Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling

Anonymous ACL submission

Abstract

Large language models are trained on massive scrapes of the web, which are often unstructured, noisy, and poorly phrased. Current scaling laws show that learning from such data requires an abundance of both compute and data, which grows with the size of the model being trained. This is infeasible both because of the large compute costs and duration associated with pre-training, and the impending scarcity of high-quality data on the web. In this work, we propose **W**eb **R**ephrase **A**ugmented **P**re-training (**WRAP**) that uses an off-the-shelf instruction-tuned model prompted to paraphrase documents on the web in specific styles such as "like Wikipedia" or in "question-answer format" to jointly pre-train LLMs on real and synthetic rephrases. First, we show that using **WRAP** on the C4 dataset, which is naturally noisy, speeds up pre-training by $\sim 3\times$. At the same pre-training compute budget, it improves perplexity by more than 10% on average across different subsets of the Pile, and improves zero-shot question answer accuracy across 13 tasks by more than 2%. Second, we investigate the impact of the re-phrasing style on the performance of the model, offering insights into how the composition of the training data can impact the performance of LLMs in OOD settings. Our gains are attributed to the fact that re-phrased synthetic data (i) incorporates style diversity that closely reflects downstream evaluation style, and (ii) has higher 'quality' than web-scraped data.

1 Introduction

Large language model (LLM) pre-training has been largely democratized and open-sourced, allowing various academic labs, and industries to pre-train custom LLMs. Yet, a key differentiator between these models is the composition and size of the data used to train them. Data curation strategies are required to filter out scrapes of the web that are unstructured and/or poorly phrased (Eisenstein, 2013). While some of these strategies have been made public (Brown et al., 2020; Wenzek et al., 2019; Penedo et al., 2023), most state-of-the-art data curation techniques are unknown to the research community, and only anecdotal evidence remains. Research on data curation requires multiple rounds of re-training, making it an expensive endeavour to document techniques that lead to practical improvements. On the other hand, scaling laws for language models (such as Chinchilla scaling laws (Hoffmann et al., 2022)) show that with increasing model sizes, we should also increase both the training compute and data size linearly. This is infeasible because (a) high-quality data is limited (Villalobos et al., 2022), and repeating for even a small number of epochs (4 or more) results in diminishing returns or overfitting (Muennighoff et al., 2023; Touvron et al., 2023; Xue et al., 2023); and (b) pre-training for such long durations is prohibitively expensive.

Meanwhile, the use of synthetic data has gained prominence in the paradigm of aligning pre-trained LLMs via instruction fine-tuning, RLHF (Ouyang et al., 2022), and instruction backtranslation (Li et al., 2023b). Recently, in the context of pre-training, synthetic data was used to generate datasets such as Tiny Stories (Eldan and Li, 2023) and Textbook quality synthetic data (Gunasekar et al., 2023; Li et al., 2023c). These were used to train smaller language models (like the Phi model family) that were as performant as larger language models on certain tasks. However, their data generation process stays largely opaque, and prohibitively expensive, requiring prompting a GPT-3.5 model for generating billions of tokens. Additionally, such data generation can create "knowledge bias" by generating data pertaining to tasks that we want to perform well on. While synthetic data has shown promise, it is unclear if this is because of its higher quality, or strategic topic selection (Maini, 2023).

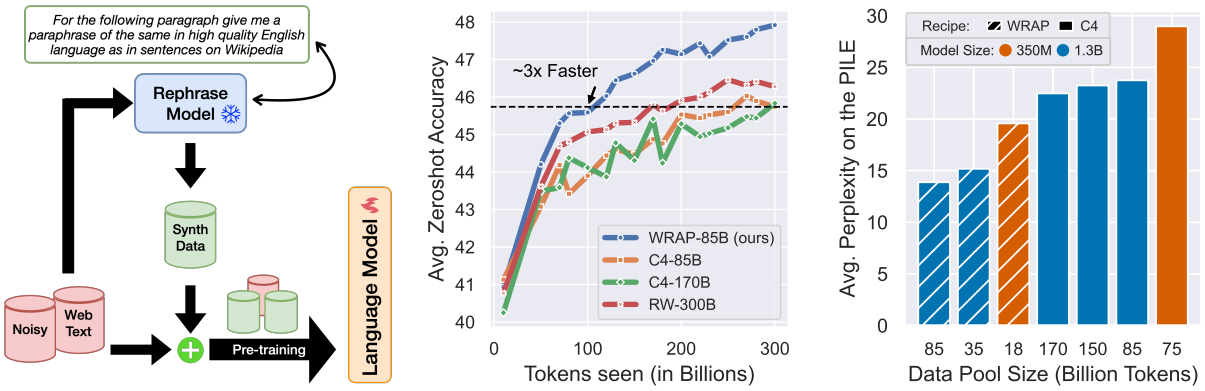


Figure 1: (a) **WRAP**: We prompt an off-the-shelf instruction-tuned model to rephrase articles on the web, and pre-train an LLM on a mixture of real and synthetic data. (b) Zero-shot performance of GPT 1.3B models trained on combinations of C4 and synthetic variations. (c) Weighted average perplexity over 21 sub-domains of the Pile for varying model sizes and amount of pre-training data.

In this work, we propose **W**eb **R**ephrase **A**ugmented Pre-training (**WRAP**)—that attempts to bridge three important challenges stemming from the ambiguity around data curation— (i) what data should you pre-train on? (ii) how can you pre-train with limited data? (iii) how can you pre-train computationally efficiently? In particular, we show that re-phrasing documents on the web using an off-the-shelf medium size LLM allows models to learn much more efficiently than learning from raw text on the web, and accounts for performance gains on out of distribution datasets that *can not* be offset with additional web data. Our proposed method involves using a pre-trained off-the-shelf LLM to re-phrase documents from a web corpus into different styles (Figure 1a).

