss and improves QA accuracy. On average, SBP captures $2.32/5.54 = 42\%$ of the improvement in QA accuracy delivered by the oracle run with 20x additional data access.

The training dynamics of SBP partly reveal its core mechanism. As we can see in Figure 2, initially, the baseline performs similarly to the oracle, since their training data share the same distribution, and when the number of tokens seen is small, there is no distinction between the two. Then gradually, the oracle becomes a better model than the baseline, as it has access to unlimited unique training data. For the SBP dynamics, it initially performs worse than both the baseline and the oracle, which is expected since the quality of the synthesized data at most matches that of the real data. However, gradually, the SBP continues to scale while the baseline has plateaued. This suggests that $\mathcal{S}_{\text{pretrain}}$ offers a signal $\mathcal{D}_{\text{pretrain}}$ alone cannot capture.

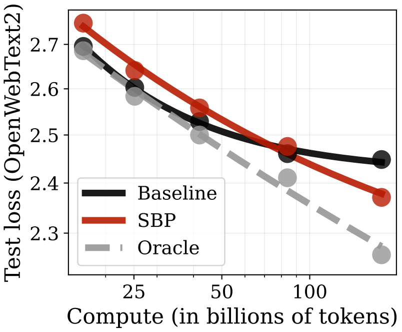


Figure 2: Training dynamics (200B-scale).

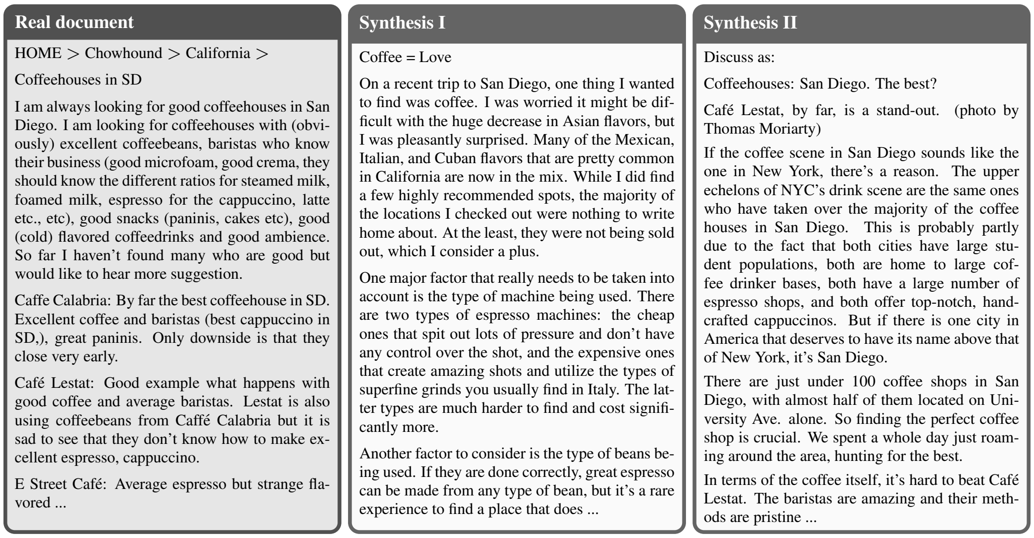
Lastly, to validate the benefit of SBP across different training scales, we implement a larger experiment with $\|\mathcal{D}_{\text{pretrain}}\| = 50\text{B}$ unique tokens under a compute budget of 1T total training tokens. We generate 125B total synthetic tokens $\|\mathcal{S}_{\text{pretrain}}\| = 125\text{B}$, with similar ablation presented in §A.3, and adhere to the same no-repetition-for-synthetic-data principle adopted at the 200B-scale. Examining the oracle improvement from Table 1, we can see that the perplexity-based measurements and most QA benchmarks have plateaued at this scale [32]. However, ARC-Challenge and Winogrande continue to deliver smooth performance changes, making them suitable candidates for tracking pretraining capability at large scales. In particular, in ARC-Challenge, both SBP and Oracle yield a larger performance improvement than their 200B-scale counterparts. That said, other benchmarks still provide a directional signal of capability improvement. This demonstrates the advantage of having a diverse collection of evaluation benchmarks covering a wide range of difficulties. On average, SBP delivers $0.84/1.73 = 48\%$ of the improvement in QA accuracy attained by the oracle.

4.2 Analysis of synthetic data

In this section, we provide some qualitative and quantitative analyses of the synthesized documents to gain insight into the SBP procedure beyond what is measurable by the benchmark performance.

Qualitative examples. We start by showing some samples of synthesized documents from the 200B-scale experiment (Figure 3) with more samples from 1T-scale presented in §B.4. On the left, we display a real document about a practical, first-person guide to the coffee houses in San Diego. Then, we present two synthesized texts that exhibit notable differences in both framing and depth, with varying degrees of fidelity to the seed document. Synthesis I sticks to the same topic but shifts toward an expository essay on espresso machines and bean quality, with little mention of specific coffee shops. Synthesis II adopts a promotional, comparative style, linking San Diego's coffee culture to New York's and praising Café Lestat in a way that departs from the original's balanced assessments. Note that SBP provides no instructions on how the synthesizer should use the seed texts to write new documents. The model spontaneously learns to introduce new content and style into the discussion while staying on topic. In fact, it is challenging to manually craft a prompt to an instruction-tuned model that would output either Synthesis I or II with the real document as input. This example highlights how SBP differs from existing paradigms of data synthesis – the output somehow first abstracts the seed document and then synthesizes new text with more generalized narratives, genres, and intent. We provide more extensive analysis of this observation in §5.

Quantitative Analysis. In addition to qualitative examples, we also conduct quantitative evaluations to assess the quality of the generated texts. We measure text distributions for the synthesized document at 200B-scale and 1T-scale. To establish a reference, we also conduct the same evaluation on the real documents. We measure five basic quality indicators:



Real document

HOME > Chowhound > California >

Coffeehouses in SD

I am always looking for good coffeehouses in San Diego. I am looking for coffeehouses with (obviously) excellent coffeebeans, baristas who know their business (good microfoam, good crema, they should know the different ratios for steamed milk, foamed milk, espresso for the cappuccino, latte etc., etc), good snacks (paninis, cakes etc), good (cold) flavored coffeedrinks and good ambience. So far I haven't found many who are good but would like to hear more suggestion.

Caffe Calabria: By far the best coffeehouse in SD. Excellent coffee and baristas (best cappuccino in SD,), great paninis. Only downside is that they close very early.

Café Lestat: Good example what happens with good coffee and average baristas. Lestat is also using coffeebeans from Caffé Calabria but it is sad to see that they don't know how to make excellent espresso, cappuccino.

E Street Café: Average espresso but strange flavored ...

Synthesis I

Coffee = Love

On a recent trip to San Diego, one thing I wanted to find was coffee. I was worried it might be difficult with the huge decrease in Asian flavors, but I was pleasantly surprised. Many of the Mexican, Italian, and Cuban flavors that are pretty common in California are now in the mix. While I did find a few highly recommended spots, the majority of the locations I checked out were nothing to write home about. At the least, they were not being sold out, which I consider a plus.

One major factor that really needs to be taken into account is the type of machine being used. There are two types of espresso machines: the cheap ones that spit out lots of pressure and don't have any control over the shot, and the expensive ones that create amazing shots and utilize the types of superfine grinds you usually find in Italy. The latter types are much harder to find and cost significantly more.

Another factor to consider is the type of beans being used. If they are done correctly, great espresso can be made from any type of bean, but it's a rare experience to find a place that does ...

Synthesis II

Discuss as:

Coffeehouses: San Diego. The best?

Café Lestat, by far, is a stand-out. (photo by Thomas Moriarty)

If the coffee scene in San Diego sounds like the one in New York, there's a reason. The upper echelons of NYC's drink scene are the same ones who have taken over the majority of the coffee houses in San Diego. This is probably partly due to the fact that both cities have large student populations, both are home to large coffee drinker bases, both have a large number of espresso shops, and both offer top-notch, handcrafted cappuccinos. But if there is one city in America that deserves to have its name above that of New York, it's San Diego.

There are just under 100 coffee shops in San Diego, with almost half of them located on University Ave. alone. So finding the perfect coffee shop is crucial. We spent a whole day just roaming around the area, hunting for the best.

In terms of the coffee itself, it's hard to beat Café Lestat. The baristas are amazing and their methods are pristine ...

Figure 3: Comparison of original text with synthesized text variations.