In our work, we tackle two important challenges faced during synthetic data curation in the works of Gunasekar et al. (2023)—generation cost, and data bias—by rephrasing articles on the web. (i) **WRAP** allows for using an open source, and much smaller LLM (1.8B/7B v/s GPT3.5) to rephrase unstructured and poorly phrased documents in different styles, since it does not rely on the LLM as a knowledge bank. (ii) Thanks to the information maintaining nature of rephrasing, we are able to leverage the natural diversity of the web rather than relying on an LLM for information, which may be prone to factual errors, and data biases. **WRAP** shows that the "style" alone can result in large gains in pre-training efficiency, and downstream performance.

Using **WRAP** on the C4, we evaluate model performance on 13 different zero-shot tasks, and 21 different language modeling domains of the Pile, and find that pre-training LLMs with synthetic data allows us to train equivalent models with 5x lesser data, or 3x lesser compute. In fact, our models, also outperform the recent TinyLLama models that were trained for 3 trillion tokens (10x data and compute) across several zero-shot Q/A tasks. We further observe a reduction in perplexity by $\sim 50\%$ on the Pile, and note that our 350M parameter model trained on combinations of real and synthetic rephrases on just 15% of the entire C4 corpus, outperforms pre-training a 1.3B parameter on the entire C4. Finally, we conduct an analysis on the potential of data leakage, properties of synthetic data styles, and how to combine synthetic data for improving **WRAP** based LLM pre-training.

2 Related Work

Neural Scaling Laws for Language Models Neural scaling laws relate the optimal number of model parameters and amount of training data for a fixed amount of compute. Hoffmann et al. (2022) presented the Chinchilla scaling laws for language models demonstrating that there was a linear relationship between the size of the model and the amount of training data needed. Their findings indicated that prior models such as Gopher (Rae et al., 2021) are severely undertrained. Recently, models such as Llama (Touvron et al., 2023) are trained with much more data. These scaling laws were drawn for the paradigm of single-epoch training. Recently, Muennighoff et al. (2023) showed that

the marginal utility of repeated data rapidly diminishes when training for more than 4 epochs, and formulated scaling laws under repeated data. Concurrently, Xue et al. (2023) showed that repeating even small fractions of the pre-training data can lead to overfitting and reduce model performance.

Dataset Selection Selecting high quality data for pre-training LLMs remains an active, high-impact, yet understudied area of research. For instance, GPT-2 model was pre-trained on all outbound links from Reddit, a social media platform, which received at least 3 karma (Brown et al., 2020). This was used as a heuristic indicator that the document may be *interesting, educational, or just funny*. Follow-up works have used other heuristics such as prioritizing documents that resemble wikipedia (Gururangan et al., 2022). Rae et al. (2021) used multiple heuristic filters to remove documents, such as the absence of certain stopwords, length of the document, percentage of alphabetic characters, mean word length, symbol-to-word ratio, percentage of lines starting with a bullet point, or ending with an ellipsis etc. Their work highlights the intricacies of filtering out text data. An alternative paradigm for building better datasets for training is to distill high-quality datasets. Xie et al. (2023) proposed a method, DoReMi, to select the best data mixture for pre-training language models by reweighting data from various domains. Concurrently, Abbas et al. (2023) showed that de-duplicating pre-training data can improve pre-training efficiency. Recently several methods were proposed for automatic filtering of low-quality data for faster fine-tuning of LLMs (Chen et al., 2023; Solaiman and Dennison, 2021; Zhou et al., 2023). Simultaneously, in the realm of image-language models such as CLIP (Radford et al., 2021), the Datacomp benchmark (Gadre et al., 2023) and recent entries (Maini et al., 2023; Yu et al., 2023) developed approaches at filtering out low-quality subsets from pre-training datasets like LAION (Schuhmann et al., 2022), or the common crawl.

Data Augmentation and synthetic data Eldan and Li (2023) showed that a synthetically generated dataset in the form of stories that toddlers can understand allows training a small language model that can generate coherent sentences. Gunasekar et al. (2023) showed that textbook quality (synthetic) data alone helps models achieve state-of-the-art performance on reasoning and coding tasks. Similar approaches are used in concurrent work for enhancing coding and mathematical reasoning abilities while finetuning (Liu et al., 2023a; Wei et al., 2023). Shumailov et al. (2023) show that training on synthetic data can actually be harmful for model performance, especially when we do multiple rounds of pre-training an LLM and then training the next LLM on data generated by the previous one. On the other hand, some other works have shown that such a strategy can actually be useful. Li et al. (2023b) and Köksal et al. (2023) discuss how a model can generate instruction data and then fine-tune on its own generated data to improve performance. Jung et al. (2023) discuss how such repeated cycles of synthetic data can help train a very small paraphrase model that outperforms GPT-3.

The vision and multimodal literatures have also seen a surge of works examining the use of synthetic data for training. The works of Bansal and Grover (2023); Trabucco et al. (2023); Azizi et al. (2023) have shown that using synthetic data in combination with real data achieves state of art model performance both in and out-of-distribution. Cubuk et al. (2020) used generative models to generate image augmentations for better domain generalization. There are also multiple studies on increasing multiplicity of augmentations and their value for improving generalization (Choi et al., 2019; Fort et al., 2021; Hoffer et al., 2020). However, Alemohammad et al. (2023) showed that generated models trained for more than five cycles of their own generated data can undergo mode collapse.

3 WRAP: Web Rephrase Augmented Pretraining

Generating synthetic data using an off-the-shelf language model can be both computationally expensive and operationally challenging. Prior approaches to generating synthetic textbook quality data using LLMs (Gunasekar et al., 2023) required (1) a language model that contains sufficient world knowledge to generate articles worth training on, thereby increasing generation cost; (2) a careful selection of prompts that enable generating high quality, and diverse articles that fill any knowledge gaps in the synthetic corpus. This challenge was highlighted in follow-up work of Li et al. (2023c), and has the potential of inadvertently creeping in biases in the language models (Maini, 2023), as opposed to those trained on the natural diversity of the web. As a remedy to the challenge of (i)

generation cost, and (ii) data diversity, we propose **WRAP** that leverages the natural diversity of articles on the web, allowing us to utilize significantly smaller LLMs (than GPT-3.5) to generate high-quality paraphrases of noisy and unstructured articles on the web.

3.1 Rephrasing the Web

It has been observed in past work that up-weighting high-quality data, such as texts from Wikipedia, can be useful to improve language modeling. These terms have generally been very loosely defined and there is only anecdotal evidence of the same (Brown et al., 2020; Wenzek et al., 2019). At the same time, web data is deficient of text in question-answering or conversational format, which is a prominent use case of language models. Based on these two insights, we design our rephrasing styles.

Rephrasing Styles In lieu of the anecdotal evidence above, we attempt rephrasing documents on the web in four different styles—(i) Easy (text that even a toddler will understand); (ii) Medium (in high quality English such as that found on Wikipedia); (iii) Hard (in terse and abstruse language); (iv) Q/A (in conversation question-answering format). In order to operationalize rephrasing in these stylistic variations, we appropriately prompt an instruction-tuned model. The rephrased examples of these four styles and the prompts templates used in our work are provided in Appendix I.

Generating Synthetic Data Now, we detail how we utilize an instruction-tuned language model to rephrase texts from web-crawled datasets such as C4 (Raffel et al., 2020) (which we use for all our experiments). In particular, we use a frozen Mistral-7B instruction-tuned model (Jiang et al., 2023) (see Ablations in Section 6 for other models). To generate synthetic data in "medium" style, the Mistral model is prompted using the following instruction: *"For the following paragraph give me a paraphrase of the same in high-quality English language as in sentences on Wikipedia"*. The prompt was created using iterative human feedback by comparing outputs of 'medium' sized LLMs with those of GPT-4. We use the model output to create a parallel corpus of "high-quality" synthetic data corresponding to the original noisy web data. Each example has a maximum of 300 tokens, which was decided based on our empirical observation that asking an LLM to rephrase more than 300 tokens, often led to loss of information. Discussions on data quality can be found in Section C.

Combining Real and Synthetic Data Our method of re-phrasing web data naturally incorporates the information diversity found on the internet. However, it does not incorporate the noise in real data. While synthetic data may help LLMs pre-train faster, we also want them to be able to understand noisy web text that may be filled with typos and linguistic errors so that the LLMs do not fail in user facing situations. In order to incorporate this style diversity in language modeling, we sample real and synthetic data in a 1:1 ratio.

3.2 Implementation Details

Architecture We train decoder-only transformer models (Vaswani et al., 2017) at three different scales, small, medium and XL. The small-scale (128M parameter) model consists of 12 layers, 12 attention heads, and a hidden dimension size of 768. The medium-scale (350M parameter) model consists of 24 layers, 16 attention heads, and a hidden dimension size of 1024. The XL-scale (1.3B parameter) model consists of 24 layers, 16 attention heads, and a hidden dimension size of 2048. We do not use dropout in either model and a maximum sequence length of 1024. The models are trained using NVIDIA's Megatron-LM repository.

Pre-training We train all our XL models for a total of 300k steps with a batch size of one million tokens, unless otherwise specified. We use a maximum learning rate of $3e^{-4}$ for the 128M, and 350M parameter models, and $2e^{-4}$ for the 1.3B parameter model. The minimum learning rate is $1e^{-5}$. We use a weight decay of 0.01, along with a gradient clipping norm of 1.0. We use cosine learning rate scheduler with a warmup for 1% of total steps; and Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4 Perplexity Evaluation

We evaluate the perplexity of the pre-trained model on the validation set of multiple out-of-distribution datasets. All models are either trained on the C4 dataset (Raffel et al., 2020), or a particular stylistic rephrase of the same. All the evaluations are done on 21 sub-domains of the Pile (Gao et al., 2020). These subsets are created from the first 10,000 documents from each domain of the Pile dataset. We

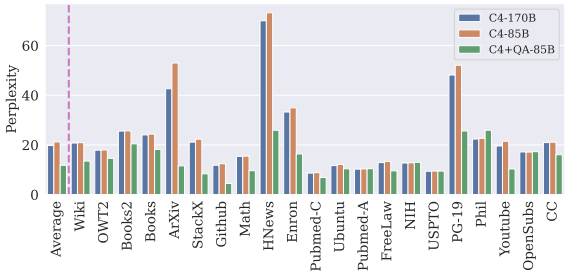


Figure 2: **WRAP (C4 + QA-85B) v/s C4**: Comparison of perplexity on the Pile for a 1.3B LLM trained for 300B tokens shows that WRAP outperforms models trained on 2x real data.

then evaluate the perplexity of the model on these subsets. Additional evaluation details are provided in Appendix D. It is important to note that we evaluate perplexities on the Pile instead of C4. Training on multiple distributions of text (synthetic and real web) does comes at a marginal cost of less than 1 perplexity on the C4 validation set. To understand our choice of evaluation, and why we observe this perplexity increase, we note that training over the C4 corpus corresponds to minimizing the objective

$$\theta_{c4} = \min_{\theta} \mathbb{E}_{x \sim D_{c4}} \left[\mathcal{L}(\theta; x) \right], \quad (1)$$

that attempts to exactly model the C4 web text. In contrast, training over multiple styles corresponds to minimizing the risk over a different distribution,

$$\theta_{\textbf{WRAP}} = \min_{\theta} \mathbb{E}_{x \sim D_{c4} \cup D_{syn}} \left[\mathcal{L}(\theta; x) \right]. \quad (2)$$

Solving for equation 2 does not minimize the risk over C4-only, and hence the small drop in performance between θ_{c4} and $\theta_{\textbf{WRAP}}$ on the C4. For meaningfully comparing models trained on the C4 and on its synthetic rephrases, we evaluate their generalization capability on 21 different domains of the Pile (Gao et al., 2020). Results for each domain are presented in Figure 2.