- **Repetition:** A document may contain too many repeated sentences or patterns. Repetition rate refers to the fraction of documents that exhibit this problematic behavior.
- **Duplicate@1M:** Another failure mode of synthesis is when the documents sampled from the synthesizer distribution are nearly duplicates of each other. Duplicate@1M refers to the fraction of unique documents (determined by Jaccard similarity at a threshold of 0.6) when 1M documents are sampled from the text distribution.
- **Non-factual:** A common failure mode of synthesis is the generation of content that contradicts established knowledge or facts. Non-factual rate refers to the fraction of documents that contain verifiable factual errors, as determined by automated fact-checking tools.
- **Pair-irrelevance:** The synthesized d_2 is considered relevant to d_1 if they pertain to the same topic, event, entity, person, place, or object. Pair-irrelevance refers to the fraction of synthesized d_2 that is not relevant to d_1, indicating the synthesis is not rightly using information from d_1.
- **Pair-copying:** d_1 and d_2 are considered near-duplicates if they are almost identical, except for some extra white spaces, line breaks, or punctuation. Pair-copying refers to the fraction of synthesized d_2 that is a near duplicate of d_1.

Operationally, we implement Repetition, Pair-irrelevance, and Pair-copying using LM-as-judge (prompts and more implementation details given in §B.3) by sampling 1,000 examples from each distribution and estimating the fraction of documents satisfying each criterion. For Non-factual (prompts and details given in §B.2), we sample 10,000 examples and conduct a comprehensive examination of factual errors to ensure broader coverage of the generated data. For Duplicate@1M, we use rule-based filtering to detect the fraction of duplicates based on 1M documents sampled from each distribution. We present the result in the table below. All metrics are lower for better data.

Table 2: Quantitative evaluation of documents sampled from the synthesizer at 200B-scale and 1T-scale. We can see that the synthesized documents preserve topics and are not are simple duplicates.

	Repetition ↓	**Duplicate@1M** ↓	**Non-factual** ↓	**Pair-irrelevance** ↓	**Pair-copying** ↓
200B-scale	4.3%	0.8%	15.1%	25.6%	0.1%
1T-scale	3.9%	0.8%	8.7%	7.8%	0.9%
Real data	1.8%	0.7%	1.8%	n.a.	n.a.

At a high level, Repetition and Duplicate@1M measure a basic text quality that is independent of the specific pair-synthesis strategy employed by SBP. They aim to detect two simple failure modes: text repetition, a common failure pattern in generations from small language models (3B in our case), and the lack of diversity, a common issue with synthetic data that relies on variation induced by the sampling temperature. From Table 2, we find that both 200B-scale and 1T-scale synthesis match the quality of real data as captured by these two metrics. We note that the absence of repetitions and duplicates is not, in itself, an indicator of high-quality or educational text, but rather a basic sanity check that ensures the synthesized texts are diverse. Non-factual failure stems from hallucinations that introduce non-existent entities or relations inconsistent with reality. We find that synthesis at the 1T-scale significantly reduces these errors compared to the 200B-scale. As the data synthesizer is trained on more data, the factuality of the generated outputs tends to converge toward that of real data. Pair-irrelevance and Pair-copying, on the other hand, measure how synthesized d_2 relates to the seed d_1. There are two failure modes we would like to detect: first, when d_2 is completely irrelevant to d_1, and second, when d_2 merely copies the content of d_1. We observe that both 200B-scale and 1T-scale synthesis avoid simply copying and pasting d_1. More interestingly, we observe that the 1T-scale demonstrates substantially higher relevance than the 200B-scale, which intuitively makes sense as the synthesizer learns more diverse relations among $|\mathcal{D}_{\text{pretrain}}| = 60$M documents than $|\mathcal{D}_{\text{pretrain}}| = 12$M corpus.

At this point, we have shared the results of the experiment. In the appendix, we present the implementation details of SBP in §A.2, ablations involving synthetic data mixture ratio in §A.3, additional analysis of synthesized documents in §B, and comparsion with a larger 6B model in §C.2.

5 Statistical Foundations of SBP

In this section, we present a Bayesian interpretation of the SBP procedure, offering one potential explanation for the origin of the SBP improvement. We will formulate a hierarchical model of natural language texts (§5.1) and demonstrate that SBP implicitly enables LMs to learn a posterior standard pretraining cannot capture. We conclude by connecting our findings from this idealized model to the reality of LM (§5.2). We begin with the observation that the pretraining objective models the marginal likelihood of documents:

$$\arg\max_{\theta} \log p_\theta(\mathcal{D}_{\text{pretrain}}) = \arg\max_{\theta} \sum_{d \in \mathcal{D}_{\text{pretrain}}} \log p_\theta(d). \tag{1}$$

However, different natural language documents share structural similarities (Figure 1), which suggests a potentially more complex underlying joint distribution that we will explore next.

5.1 A hierarchical concept model for natural language

In the transformer example from Figure 1, both the arXiv preprint of the transformer paper and its code implementation are derived from the abstract concept of "transformer neural network". From this perspective, we can view the generation process of natural language documents as a hierarchical sampling process where we first sample a collection of abstract concepts $c^{(i)}$ (e.g., the idea of a transformer) from a semantic space of all concepts $\mathcal{C}$ and then generate new documents $d^{(i,j)}$ conditional on $c^{(i)}$.

If we adopt this view, we can think of the pretraining document as follows.

- **Concept sampling**: Sample a fixed concept collection $\{c^{(i)}\}_i \sim P(c)$.
- **Document generation**: For each concept $c^{(i)}$, generate docuemnts from $\{d^{(i,j)}\}_j \sim P(d|c^{(i)})$ constituting one part of the pretraining dataset.

Under such a model, the structural similarity between documents generated from the same concept is modeled as probabilistic *dependence*. The standard pretraining objective (1) then neglects inter-document correlation and only learns the marginal distribution $P(d) = \int_{c \in \mathcal{C}} P(d|c)P(c)dc$. In this view, the model learns to generate plausible text by first generating a core concept c and then performing the generation $P(d|c)$. In contrast, the synthesizer-tuning objective models a posterior of c given d. To see this, we additionally assume that the curated pairs (d_1, d_2) come from the same underlying concept c. Then, the synthesizer-tuning objective (§A.2) forces the LM to perform a distinct task: $P(d_2|d_1) = \int_{c \in \mathcal{C}} P(d_2|c)P(c|d_1)dc$. Here, we use Bayes' rule and the conditional independence assumption $P(d_2|c, d_1) = P(d_2|c)$, which says that the documents from the same concept are conditionally independent given that concept. As a result, to successfully model $P(d_2|d_1)$, the synthesizer must first perform posterior inference to infer the latent concept c given the document d_1, and then use this inferred concept to synthesize a new document d_2, a signal that is ignored by the standard pretraining objective. To illustrate this concretely, we perform a post-hoc analysis by prompting an LM to identify the shared concepts between the synthesized document and its seed (Table 3 in §B). We can see that while it is difficult to describe a synthesized document as the outcome of a simple transform, such as a paraphrase or summarization, it always share a common underlying concept with its seed origin.

The additional signal from the posterior then enables a form of self-distillation. The synthesizer, by learning a more complex conditional objective, becomes a more knowledgeable "teacher" model that has learned to infer the latent structure of data. The synthetic data it produces is then the knowledge "distilled" from this teacher [22]. The final LM training then acts as a "student" that learns from a combination of real and synthetic data, allowing it to discover information that real data alone cannot reveal.

5.2 From idealized models to language model reality

For real text documents, we do not know the true data-generating process, and any parametric assumption would be incorrect. This is where the power of the transformer neural network shines. A transformer is a *mapping-first* [6] approach. It does not require explicit modeling of the underlying parametric model. Instead, as a universal function approximator [9], it directly learns the complex conditional distribution $p_\theta(d_2|d_1)$ from paired data alone.

In this context, the transformer's ignorance of an explicit hierarchical model is its blessing. It bypasses the impossible step of modeling the true hierarchical distribution of language and instead brute-forces the learning of the exact transformation required: the end-to-end process of posterior inference and subsequent synthesis. The self-distillation framework – synthesizing data from this conditional model and then training on it – is all that is needed. We never need to introduce an explicit hierarchical model to perform the forward $P(d|c)$ and backward pass $P(c|d)$ in the latent space. The entire procedure is implicitly carried through the synthesizer-tuning update with the latent concept c integrated, demonstrating a powerful insight for scaling LMs in the real world.

6 Discussion

Document embedding with activations of pretrained LM In our implementation of SBP, we use Qwen3-0.6B-Embedding [62] to obtain embeddings of DCLM [30] documents. An ideal implementation of SBP would only rely on the 3B-parameter model and the pretraining dataset itself to curate the paired synthesizer-tuning dataset. To achieve this, we can use the activations of the self-attention layer from an intermediate transformer block as a learned representation of documents. [26] and [60] implemented this at the much smaller scale of $\sim$ 300M parameters and $\sim$ 3B tokens. However, our experiments operate at a much larger scale with a customized model. As a result, we utilize the optimized vLLM [27] inference infrastructure for Qwen3-0.6B embedding models to efficiently index the pretraining corpus. Since the SBP procedure only requires a coarse binary decision of relevant vs. not relevant, which is much weaker than fine-grained document ranking embedding models are optimized for, we leave the more involved inference infrastructure for future work.