Data Complexity In Figure 1c, we show that models trained for fewer tokens (150B) and even smaller 350M models outperform training on the full C4 for 300B tokens indicating faster learning with synthetic rephrases. On some domains such as ArXiv and HackerNews, we observe that training with synthetic data allows reducing the perplexity by nearly 3x of the perplexity of models trained on real data alone. This suggests that in many cases it may not be possible to offset the performance advantage of pre-training on synthetic data by merely training on more real data. On an average of 21 subsets of the Pile, our models improve perplexity by 50% over models trained on real data alone.

Learning Speed We observe that even at the first checkpoint (10B tokens) of **WRAP** training, the average perplexity of the LLM on the Pile is lower than that achieved by pre-training on C4 for 15 checkpoints. This suggests a 15x pre-training speed-up. We defer the discussion on learning speed to 'zero-shot' tasks in order to make more meaningful comparisons.

5 Zero-shot Tasks

We now evaluate our pre-trained language models on various zero-shot question answering (QA) benchmarks using the LLM Evaluation Harness[1] (Gao et al., 2023).

5.1 Datasets

We evaluate our models on a total of 13 different zero-shot benchmarks to assess their abilities across various natural language tasks like common sense reasoning, language and knowledge understanding and mathematical reasoning.

General Understanding The General Understanding category comprises datasets testing broader cognitive skills and language comprehension. **ARC Easy (ARC-E)** (Clark et al., 2018) is the less challenging counterpart of ARC-C, featuring questions that require basic reasoning skills. **BoolQ** (Clark et al., 2019) includes boolean questions that focus on reading comprehension and general language understanding. **Winogrande (Wino.)** (Sakaguchi et al., 2021) challenges models with common sense reasoning in language, particularly in pronoun disambiguation. **PIQA** (Bisk et al., 2020) assesses understanding of physical processes, an essential part of practical common sense. **HellaSwag** (Zellers et al., 2019) tests the ability to complete scenarios coherently, demanding both language understanding and common sense. **TruthfulQA** (Lin et al., 2021) is centered on generating truthful, accurate answers, thus testing the model's factual correctness. **OpenBookQA (OBQA)** (Mihaylov et al., 2018) evaluates the understanding of a broad range of facts and concepts. Finally, **LogiQA-2** (Liu et al., 2023b) assesses the model's capacity to comprehend logical principles.

[1]We use git commit - `89618bf8` for consistency across all experiments with a batch size of 32.

Specialized Knowledge In the Specialized Knowledge category, we include datasets that demand expertise in specific domains. The **ARC Challenge (ARC-C)** (Clark et al., 2018) contains challenging science exam questions from grades 3 to 9, demanding advanced knowledge. **SciQ** (Johannes Welbl, 2017) provides science exam questions to test model understanding and reasoning in the scientific domain. **PubMedQA** (Jin et al., 2019) focuses on comprehension in biomedical literature. **MathQA** (Amini et al., 2019) tests mathematical problem-solving, requiring both numerical comprehension and reasoning. Lastly, **MMLU** (Hendrycks et al., 2021) spans multiple domains, from professional to academic, testing the model on specialized subjects.

5.2 Results

We compare the performance of a model trained on a mixture of real and synthetic data with models trained on various splits of real data. In all our experiments, we use the C4 (Raffel et al., 2020) dataset for rephrasing and producing splits of synthetic data. We use the abbreviation 'Real Tok.' to denote the number of tokens of web data available for pre-training. In the 'Synthetic + Real' experiments, we augment the same number of synthetic rephrases. We choose 'Real Tokens' as the metric of comparison because we can potentially rephrase the same document multiple times, implying that the total corpus size is not meaningful, and corpus 'knowledge' is the actual currency of interest.

Baselines Methods We pre-train LLMs of (i) Half of C4, and the (ii) Full C4 corresponding to approximately 85 Billion and 170 Billion real tokens respectively (Raffel et al., 2020). We also pre-train our own models on (iii) 160 Billion and (iv) 320 Billion tokens of the RefinedWeb Dataset (Penedo et al., 2023). Additionally, we also compare with the (iv) Pythia-1.4B model that was trained on the Pile (Gao et al., 2020). This dataset is no longer publicly available, hence we utilize a pre-trained model. Finally, we also compare with the recent (v) TinyLlama model (Zhang et al., 2024) that was trained for 3 epochs on 1 Trillion tokens of data from SlimPajama (Shen et al., 2023) and StarCoder (Li et al., 2023a).

General Improvements Across all tasks in Table 1, we observe that models trained on synthetic data combined with the C4 dataset (Synthetic+C4) exhibit an overall higher average performance of 52.3% as compared to those trained solely on the real C4 dataset with a 85B token split, which scored an average of 50.2%. This shows that the inclusion of synthetic data can enhance the general understanding capabilities of NLP models. Moreover, even the TinyLLama model trained for 10x compute and data, performs comparably to models trained on real data only. This suggests that the gains from filtering out, or adding more real data are low. As opposed to this, **WRAP** shows that pre-training on even small amounts of synthetic data can contribute to large performance gains.

Specialized Knowledge Tasks The key message from the results in Table 1 is that synthetic data can not impart 'new knowledge'. It can only help pre-train faster, which was also the premise of our work. In particular, we note several key findings:

1. Pre-training on larger datasets helps improve performance, by presumably exposing the LLM to more "knowledge". For instance, the Pythia (300B) model achieves an average score of 43.4%, outperforming the smaller C4 (85B) dataset's score of 42.1%.

2. Despite the advantages of a larger dataset, the improvements saturate. For example, RefinedWeb (320B) model outperforms the RefinedWeb (160B) model by only 0.6%. Similary, the TinyLlama model (1T tokens) performs comparably to the **WRAP** model, which only had 85B tokens of raw web data.

Specific Improvements We see maximum improvement in the TruthfulQA dataset, with the Synthetic (85B) model scoring 44.0%, which is significantly higher than any other model's performance on this dataset. This is potentially because instruction-tuned LLMs already correct potential misconceptions while rephrasing the text. Interestingly, we notice how adding real data to the synthetic model (Synthetic+C4) reduces the performance on TruthfulQA by 4%, down to 40.5%, indicating a potential dilution of the benefits gained from synthetic data when combined with real data. Other datasets such as HellaSwag, and BoolQ, for which C4 trained models do well, continue to show the benefits of incorporating combinations of C4 and synthetic rephrases.

6 Analysis and Ablations

We further ask the following Research Questions (RQs) to investigate in a finer granularity how to

Dataset (Real Tok.)	ARC-E	BoolQ	Wino.	PIQA	HellaSwag	TruthfulQA	OBQA	LogiQA	Avg
Half C4 (85B)	53.2	59.1	57.3	76.0	61.0	34.1	34.2	26.6	50.2
Full C4 (170B)	54.6	54.2	59.0	76.1	61.2	33.5	36.8	26.9	50.3
RW (160B)	58.0	60.7	57.5	74.9	59.4	36.8	35.2	26.8	51.2
RW (320B)	56.7	61.1	57.1	75.2	59.2	36.0	35.4	27.2	51.0
Pythia-Pile (300B)	53.9	63.3	57.5	71.0	52.0	38.9	33.2	27.3	49.6
TinyLlama (1T)	55.3	57.8	59.1	73.3	59.2	37.6	36.0	27.0	50.7
Synthetic (85B)	57.7	60.0	58.8	76.9	57.8	44.0	34.2	26.3	52.0
Synthetic+C4 (85B)	57.4	62.2	58.9	76.0	60.8	40.6	35.3	27.1	52.3

Dataset (Real Tok.)	ARC-C	SciQ	PubMedQA	MathQA	MMLU	Avg
Half C4 (85B)	29.5	76.8	57.2	22.9	24.2	42.1
Full C4 (170B)	29.7	77.3	57.4	23.8	23.9	42.4
RW (160B)	29.3	81.4	56.2	23.4	25.9	43.2
RW (320B)	30.0	83.1	57.4	23.0	25.4	43.8
Pythia-Pile (300B)	28.6	79.2	60.6	24.3	24.3	43.4
TinyLlama (1T)	30.1	81.8	61.4	23.7	25.8	44.6
Synthetic (85B)	32.3	78.4	60.2	23.2	24.6	43.7
Synthetic+C4 (85B)	31.5	79.0	61.5	23.5	24.8	44.0

Table 1: **(Left)** Evaluation of 1.3B parameter LLMs on "General Understanding Tasks" focusing on general reasoning, language understanding, and common sense. **(Right)** Evaluation on 'Specialized Knowledge Tasks' that require specific domain knowledge such as science and mathematics. Results for **WRAP** are averaged over 3 runs.

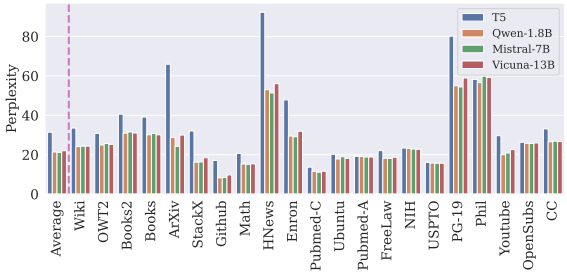


Figure 3: **Importance of High Quality Paraphraser:** Perplexity across all the Pile domains for **WRAP** on data generated by different LLMs. Results show that even small models like Qwen-1.8B can generate paraphrases of high quality. Though, a low quality rephraser like our fine-tuned T5-base model leads to significantly worse language modeling.

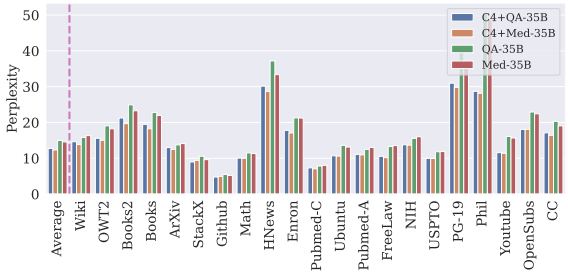


Figure 4: **Importance of Real Data:** Comparing perplexity on the Pile when pre-training on C4 with synthetic data vs. synthetic data only. Models are 1.3B parameters trained for a total of 150B tokens on a real data subset containing 35 Billion tokens of the C4.

enhance performance optimally. We discuss additional data and method ablations in Appendix G.