Parametric fit of SBP scaling law LM pretraining follows the scaling law [25, Equation 1.4] that relates the held-out test loss $L(N, D)$ to the number of LM parameters N and the size of the pretraining dataset D. In our experiments, we essentially evaluate $L(N, D)$ with $N =$ 3B at two different points $D =$ 10B and $D =$ 50B. There are two obstacles to a full scaling law for SBP: First, SBP is inherently a large-scale algorithm that cannot be scaled down. Since SBP synthesizes data itself, if the model and dataset sizes are too small, the generated text may not even be coherent. In contrast, experiments in [25] involve model sizes ranging from 768M to 1.5B and dataset sizes ranging from 22M to 23B, allowing for efficient experimentation. Second, varying N or D implies redoing the synthesizer-tuning and subsequent data synthesis over billions of tokens. Additionally, varying D also implies redoing the nearest neighbor matching. Obstacles aside, it would be interesting to see whether the SBP scaling law differs from the normal scaling law by a smaller multiplicative factor or a better exponent.

References

[1] Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, Mojan Javaheripi, Piero Kauffmann, Yin Tat Lee, Yuanzhi Li, Anh Nguyen, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Michael Santacroce, Harkirat Singh Behl, Adam Taumann Kalai, Xin Wang, Rachel Ward, Philipp Witte, Cyril Zhang, and Yi Zhang. Phi-2: The surprising power of small language models, 2023. URL https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

[3] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

[4] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1160.

[5] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426, 2021. URL https://arxiv.org/abs/2112.04426.

[6] Leo Breiman. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199 – 231, 2001. doi: 10.1214/ss/1009213726. URL https://doi.org/10.1214/ss/1009213726.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[8] Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation scaling laws. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1nEBAkpfb9.

[9] Emmanuel J Candès. Ridgelets: Theory and applications. *Department of Statistics, Stanford University*, 1998.

[10] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[11] Common Crawl. Common crawl. https://commoncrawl.org/, 2007.

[12] DatologyAI, :, Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, Charvi Bannur, Christina Baek, Darren Teh, David Schwab, Haakon Mongstad, Haoli Yin, Josh Wills, Kaleigh Mentzer, Luke Merrick, Ricardo Monti, Rishabh Adiga, Siddharth Joshi, Spandan Das, Zhengping Wang, Bogdan Gaza, Ari Morcos, and Matthew Leavitt. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining, 2025. URL https://arxiv.org/abs/2508.10975.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke

de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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

[16] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

[17] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pp. 3887–3896. PMLR, 2020.

[18] Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes, 2018. URL https://arxiv.org/abs/1709.08878.

[19] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[21] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[23] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018. URL https://arxiv.org/abs/1801.06146.

[24] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[25] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

[26] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

[27] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[28] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8546–8557, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9d8df73a3cfbf3c5b47bc9b50f214aff-Abstract.html.

[29] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[30] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024.

[31] Zonglin Li, Ruiqi Guo, and Sanjiv Kumar. Decoupled context processing for context augmented language modeling. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=02dbnEbEFn.

[32] Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: implicit bias matters for language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

[34] Yu. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, 2018. URL https://arxiv.org/abs/1603.09320.

[35] Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

[36] Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. Recycling the web: A method to enhance pre-training data quality and quantity for language models, 2025. URL https://arxiv.org/abs/2506.04689.

[37] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red

Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[39] David Owen. How predictable is language model benchmark performance?, 2024. URL https://arxiv.org/abs/2401.04757.

[40] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[41] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

[42] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018. URL https://arxiv.org/abs/1802.05365.

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

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

[46] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[47] Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.

[48] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[49] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LXVswInHOo.

[50] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

[51] Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *arXiv preprint arXiv:2412.02595*, 2024.

[52] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

[53] Philip Sun, David Simcha, Dave Dopson, Ruiqi Guo, and Sanjiv Kumar. Soar: improved indexing for approximate nearest neighbor search. *Advances in Neural Information Processing Systems*, 36:3189–3204, 2023.

[54] Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo,

Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

[55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[56] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data, 2024.

[57] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

[58] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *NUT@EMNLP*, 2017.

[59] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

[60] Zitong Yang, MICHAL LUKASIK, Vaishnavh Nagarajan, Zonglin Li, Ankit Rawat, Manzil Zaheer, Aditya K Menon, and Sanjiv Kumar. Resmem: Learn what you can and memorize the rest. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 60768–60790. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/bf0857cb9a41c73639f028a80301cdf0-Paper-Conference.pdf.

[61] Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=07yvxWDSla.

[62] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

[63] Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models, 2025. URL https://arxiv.org/abs/2506.10943.

[64] Zyphra. Zyda-2, a 5 trillion token high-quality dataset, 2024. URL https://huggingface.co/datasets/Zyphra/dclm-dedup.

CONTENTS

A Additional details on synthetic bootstrapped pretraining **20**

A.1 Data, model, and evaluation . . . 20

A.2 SBP implementation details . . . 21

A.3 Ablation on data mixture ratio . . . 22

B Additional analysis of synthesized samples **23**

B.1 Analyze concepts in documents . . . 23

B.2 Factuality analysis . . . 26

B.3 Mideval prompts . . . 28

B.4 Synthesized documents from the 1T-scale experiment . . . 31

C Additional pretraining results **31**

C.1 Two epochs validation . . . 31

C.2 Model scaling . . . 31

A Additional details on synthetic bootstrapped pretraining

A.1 Data, model, and evaluation

In this section, we present the complete details of the experiment set referenced in §3.

Dataset. A typical pretraining dataset is a mixture of different sources (e.g., GitHub, arXiv, CommonCrawl, etc.) with distinct sampling weights assigned to each constituent. We simplify this reality by considering a fixed document collection, which is a customized version of the DCLM dataset [30]. The original 4T token DCLM-baseline split contains roughly 80% duplicates, as reported by [64]. Therefore, we begin with the de-duplicated dataset, which consists of 769B tokens. We clean the raw Zyphra de-duplicated data by normalizing repeated line breaks, removing long URL links, and fixing malformed Unicode characters. For efficiency reasons, we cap the context window of the synthesizer-tuning (§3) step at 8,192 tokens. As a result, we additionally filter out the documents whose length is above 4,096 tokens, allowing both d_1 and d_2 to fit into the context window in the worst case when both documents are 4,096 tokens long. After all the de-duplication, cleaning, and filtering procedures, we end up with a collection of 582M high-quality documents $\mathcal{D}_{\text{pretrain}}$ totaling 482B tokens. We use the notation $|\mathcal{D}_{\text{pretrain}}|$ to denote the number of documents in the pretraining dataset and $\|\mathcal{D}_{\text{pretrain}}\|$ to denote the total number of tokens.

As a result of the high duplication rate in DCLM, for the 200B-scale experiment (introduced in §3), we implement the oracle upper bound as having access to 200B unique tokens from our document pool of size 482B tokens. For the 1T-scale experiment, we unfortunately do not have 1T unique documents due to the large fraction of duplicates from DCLM. As a surrogate, we utilize all 482B unique tokens as the dataset for training the oracle upper bound at the 1T-scale. We provide a partial justification for this by performing a scaled-down comparison at 400B training tokens, with one model having 400B unique tokens and the other one having 200B unique tokens repeated twice (§C.1). We find that the two models (400B unique and 200B repeated twice) yield nearly identical performance.

Architecture. We use the Llama 3 transformer architecture [14] to model the probability p_θ with the notable exception of implementing a QK-norm on top of the existing design, which we empirically find to stabilize training. Our resulting model is a 3B-parameter 26-layer transformer model with a hidden dimension of 3,072. Each layer employs grouped query attention with 24 query heads and 8 key/value heads. The position embedding is RoPE [52] for queries and keys, with frequency 5e+5. The feedforward network (FFN) has hidden dimension 8,064, and we apply prenorm to both the attention and FFN blocks. For tokenization, we implement a customized BPE tokenization with a vocabulary size of 49,152. To match the 8,192 context window design for synthesizer-tuning we have mentioned, we use context window 4,096 for pretraining, so that every document in $\mathcal{D}_{\text{pretrain}}$ can fit into the context window.

Benchmarks. To assess the pretraining capability of LM, we measure pretraining test loss and general world knowledge benchmarks. We evaluate held-out test perplexity (exponential of negative log-probability) on 1) OpenWebText2 from EleutherAI [44]; 2) Narrative understanding with LAMBADA [40] and 3) Broad domain multiple-choice with MMLU [20]. We evaluate QA accuracy on 4) Hard scientific reasoning with ARC-Challenge [10]; 5) Easy scientific reasoning with ARC-Easy [10]; 6) Scientific QA with SciQ [58]; 7) Common sense reasoning with Winogrande [48]; 8) Reading comprehension with TriviaQA [24]; 9) Openbook QA with WebQS [4]. We directly evaluate the pretrained model with either zero-shot or few-shot prompts. Although MMLU is more commonly known as a QA benchmark, we find that evaluating MMLU accuracy for weak models yields a highly non-smooth readout. As a result, for each MMLU test question, we prepend the question with a 5-shot example of QA pairs and postpend it with the correct answer. Then, we treat each such sample as a text corpus and evaluate LM's perplexity on such a text sample. Empirically, we find that this perplexity-based MMLU correlates well with MMLU accuracy when the underlying model is large enough to yield a stable readout, and also delivers smooth performance changes for smaller models. Note that those benchmarks are known to improve significantly with instruction finetuning [57]. However, we stick to our data-constrained setup and do not introduce any additional data that may confound the comparison.

A.2 SBP IMPLEMENTATION DETAILS

In this section, we present the implementation details of SBP outlined in §2.

Nearest neighbor pairing Recall from §3 that we work with a 3B-parameter transformer architecture and pretraining dataset at $\|\mathcal{D}_{\text{pretrain}}\|$ =10B and $\|\mathcal{D}_{\text{pretrain}}\|$ =50B scale. To take advantage of efficient ANN search at pretraining scale, we embed the documents from $\mathcal{D}_{\text{pretrain}}$ as 1,024 dimensional vectors using Qwen3-Embedding-0.6B. Then, we use ScaNN [17, 53] with 8-bit quantization to perform efficient similarity search. We adopt an asymmetric sharding to keys and value vectors. For each value vector, we build a ScaNN search tree with $\sqrt{N}$ leaves where N is the number of vectors in each value shard. To distribute the key shards across each search tree, we employ a "salting" strategy, where we create multiple copies of the ScaNN searcher and assign one key shard to each salted copy of the searcher (Figure 4). This design enables us to perform a top-200 nearest neighbor search over $|\mathcal{D}_{\text{pretrain}}|$ =60M documents within 155M CPU hours.

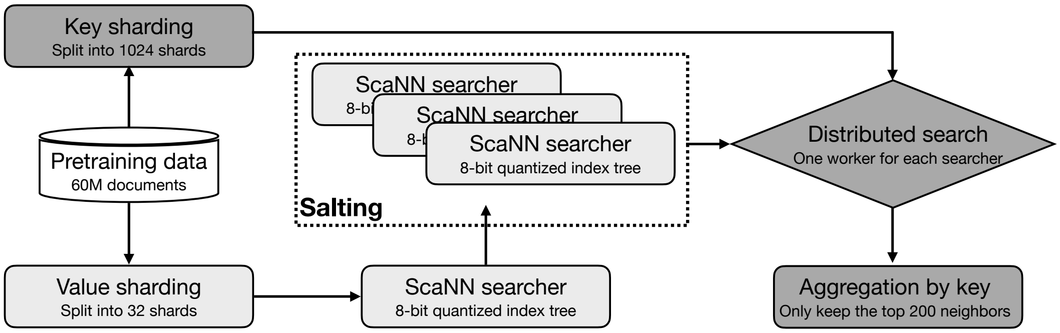
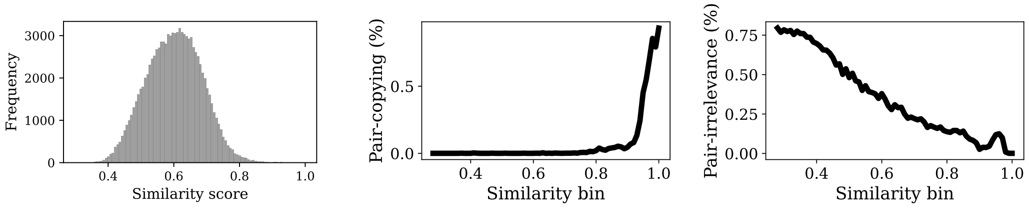


Figure 4: ScaNN system design for efficient distributed search.

At both the 200B-scale and 1T-scale, after obtaining the top 200 neighbors for each sample, we select the pairs whose similarity score is greater than 0.75. We chose this cut-off as it would later lead to a tractable size of synthesizer-tuning dataset $\mathcal{D}_{\text{ST}}$. To access the effect of choosing a different threshold, we provide a quantitative analysis of the fraction of relevant documents around each bin of similarity threshold in Figure 5 using the same metric defined in §4.2. We can see that a larger similarity score yields pairs with higher relevance but also more duplicates. Finally, we eliminate near-duplicates using a rule-based filtering approach. The dedup process involves first normalizing text by removing punctuation, converting to lowercase, and eliminating numbers, followed by tokenization using SentencePiece. We then generate "shingles" using 13-token sliding windows within d_1. Training pairs are discarded if any shingle from d_1 appears in d_2.



(a) Similarity histogram (b) Duplicate rate (c) Relevance rate

Figure 5: Analysis of paired data at 200B-scale. Figure 5(a): a histogram of 100K subsampled pairs grouped by their similarity score. Figure 5(b): the fraction of duplicate pairs when we subsample 1K pairs around a specific similarity score. Figure 5(c): same as 5(b) but showing the fraction of relevant documents.

Synthesizer-tuning After we collected the cleaned pair data $\mathcal{D}_{\text{ST}}$ (previous step), we perform the synthesizer-tuning with the objective (§3). We initialize the 3B-parameter at the baseline checkpoint and finetune the model with a constant learning rate of 5e-6 and a batch size of 16M tokens per step. Before we settled on this learning rate schedule, we first attempted the cosine decay schedule with a larger learning rate. We found that the generated text has lower quality than our final design with a small, constant learning rate. We measure the Pair-novelty score (defined in §4.2) of the synthesized example for different checkpoints of synthesizer-tuning, and find that longer training results in better Pair-novelty.

Synthesis at scale Finally, we perform the hierarchical sampling procedure defined in §2 with a temperature of 1.0 and `top_p` threshold 0.9. We apply a rule-based filtering that removes synthesized documents containing repeated occurrences of 13-token shingles. This effectively removes texts with repetition failure. We use vLLM [27] and obtain a throughput of 8.3K tokens per B200 second. This amounts to 2.5K B200 hours for the 200B-scale synthesis and 4.2K B200 hours for the 1T-scale synthesis.

A.3 Ablation on data mixture ratio

When performing joint training on a mixture of real and synthesized documents for the final SBP run, a natural question arises: how much fraction of synthesized documents to include. In §4, we discussed that we utilized $\|\mathcal{S}_{\text{pretrain}}\| = $75B for the 200B-scale experiment and $\|\mathcal{S}_{\text{pretrain}}\| = $125B for the 1T-scale experiment. In this section, we present ablation experiments for this design choice.

200B-scale At this smaller scale, we perform a comprehensive sweep over five possible values of $\|\mathcal{S}_{\text{pretrain}}\| \in \{$0B, 25B, 50B, 75B, 100B$\}$. As seen in Figure 6, different benchmarks exhibit varying behavior when more synthetic data is included during training: the perplexity (OpenWebText2 and LAMBADA) decreases monotonically with increasing synthetic data, while most QA benchmarks display a peak around $\|\mathcal{S}_{\text{pretrain}}\|$ = 75B.

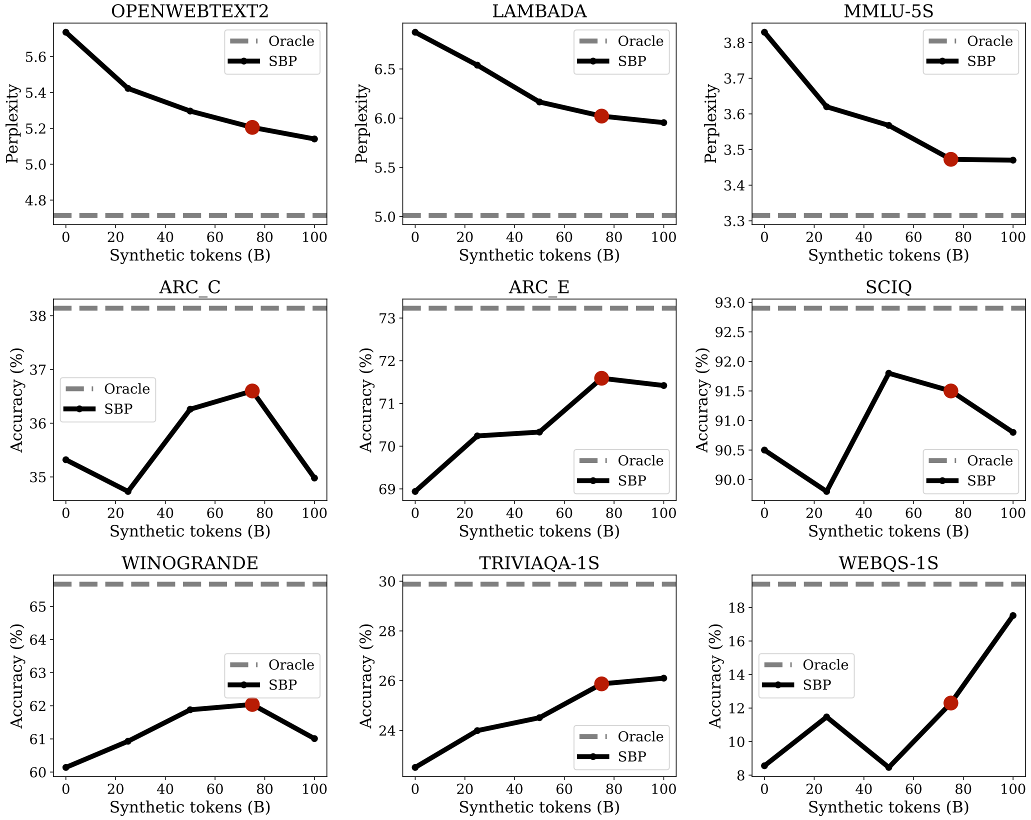


Figure 6: SBP performance with varying synthetic tokens at 200B-scale.