RQ1: How important is to have a high-quality re-phraser? We use data from four distinct rephrasing models (T5-base (Raffel et al., 2020), Qwen-1.8B-chat (Bai et al., 2023a), Mistral-7B-chat (Jiang et al., 2023), and Vicuna-13B-chat-v1.3 (Chiang et al., 2023)) and train a 345M model for 30B tokens. We generate data from all models using the same prompt. In case of the T5-base model, we finetune the model for 1 epoch on rephrase pairs from the Vicuna-13b-chat model. We find that pre-training on data generated by smaller re-phrase models like Qwen-1.8B and Mistral-7B achieve lower perplexity than Vicuna 13B (Figure 3). At the same time, our fine-tuned T5-base model performs significantly worse than the rest. Even then, all rephrase models reduce perplexity over only real C4 data. It remains an open question to test the limits of how small can we train a paraphrase model to generate high quality synthetic data to further scale the applicability of **WRAP**.

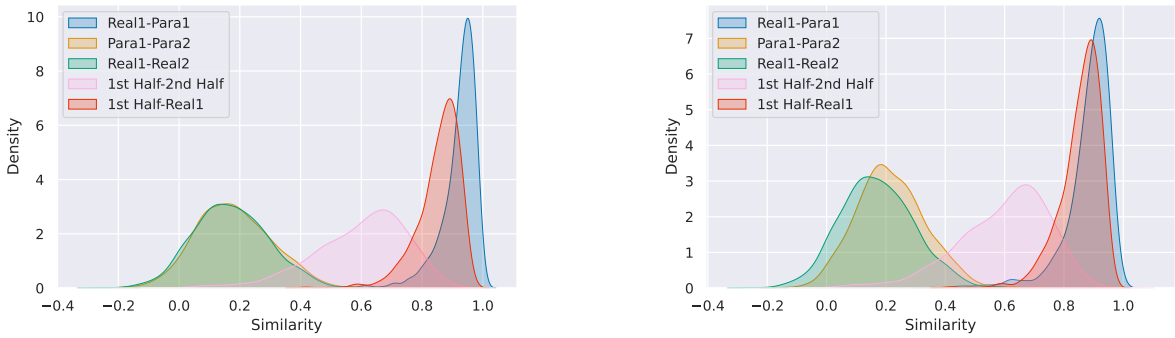
RQ2: How important is it to have real C4 data? Our findings in Tables 1 indicate that synthetic data using the QA prompt are sufficient for strong performance on QA tasks. However, when evaluated on Pile perplexity, we observe significant degradation in perplexity across many sub-domains in Figure 4. This is likely because synthetic data is very clean containing few special characters and being highly structured. In contrast several subdomains of the Pile such as OWT, and Hackernews have such special tokens. On domains such as Philpapers and Gutenberg, we observe that dropping real C4 text from the pre-training data, and training on synthetic documents alone drops performance significantly (increase in perplexity). This is once again attributed to the fact that synthetic data does not contain certain 'tags' and 'styles' that are prevalent in real data scrapes, and emphasized how **WRAP** is a better strategy than pre-training on synthetic data alone. In terms of performance on zero-shot tasks, the presence of real data helps improve zero-shot performance in Tables 2. Since these tasks contain well-written Q/A pairs, the effect is not as prominent as on Pile.

RQ3: Is there data leakage from the rephrase model to the trained model? We investigate whether our synthetic data maintains similar semantic meaning while being stylistically different

Dataset (Real Tok.)	ARC-E	BoolQ	Wino.	PIQA	HellaSwag	TruthfulQA	OBQA	LogiQA	Avg
Med+C4-35B	53.2	57.0	55.7	74.8	57.6	36.5	33.2	26.2	49.3
QA+C4-35B	55.1	63.3	55.7	75.6	57.9	41.4	34.0	26.1	51.1
Med-35B	49.6	59.5	53.4	73.4	52.7	36.3	32.2	26.0	47.9
QA-35B	55.9	62.0	53.9	76.9	54.7	43.0	34.0	27.7	51.0

Dataset (Real Tok.)	ARC-C	SciQ	PubMedQA	MathQA	MMLU	Avg
Med+C4-35B	29.6	74.3	46.2	22.9	25.2	39.6
QA+C4-35B	30.3	76.8	62.2	23.0	26.1	43.7
Med-35B	28.8	73.8	59.4	22.6	24.7	41.9
QA-35B	31.1	77.4	59.2	22.4	25.0	43.0

Table 2: **Importance of Real Data:** (Left) Evaluation of $\sim$ 1.3B parameter LLMs trained for 150B tokens on General Understanding Tasks. (Right) Evaluation on Specialized Knowledge tasks. Results show that adding real data helps improve model performance when pre-training on 'Medium' or 'Wikipedia-style' paraphrases.



(a) Cosine similarity Medium synthetic data

(b) Cosine similarity QA synthetic data

Figure 5: Comparison between synthetic and real data from the C4 corpus showing that synthetic data maintains semantic meaning compared with the real C4 data and primarily changes style for (a) medium rephrases of C4, and (b) QA rephrases of C4.

from the original C4 data and matching the style of different PILE domains. We start by comparing pairs of examples of synthetic and real data to confirm the performance gain is not attributed to knowledge leakage from the rephrase models. We subset the first 1000 samples from each dataset.

We show the cosine similarity of the sentence embeddings from a pre-trained BERT model trained with SimCSE objective (Gao et al., 2021) for medium and qa prompts in Figure 5(a) and (b). When computing similarity, we remove outliers. Figures with distributions use a gaussian Kernel Density Estimator (KDE) to construct distributions for statistics from 1000 values. The cosine similarity of real-synthetic pairs is higher than several baselines including two random real samples from C4, a continuation baseline which computes cosine between the first half of a sample and the full sample, and cosine similarity between the first and second half of the same sample. High similarity indicates that re-phrases maintain similar meaning to their real counterparts without adding information.