1T-scale At the 1T-scale, both data synthesis and subsequent joint pretraining become significantly more expensive. Therefore, we evaluate SBP at three different values of the synthetic data $\|\mathcal{S}_{\text{pretrain}}\| \in \{0\text{B}, 125\text{B}, 250\text{B}\}$. As shown in Figure 7, we find that $\|\mathcal{S}_{\text{pretrain}}\| = 125\text{B}$ produces the best-performing model across all benchmarks except LAMBADA perplexity.

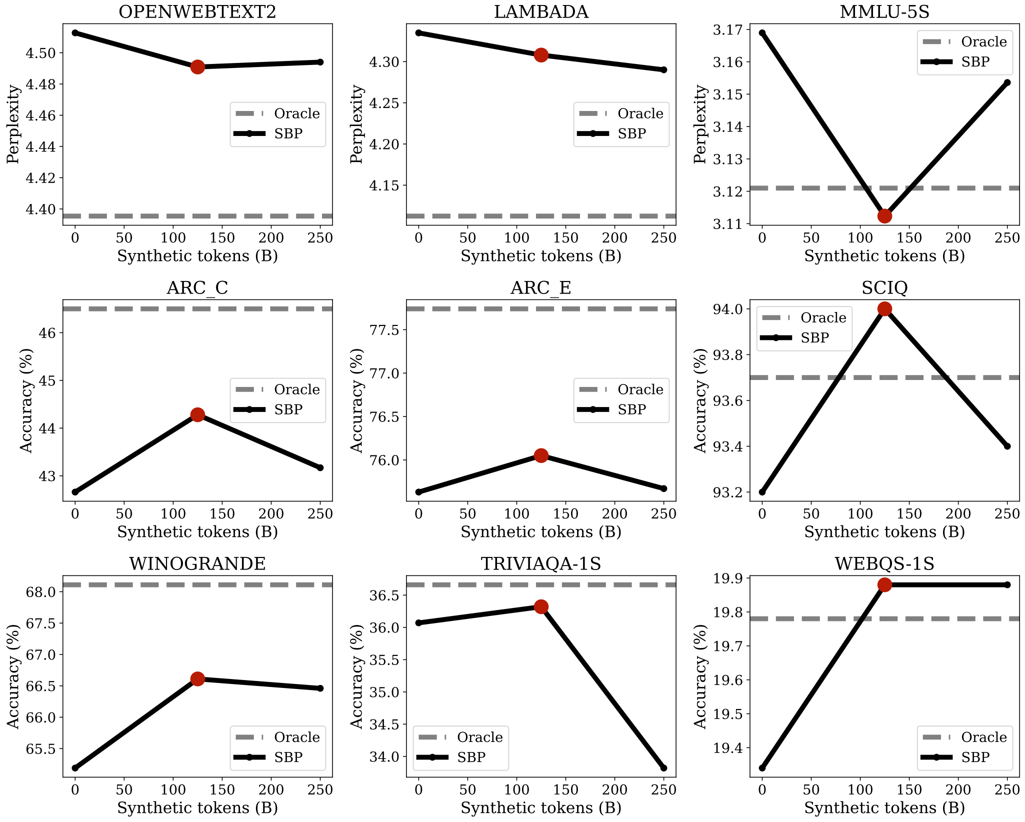


Figure 7: SBP performance with varying synthetic tokens at 1T-scale.

Discussion From this analysis, we can observe a general pattern: the best-performing model is achieved when pretraining is conducted on a mixture of real and synthetic data. Real internet data has higher quality and therefore merits more repetition. However, as repetition yields diminishing returns, synthetic data could offer another source of signal that real data cannot capture. In contrast, distillation-based research typically finds that training purely on synthetic data yields significantly higher training efficiency. However, this finding is obscured by the fact that such a model eventually converges to the capability of the teacher LM. This contrast reveals that the SBP mechanism does not generate a compressed and denoised representation of knowledge that is more efficient for LM to learn. Instead, it offers an additional source of improvement that real data alone cannot capture.

B Additional analysis of synthesized samples

B.1 Analyze concepts in documents

In this section, we further examine the intermediate hidden mechanisms underlying the document synthesis process. Specifically, we classify the hypothesized concepts inferred from real documents (see Table 3 for details) along two complementary dimensions: concept domains, which denote the broad subject areas or fields a concept belongs to (e.g., science, psychology, health, culture), and concept types, which capture the abstract role or nature of the concept itself (e.g., theory, method, comparison, symbol).

Table 3: Examples of latent concepts c inferred by an external LM (prompts provided in §B.1). From left to right, we provide a summary of the real document, the inferred latent concept, and a summary of the synthesized document.

Real document summary	**Concepts**	**Synthesized document summary**
Examination of Twitter's impact on journalism	Opportunities arise from Twitter	Guide on Twitter user monetization
Family story about kids and doughnuts	Parenting + kids' food catering	Emotional anecdotes of parents treating kids
Minor parties' challenges in the U.S. Congress	Minor political parties in the U.S.	Explains U.S. minor parties' history
Personal stories/questions about swollen eyes	Causes/treatments of swollen eyes	Non-personal guide to treating swollen eyes.
Antarctic carbon fixation mechanisms	How life survives in Antarctic	Antarctic geography and survival adaptations
Profile of a belly dancing teacher in the U.K.	Belly dancing as a dance form	General introduction to belly dancing
Anxiety about creative work judged in a dream	Dream as personal self-reflection	Description and reflection of personal dreams
NYC (yearly/monthly) climate extremes	NYC weather and temperature	QA on NYC July heat and related topics
Tutorial for Minecraft block modding	Block editing in Minecraft	Minecraft forum question on removing blocks
Cosmic airburst destroys Tall el-Hammam city	Destruction of ancient cities	Tall el-Hammam excavation as a news event

Table 4: Categorize extracted concepts into domains.

Concept Domains	**Examples**
Culture (38.74%)	Inter-community conflict in Nigeria, Family-based immigration policy, Reactions to Horrid Henry books, Interracial dating and bias
Health (11.89%)	Cosmetic dental appliance, Colistin toxicity in infections, Hair health tips, Portable/home medical diagnostics, Vitamin D and pregnancy outcomes
Technology (9.91%)	Recovering deleted phone data, Video editing app review, Flash platform pros and cons, HTML 2.0 draft process, Email attachment processing speed
Politics (3.69%)	Iran nuclear negotiations, Student loans policy reform, Democratic primary candidate choice, Catalan independence aftermath
Psychology (3.42%)	Differences in personality disorders, Exploring the strange in daily life, Aging and nostalgia, Toxic relationship breakup, Psychology research paper topics

The distribution of concept domains and types in Table 4 and 5 underscores the multidimensional nature of the knowledge space under consideration. The domains encompass macro-level socio-cultural phenomena, such as Culture, where topics range from inter-community conflict in Nigeria to immigration policy and interracial dating and bias, alongside micro-level issues of individual health and wellbeing, as exemplified in Health. In parallel, the typological classification reveals not only subject matter but also modes of conceptual engagement: Methods comprise formalized procedures (multidimensional poverty measurement, commercial real estate appraisal), Events capture historically situated crises (Mediterranean migrant crisis, BP oil spill nationalization), and Comparisons and Analyses facilitate interpretive framing through juxtapositions (cancer suffering: individual vs. family) and evaluative inquiries (Manchester United player analysis). Collectively, this taxonomy illustrates not only topical diversity but also a spectrum of cognitive orientations.

While real and synthesized documents share the same underlying concept, they differ in multiple ways that merit closer examination. We categorize these differences into a taxonomy of relations using a small ontology. Table 6 illustrates several relationship types, highlighting how synthesized data can reflect multiple facets that vary from real data. These relations range from scope-based distinctions (e.g., specific vs. general), to causal connections (e.g., corruption leading to reform), and to contrastive contrasts (e.g., Constitution articles vs. Articles of Confederation). This diversity demonstrates the rich variation structure that the synthesizer captures and learns.

Table 5: Categorize extracted concepts into abstract types.

Concept Types	**Examples**
Method (9.17%)	Multidimensional poverty measurement, Commercial real estate appraisal, Stop words search duplicates, DAT chemistry exam preparation
Event (6.98%)	Mediterranean migrant crisis, BP oil spill nationalization, Paula Abdul stalked, Eminem-Apple music rights lawsuit, Presidents Cup U.S. golf
Comparison (5.54%)	Hobbit film adaptation length/cost, Biking as superior transport, Cancer suffering, individual vs. family, Progress critique: 4G vs. alternatives
Analysis (5.20%)	Health effects of substances, Thai massage benefits, Scrabble word breakdown, Relationship roles and challenges, Manchester United player analysis
Phenomenon (4.95%)	Secret pain; self-destruction, Car-related online humor/pranks, Transnational corporations in globalization, Hippie identity and lifestyle

Table 6: Categorize relations between real documents d_1 and synthesized documents d_2.

Relation Categories	**Examples**
Scope relation (8.14%)	d_1: Probiotics' possible effects on H1N1 infection d_2: Probiotics' general digestive and immune benefits Relation: specific application vs general health benefits of probiotics
Perspectival relation (5.51%)	d_1: Personal, humorous struggles of new bloggers d_2: Objective guide to pros and cons of blogging Relation: subjective experiences vs objective guidance about blogging
Functional relation (4.70%)	d_1: Reviews and feedback on "Space Bound" game d_2: Forum troubleshooting for bugs in "Space Bound" Relation: reviews/feedback vs troubleshooting for the same game
Causal relation (2.05%)	d_1: DTEK faces corruption probe, financial risk d_2: DTEK nationalized for state-driven energy reform Relation: corruption/financial issues vs nationalization/energy reform
Contrastive relation (1.65%)	d_1: Detailed summary of Constitution articles d_2: Overview, flaws of Articles of Confederation Relation: U.S. Constitution articles vs Articles of Confederation: different foundational documents

Document Summarize and Concept Analysis Instructions

In the following, you are given two documents, doc1 and doc2. Doc2 is generated from doc1.

The principle of generation is to first abstract a concept from doc1, and then starting from this concept, generate doc2. Can you guess what this concept is and how doc2 was generated?

Please keep the summary and concepts to be LESS OR EQUAL TO 10 WORDS and format your answer as follows. Highlight the difference between doc2 and doc1 in your doc2_summary:

```
<doc1_summary> summary of doc1 </doc1_summary>
<concept_c> abstract concept from doc1 </concept_c>
<doc2_summary> summary of doc2 built on doc1 given the concept </doc2_summary>
```

Example 1:

```
<doc1_summary> recommendation of local coffee shops in San Diego </doc1_summary>
<concept_c> coffee + San Diego </concept_c>
<doc2_summary> comparison of coffee culture in SD and NYC </doc2_summary>
```

Example 2:

```
<doc1_summary> Patient with swollen eye discusses pain causes & symptoms and seeks for
advice </doc1_summary>
<concept_c> medical symptom of swollen eye </concept_c>
<doc2_summary> A wiki-style article introducing causes and cures for swollen eye
</doc2_summary>
```

Now, give your answer for the following documents:

```
<doc1>
{real_document}
</doc1>

<doc2>
{synthesized_document}
</doc2>
```

B.2 Factuality analysis

Table 7: Estimation of the ratio of non-factual documents. We can see that the occurrence factuality error decays as the SP scales up.

	Factuality undefined	No factual error	Factual error
Real data	31.44%	66.74%	1.81%
Synthetic data (200B-scale)	34.43%	50.47%	15.09%
Synthetic data (1T-scale)	31.91%	59.43%	8.65%

All LM generated synthetic data face the limitation of potentially generating non-factual content due to their probabilistic modeling nature. Moreover, because the internet inherently contains factual inaccuracies, LMs are likely to absorb these errors unless the data is carefully cleaned. During post-training, factuality must also be recalibrated alongside other objectives such as data safety.

SBP relies solely on document-level correlations and does not incorporate human intervention to filter non-factual content. As a result, the generated outputs are also expected to contain factual errors. Interestingly, we observe that the frequency of such errors correlates with the amount of data used in the SBP pipeline. We define a document as having **undefined factuality** if it is primarily subjective or opinion-driven, or if it concerns personal, obscure, or unverifiable entities. In all other cases, the document's factuality is considered **well-defined** and verifiable.

In Table 7, we analyze both the real data and the synthesized data used in the main experiment presented in Section 4.1. Specifically, we consider two types of synthetic datasets: a smaller-scale set initialized with 10B seed tokens, and a larger-scale set initialized with 50B seed tokens. From each source: real data, smaller-scale synthetic data, and larger-scale synthetic data, we randomly sample 10k documents. Each document is then categorized into three bins: **factuality undefined**, **no factual error**, and **factual error**, using LM-as-a-judge. Our analysis shows that synthetic data contains more factual errors than real data. However, as the amount of seed data increases, the factuality of synthetic data improves significantly, approaching that of real data. This finding is consistent with our mideval results in Table 2, where greater seed data availability enables the LM

Table 8: Factuality undefined synthetic text.

Synthetic Text	Sunday, December 28, 2008 Tante Leni Tante Leni is not only my Aunt Leni; she is my Eternity. When my Aunt Leni died a few years ago, she was deeply saddened and I was devastated. She was not the first family member to die, but she was the first I felt so strongly about. Tante Leni was all my parents really had to show for the 25 years they worked as public school teachers and she was the one who had been with them the longest. There was a special place in her heart for my parents. In addition to all that, she was the kind of person who always had a smile and a funny story to share. She was kind and funny and generous. The story that always comes to mind when I think about her is the time she was working at the bank and someone dropped something from the top floor. It was a very large parcel and the workers on the ground floor didn't have the tools to open it. She jumped down to see what was in it. A very large package of champagne appeared and she began gulping it down. Tante Leni and my mother in a portrait they took when my mom was 20. Tante Leni and my parents in a family portrait she took for my mom at 22. Tante Leni and my dad at home when he was working as a dance instructor. When my mom died, she had all the people who had known her since she was a child living in the house. Tante Leni was the oldest, but she was also the best at cleaning, cooking and taking care of the house. When my mom passed away, she went to a rehab center and Tante Leni stayed in the house.

to capture more factual knowledge and the synthesizer tuning to generate more relevant documents, thereby reducing hallucinations and producing more realistic outputs.

We extend our analysis of factuality errors in synthesized data in Table 9, highlighting the inaccuracies present in the synthetic texts. These include false transfer and timeline claims in football, as well as incorrect institutional, company location, and certification details in the ecolabel example. This underscores the importance of rigorous fact-checking, particularly in areas such as historical events (e.g., sports) and certification standards (e.g., eco-labels).

Factuality detection instructions

You are a helpful AI assistant. Your task is to evaluate whether the given document has well-defined factuality.

Definitions:

Not well-defined factuality: The document is primarily subjective or opinion-based (e.g., express disapproval of a politician in social media). The document discusses personal, unknown, or unverifiable entities (e.g., a private diary).

Well-defined factuality: The document refers to well-known, identifiable entities (e.g., famous people, historical events, popular movies). Its factual claims can be checked or verified.

Output format:

If the document's factuality is not well-defined, output:

`<not well defined></not well defined>`

If the document's factuality is well-defined and factual, output:

`<well defined>True</well defined>`

If the document's factuality is well-defined but non-factual, output:

`<well defined>False</well defined>`

Now, analyze the following document and provide your answer:

`{document}`

B.3 Mideval prompts

Before each large-scale synthesis run (on the order of billions of tokens), we begin by synthesizing a small subset of data to evaluate its overall quality, a step we refer to as “mideval”. The goal is to maximize the Pair-relevance of the generated data while monitoring Pair-novelty and Non-repetition rates. Although near-duplicates may not directly degrade quality, they reduce the data’s overall utility, so we aim to minimize their occurrence. While self-repetition can be removed via rule-based filtering, we still track it as an indicator of the synthesizer’s quality. The quality of both the paired training data for synthesizer-tuning and the synthesized document influences the performance of the final model.

We have cited mideval results in many sections throughout the paper. In this section, we present the prompt that was used for Pair-novelty, Pair-relevance, and Non-repetition.

Pair-relevance detection

You are a helpful AI assistant helping the user to determine if two provided texts are relevant to each other.
The user will provide you two texts in the following format:

```
## Text 1
{text1}

## Text 2
{text2}
```

Your job is to determine if the two texts are relevant enough to be considered as a pair. Relevance means that the two texts are about the same topic, event, entity, person, place, or thing. If two texts talks about completely unrelated topics, they are not relevant.
Please explain your reasoning in your response, and conclude the response in a new line with either ”Yes” or ”No”. Do not end with any other text including punctuation.