7 Conclusion

Strong language models are being pre-trained on combinations of real and synthetic data. Using synthetic data enables baking in desirable attributes such as fairness, bias, and style (like instruction following) directly into the data, eliminating the need to adjust the training algorithm specifically. This offers an alternative approach to aligning language models to human values. The recent uptick in interest around synthetic data, especially for instruction-tuning language models, is noteworthy, with concurrent researchers also leveraging it for pre-training. As we transition into this paradigm, understanding the properties of the data fed to our models is paramount. This paper aims to be a comprehensive guide on employing different synthetic style data in LLM pre-training. We delve into its significance from two vantage points: (1) In scenarios with scarce high-quality data, synthetic rephrases offer more value than mere repetition of existing data; (2) Synthetic data can be a boon for generalization on different text domains, and for generating text in styles that are underrepresented in the pre-training dataset. As practitioners generate synthetic data for training models, they will be faced with important and expensive design choices—(i) How important is the quality of the synthetic data generator?; (ii) How to balance real and synthetic data? (iii) When does training on synthetic data reach a point of diminishing returns? We take a first step towards answering these questions.

8 Limitations

8.1 Cost Analysis

Should you generate synthetic data, or just train longer on real data?

The applications of **WRAP** lies in both paradigms—(i) low-resourced data settings such as a language model for Finnish language (Luukkonen et al., 2023), and (ii) data-rich settings such as training on the common crawl. In the former, there is no alternative option of naively gathering more data, and hence, synthetic data is a natural solution that should outperform training on in-domain data alone. However, there is a significant interest in training language models on English, or more broadly, general web data. Is using synthetic data a viable option even in this paradigm?

Before, we dive into the feasibility of pre-training on synthetic data, we should acknowledge the results of Table 1. The TinyLlama model trained for 3 Trillion tokens also underperforms a model jointly trained on real and synthetic data. In fact, it performs quite comparably to the models that were trained for 300B tokens on just real data as well. This suggests that the ceiling for improvement by training for longer may not be that high (for a model of size 350M/1.3B parameters; larger models may benefit from training for longer).

To analyze this cost trade-off, we compare the cost of generating synthetic data, versus that of training a language model on extra data. For our synthetic data generation experiments, we use the vLLM (Kwon et al., 2023) library for fast generation. In particular, we are able to generate 3M tokens per hour on a single A100 when using the Mistral-7B. Generating 85B tokens (as in our work) accounts for about 25K GPU hours.

In comparison, on 64 A100s, we achieve a throughput of 0.5M tokens per second. Assuming training for 300B tokens, would mean 256 GPU days, accounting for about 6k GPU hours to train a single model. On the contrary, training a 13B model would take about 30K GPU hours. At the scale of training a 13B model, reducing the training cost by 3-10x can incorporate the cost overhead of training with synthetic data in a single run.

While the cost of generating high quality data is still relatively high, two important sources of improvement impact this cost analysis. First, if we use the Qwen-1.8B model (Bai et al., 2023b) for rephrasing, we are able to get a 3x higher token throughput. As seen in our preliminary results in Fig 3, the model pre-trained on rephrases generated by Qwen model performs comparably to that by the Mistral model. This reduces the cost of generation by 3x. More recent work in speculative decoding (Liu et al., 2023c) and optimized inference (Xia et al., 2024) suggest that we can leverage another 3-5x improvement in the generation cost. Hence, indeed, even at the scale of just 1.3B parameter model training, we can already improve upon the cost of pre-training using just real data.

Two additional important advantages of synthetic data generation that could not be accounted for in the discussion above:

1. The cost of synthetic data generation is a one-time investment, and we may train many models of varying scales once the data is generated.

2. Data generation is 100% parallelizable, whereas training requires the availability of a big cluster with fast inter-node connectivity. This is much more expensive. On the other hand, generation can be thought of as a side process that can fill in the empty GPUs in any large-scale compute cluster, and runs on single GPU machines.

8.2 Diversity of Synthetic Generations

Another limitation is enforcing the diversity in the generated data. This diversity comes from both the "style" and the "knowledge" contained in the generated data. Recent works (Li et al., 2023b,c) used a selection of topics, or scenarios to seed the model to generate novel texts. Still, a recent study by Padmakumar et al. (2023) showed that using language models for AI-assisted writing tends to reduce content diversity, particularly when using instruction-tuned models. While we used the paradigm of rephrasing specifically to mitigate the issues pertaining to the diversity of novel content generation, it remains for future work to assess the presence (or lack of) and impact of content diversity in paraphrase models.

References

Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *ArXiv*, abs/2303.09540.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei,

Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. 2023. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *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)*, pages 2357–2367.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. 2023. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023b. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Hritik Bansal and Aditya Grover. 2023. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Dami Choi, Alexandre Passos, Christopher J Shallue, and George E Dahl. 2019. Faster neural network training with data echoing. *arXiv preprint arXiv:1907.05550*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *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)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl,

Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. Nl-augmenter: A framework for task-sensitive natural language augmentation.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

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

Stanislav Fort, Andrew Brock, Razvan Pascanu, Soham De, and Samuel L Smith. 2021. Drawing multiple augmentation samples per image during training efficiently decreases test error. *arXiv preprint arXiv:2105.13343*.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. 2023. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

Edward Gibson et al. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Suchin Gururangan, Dallas Card, Sarah Dreier, Emily Gade, Leroy Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A Smith. 2022. Whose language counts as high quality? measuring language ideologies in text data selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2562–2580.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. 2020. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2023. Impossible distillation: from low-quality model to high-quality dataset & model

for summarization and paraphrasing. *arXiv preprint arXiv:2305.16635*.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Optimizing instruction tuning for long text generation with corpus extraction. *arXiv preprint arXiv:2304.08460*.

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

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023a. Starcoder: may the source be with you!