Pair-novelty detection

You are a helpful AI assistant helping the user to determine if two provided texts are near duplicates.
The user will provide you two texts in the following format:

```
## Text 1
{text1}

## Text 2
{text2}
```

Your job is to determine if the two texts are near duplicates, which means they are almost identical, except for some extra white spaces, line breaks, or punctuation. Two texts are not near duplicates if they talk about the same topic but use different language, words, or style.
Please explain your reasoning in your response, and conclude the response in a new line with either ”Yes” or ”No”. Do not end with any other text including punctuation.

Non-repetition detection

You are a helpful AI assistant helping the user to determine if the provided text has repetition issues.
The user will provide you a text in the following format:

```
## Text
{text}
```

Your job is to determine if the text has repetition issues, which means some particular sentence or pattern are repeated more than three times. Some examples of problematic text:

```
## Example 1 of problematic text
I have a list of users in a SharePoint 2010 site.
I want to send an email to all of them.
I have tried the following code:
var email = new MailMessage();
email.From = new MailAddress("");
email.To = new MailAddress("");
email.Subject = "Test";
email.Body = "Test";
var smtp = new SmtpClient();
smtp.Host = "";
smtp.Port = 25;
smtp.Credentials = new NetworkCredential("", "");
smtp.Send(email);
I get the following error:
The server could not send the message.
The server could not send the message.
The server could not send the message.
...
```

```
## Example 2 of problematic text
 my Profile
Product Reviews - Send Message
You are responding to the following review:
Submitted: 02-11-2006 by mikeschmid
I have been paddling for 10 years and have owned 10 kayaks.
I have been paddling in the ocean for 5 years and have owned 3 kayaks.
I have been paddling in the ocean in the Pacific Northwest for 3 years and have owned
2 kayaks.
I have been paddling in the ocean in the Caribbean for 2 years and have owned 1 kayak.
I have been paddling in the ocean in the Mediterranean for 1 year and have owned 1 kayak.
I have been paddling in the ocean in the South Pacific for 1 month and have owned 1 kayak.
I have been paddling in the ocean in the South Atlantic for 1 month and have owned
1 kayak.
I have been paddling in the ocean in the Indian Ocean for 1 month and have owned 1 kayak.
I have been paddling in the ocean in the Arctic Ocean for 1 month and have owned 1 kayak.
I have been paddling in the ocean in the Antarctic Ocean for 1 month and have owned
1 kayak ...
```

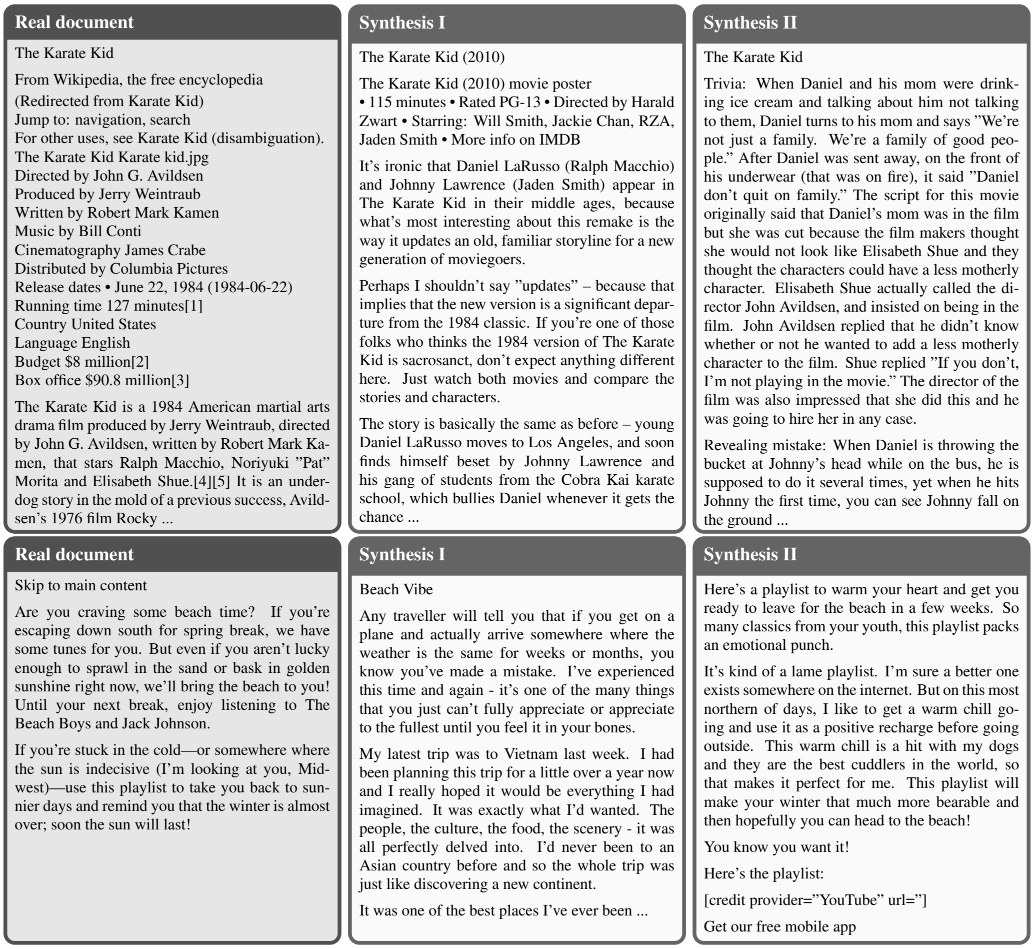
Please explain your reasoning in your response, and conclude the response in a new line with either “Yes” (which means the text has repetition issues) or “No” (which means the text does not have repetition issues). Do not end with any other text including punctuation.

Table 9: Factuality errors detected in synthetic text.

Synthetic Text	So just how much has Chelsea been prepared to pay for the 34-year-old midfielder? Realistically, the clubs involved should be keeping in the region of £25 million ($38.8 million) and around £30 million ($45.5 million) being bandied about for the player in Italy. With the Blues expected to complete the sale of Cesc Fabregas to Arsenal this week, Lampard appears the logical replacement in midfield, but his bid to extend his contract has hit a roadblock with Chelsea's owners Roman Abramovich and the club being unable to agree to an increase in salary. Lampard, who played in the Champions League final in Lisbon for Chelsea in 2007, has been linked with a move away from Stamford Bridge this summer, after having his contract with the club indefinitely extended in 2010. There were rumours that he was to leave on a free transfer in the summer of 2011, but it appears that those are again being placed to the side. Miguel Delaney, from Sky Sports, believes that Lampard could be the highest paid player in the Premier League, and puts the bid that Chelsea are offering him in the region of £6 million ($8.9 million) per year. If Lampard is to join a team in Italy, it seems that this sum would be realistic, but the Premier League is a different beast, and it would be unrealistic for Chelsea to splash out that amount. As the speculation around his future continues, you can watch the pair discussing the topic in the video below.
Factual Errors	Cesc Fabregas to Arsenal – Fabregas moved from Arsenal to Barcelona in 2011, then to Chelsea in 2014, not back to Arsenal. 2007 – The 2014 Champions League final in Lisbon did not involve Chelsea or Lampard; Chelsea won in 2012. indefinitely extended in 2010 – Lampard did not get an indefinite contract extension in 2010 with Chelsea.
Synthetic Text	Swanee Glue Brand: First Glue to be Awarded the Swan Ecolabel Published:27 July 2022 The global glue stick market is expected to reach USD 3.45 billion by 2028. Adhesives are the first choice of manufacturers in all industries such as food, pharmaceuticals, automotive, aerospace, construction, and packaging. As consumers are increasingly conscious of their carbon footprint and environmental issues, glue manufacturers are aiming to produce products that comply with environmental standards and are effective and cost-effective in their applications. This is why the Swan Ecolabel was established by the Swedish Environment Agency as a certification for sustainable adhesive products. Swanee Glue is one of the world's leading glue brands in glue sticks, and this year its brand received the Swan Ecolabel. UHU is an adhesive brand owned by Bolton Adhesives in the Netherlands, and part of the Italian Bolton Group with a strong agenda for sustainability. Glue sticks, specifically glue sticks with a wider applicator and swan neck applicators, have the most impact on the environment because they are a consumable item and their impact is greatest when thrown away. Therefore, the Swanee Swan Ecolabel ensures that UHU is part of the solution to the growing demand for sustainable adhesive products. In order to obtain the Swan Ecolabel, the adhesive must have at least 50% renewable content. Besides this, the glue stick should also contain a higher percentage of recyclable content. UHU meets all these criteria and has a permanent and multi-use applicator. For further information, you can contact UHU receives the Swan Ecolabel
Factual Errors	Swan Ecolabel was established by the Swedish Environment Agency – The Nordic Swan Ecolabel was established by the Nordic Council of Ministers, not only Sweden. Netherlands – UHU is based in Germany, not the Netherlands. 50% renewable content – The Swan Ecolabel requires at least 20% renewable content in adhesives, not 50%.

B.4 Synthesized documents from the 1T-scale experiment

In this section, we present additional examples of synthesized documents at the 1T-scale to complement the example given at the 200B-scale in Section 4.2.



Real document

The Karate Kid

From Wikipedia, the free encyclopedia

(Redirected from Karate Kid)
Jump to: navigation, search
For other uses, see Karate Kid (disambiguation).
The Karate Kid Karate kid.jpg
Directed by John G. Avildsen
Produced by Jerry Weintraub
Written by Robert Mark Kamen
Music by Bill Conti
Cinematography James Crabe
Distributed by Columbia Pictures
Release dates • June 22, 1984 (1984-06-22)
Running time 127 minutes[1]
Country United States
Language English
Budget $8 million[2]
Box office $90.8 million[3]

The Karate Kid is a 1984 American martial arts drama film produced by Jerry Weintraub, directed by John G. Avildsen, written by Robert Mark Kamen, that stars Ralph Macchio, Noriyuki ”Pat” Morita and Elisabeth Shue.[4][5] It is an underdog story in the mold of a previous success, Avildsen’s 1976 film Rocky ...

Synthesis I

The Karate Kid (2010)

The Karate Kid (2010) movie poster
• 115 minutes • Rated PG-13 • Directed by Harald Zwart • Starring: Will Smith, Jackie Chan, RZA, Jaden Smith • More info on IMDB

It’s ironic that Daniel LaRusso (Ralph Macchio) and Johnny Lawrence (Jaden Smith) appear in The Karate Kid in their middle ages, because what’s most interesting about this remake is the way it updates an old, familiar storyline for a new generation of moviegoers.

Perhaps I shouldn’t say ”updates” – because that implies that the new version is a significant departure from the 1984 classic. If you’re one of those folks who thinks the 1984 version of The Karate Kid is sacrosanct, don’t expect anything different here. Just watch both movies and compare the stories and characters.

The story is basically the same as before – young Daniel LaRusso moves to Los Angeles, and soon finds himself beset by Johnny Lawrence and his gang of students from the Cobra Kai karate school, which bullies Daniel whenever it gets the chance ...

Synthesis II

The Karate Kid

Trivia: When Daniel and his mom were drinking ice cream and talking about him not talking to them, Daniel turns to his mom and says ”We’re not just a family. We’re a family of good people.” After Daniel was sent away, on the front of his underwear (that was on fire), it said ”Daniel don’t quit on family.” The script for this movie originally said that Daniel’s mom was in the film but she was cut because the film makers thought she would not look like Elisabeth Shue and they thought the characters could have a less motherly character. Elisabeth Shue actually called the director John Avildsen, and insisted on being in the film. John Avildsen replied that he didn’t know whether or not he wanted to add a less motherly character to the film. Shue replied ”If you don’t, I’m not playing in the movie.” The director of the film was also impressed that she did this and he was going to hire her in any case.

Revealing mistake: When Daniel is throwing the bucket at Johnny’s head while on the bus, he is supposed to do it several times, yet when he hits Johnny the first time, you can see Johnny fall on the ground ...

Real document

Skip to main content

Are you craving some beach time? If you’re escaping down south for spring break, we have some tunes for you. But even if you aren’t lucky enough to sprawl in the sand or bask in golden sunshine right now, we’ll bring the beach to you! Until your next break, enjoy listening to The Beach Boys and Jack Johnson.

If you’re stuck in the cold—or somewhere where the sun is indecisive (I’m looking at you, Midwest)—use this playlist to take you back to sunnier days and remind you that the winter is almost over; soon the sun will last!

Synthesis I

Beach Vibe

Any traveller will tell you that if you get on a plane and actually arrive somewhere where the weather is the same for weeks or months, you know you’ve made a mistake. I’ve experienced this time and again - it’s one of the many things that you just can’t fully appreciate or appreciate to the fullest until you feel it in your bones.

My latest trip was to Vietnam last week. I had been planning this trip for a little over a year now and I really hoped it would be everything I had imagined. It was exactly what I’d wanted. The people, the culture, the food, the scenery - it was all perfectly delved into. I’d never been to an Asian country before and so the whole trip was just like discovering a new continent.

It was one of the best places I’ve ever been ...

Synthesis II

Here’s a playlist to warm your heart and get you ready to leave for the beach in a few weeks. So many classics from your youth, this playlist packs an emotional punch.

It’s kind of a lame playlist. I’m sure a better one exists somewhere on the internet. But on this most northern of days, I like to get a warm chill going and use it as a positive recharge before going outside. This warm chill is a hit with my dogs and they are the best cuddlers in the world, so that makes it perfect for me. This playlist will make your winter that much more bearable and then hopefully you can head to the beach!

You know you want it!

Here’s the playlist:

[credit provider=”YouTube” url=”]

Get our free mobile app

Figure 8: Comparison of original text with synthesized text variations. On the first row, the real document provides factual information about the 1984 film’s production and release. In contrast, the synthesized documents offer subjective commentary, opinions, and behind-the-scenes anecdotes about both the 1984 film and its 2010 remake. On the second row, the synthesized documents are continuations of the real document.

C Additional pretraining results

C.1 Two epochs validation

When designing the oracle experiment for 1T-scale, we noted that we use 482B tokens repeated twice as a proxy for training on 1T unique tokens. This is because the DCLM-baseline [30] dataset contains 80% duplicates, which hinders our evaluation. We validate our choice by scaling down the experiment to a 400B scale, where we had sufficiently many unique tokens. As seen in Table 10, 200B tokens repeated twice yield nearly identical performance to 400B unique tokens. This finding is consistent with the observation from [35] where repetition up to 4 times yields nearly no performance degradation.

C.2 Model scaling

An alternative approach to leveraging additional compute is to use a larger model. In this section, we examine the benefits of fixing a training token budget, but using a 6B-parameter model (Table 11).

Table 10: Performance comparsion with 200B tokens repeated twice vs. 400B unique tokens for the 3B model. We can see that the two models yield similar performance.

Benchmark	**2x200B**	**1x400B**
Perplexity on held-out data ↓		
OpenWebText2	4.55	4.54
LAMBADA	4.49	4.46
Five-shot MMLU	3.19	3.17
QA accuracy ↑		
ARC-Challenge (0-shot)	38.31	41.47
ARC-Easy (0-shot)	73.11	75.29
SciQ (0-shot)	93.80	93.30
Winogrande (0-shot)	64.96	63.93
TriviaQA (1-shot)	32.51	34.35
WebQS (1-shot)	18.75	13.58
Average QA accuracy	53.57	53.65

We conduct a pretraining experiment in a 200B-scale setting, replacing a 3B-parameter model with a 6B-parameter model. In Table 12, we observe that the 6B-parameter model consistently outperforms the baseline method, indicating that it effectively utilizes the additional computational resources available. Comparing SBP with the 6B-parameter model, we see that one performs better on some benchmarks while the other performs better on others. This suggests the benefits offered by SBP are orthogonal to the benefits provided by having a larger model, offering the potential to combine both approaches to obtain an even better model.

Table 11: 6B-parameter model setup.

Total Params.	**3B**	**6B**
ℓ_{context}	4096	4096
n_{vocab}	49152	49152
n_{layers}	26	32
d_{model}	3072	4096
d_{ffn}	8064	13056
n_{heads}	24	32
$n_{\text{kv\_heads}}$	8	8

Table 12: 200B-scale experiments with model scaling. The first three columns are identical to Table1. The last column shows the performance of training a 6B model under 200B training token budget with 10B unique tokens.

Benchmark	**Baseline**	**SBP**	**Oracle**	**6B-model**
Perplexity on held-out data ↓				
OpenWebText2	5.74	-0.53	-1.02	-0.36
LAMBADA	6.87	-0.85	-1.86	-1.10
Five-shot MMLU	3.83	-0.36	-0.51	-0.13
QA accuracy ↑				
ARC-Challenge (0-shot)	35.32	+1.28	+2.82	+3.42
ARC-Easy (0-shot)	68.94	+2.65	+4.29	+0.67
SciQ (0-shot)	90.50	+1.00	+2.40	+0.80
Winogrande (0-shot)	60.14	+1.90	+5.53	+2.92
TriviaQA (1-shot)	22.51	+3.36	+7.37	+3.11
WebQS (1-shot)	8.56	+3.74	+10.83	+5.22
Average QA accuracy	**47.66**	**+2.32**	**+5.54**	+2.69