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

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. 2023a. Tinygsm: achieving> 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023b. Logiqa 2.0 — an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–16.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Ion Stoica, Zhijie Deng, Alvin Cheung, and Hao Zhang. 2023c. Online speculative decoding. *arXiv preprint arXiv:2310.07177*.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, et al. 2023. Fingpt: Large generative models for a small language. *arXiv preprint arXiv:2311.05640*.

Pratyush Maini. 2023. Phi-1.5 model: A case of comparing apples to oranges?

Pratyush Maini, Sachin Goyal, Zachary C Lipton, J Zico Kolter, and Aditi Raghunathan. 2023. T-mars: Improving visual representations by circumventing text feature learning. *arXiv preprint arXiv:2307.03132*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Masanori Oya. 2021. Three types of average dependency distances of sentences in a multilingual parallel corpus. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 652–661.

Vishakh Padmakumar, Behnam Hedayatnia, Di Jin, Patrick L Lange, Seokhwan Kim, Nanyun Peng, Yang Liu, and Dilek Hakkani-Tur. 2023. Investigating the representation of open domain dialogue context for transformer models. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 538–547.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. 2023. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. Model dementia: Generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Irene Solaiman and Christy Dennison. 2021. Process for adapting language models to society (palms) with values-targeted datasets. *Advances in Neural Information Processing Systems*, 34:5861–5873.

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

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. 2023. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Source code is all you need. *arXiv preprint arXiv:2312.02120*.

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

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *arXiv preprint arXiv:2305.10429*.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis. *arXiv preprint arXiv:2305.13230*.

Haichao Yu, Yu Tian, Sateesh Kumar, Linjie Yang, and Heng Wang. 2023. The devil is in the details: A deep dive into the rabbit hole of data filtering. *arXiv preprint arXiv:2309.15954*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A Dataset Details

A.1 Training Dataset

The primary pretraining corpus in our experiments is Colossal Clean Crawled Corpus (C4), a curated English text dataset comprising over 170 billion tokens. This corpus is derived from CommonCrawl, a common practice in the pretraining of LLMs (Brown et al., 2020; Raffel et al., 2020; Touvron et al., 2023). This data source is also prominently featured in openly available LLM pretraining corpora, including The Pile (Gao et al., 2020) and RedPajama (Computer, 2023). There are different versions of CommonCrawl data and our selection of C4 for pretraining is driven by driven by its size and quality.

We also compare with pre-training on the Refined Web corpus (Penedo et al., 2023). The dataset is also derived from the CommonCrawl, however has a more stringent filtering process. Our selection of Refined Web is for comparing synthetic rephrases to high quality subsets of web data, which were shown to achieve similar performance compared with curated datasets (Penedo et al., 2023). For our experiments we used the first 3050 files and train for 300B tokens to match training on C4. We aso conduct experiments with the first 1650 files to account for multiple epochs on the Refined Web dataset.

A.2 Pile Perplexity Evaluation

For the evaluation phase, we employed 20 subsets from the Pile corpus. We excluded the Europarl subset because it contained non-English language. The subsets used are: CC, StackExchange, Wikipedia, GitHub, PubMed Abstracts, Openwebtext2, Freelaw, Math, NIH, USPTO, Hackernews, Enron, Books3, PubMed Central, Gutenberg, Arxiv, Bookcorpus2, Opensubtitles, Youtubesubtitles, Ubuntu, and Philpapers. We take the first 10000 samples from each subset and split into documents of maximum length 1024. The reported average in all perplexity plots is a weighted average over the perplexity of all domains according to the ratios in Table 3.

A.2.1 Pile Weighted Average Ratios

We report the ratios for samples according to the first 10,000 documents from our Pile validation set in Table 3. Note that there are some slight variations in the ratios compared with those reported in (Gao et al., 2020), but most ratios are similar.

Dataset	Our Ratio (%)	Ratio (%)
ArXiv	10.4	9.0
BookCorpus2	0.8	0.8
Books3	11.8	12.1
Pile-CC	14.0	18.11
Enron	0.1	0.1
EuroParl	1.1	0.7
FreeLaw	5.3	6.1
Github	10.9	7.6
Gutenberg	1.5	2.2
Hackernews	0.6	0.6
Dm Mathematics	2.0	1.2
NIH	0.2	0.3
OpenSubtitles	1.3	1.6
OpenWebText2	8.2	10.0
PhilPapers	0.7	0.4
PubMed Abstracts	0.7	3.1
PubMed Central	14.9	14.4
StackExchange	5.8	5.1
Ubuntu	1.3	0.9
USPTO	2.7	3.7
Wikipedia	3.4	1.5
YoutubeSubtitles	0.6	0.6

Table 3: Pile ratios for our evaluation compared with published ratios.

A.3 Zero-shot Evaluation Dataset

We evaluate our models on a total of 13 different zero-shot benchmarks to assess their abilities across various natural language tasks. These benchmarks are categorized into two subsets: Specialized Knowledge and General Understanding.

Specialized Knowledge This subset comprises datasets that focus on domain-specific knowledge and expertise.

- **ARC Challenge (ARC-C)**: This dataset is part of the AI2 Reasoning Challenge (ARC) (Clark et al., 2018), containing science exam questions from grades 3 to 9. The ARC Challenge set includes more difficult questions that necessitate higher-order reasoning.
- **SciQ**: A dataset of science exam questions, specifically designed to evaluate the ability of NLP models in understanding and reasoning within the scientific domain (Johannes Welbl, 2017).
- **PubMedQA**: This dataset focuses on biomedical literature and is designed to evaluate the understanding of medical and healthcare-related information (Jin et al., 2019).
- **MathQA**: This dataset challenges models in mathematical problem-solving, requiring both numerical understanding and reasoning skills (Amini et al., 2019).
- **MMLU**: Multi-domain question answering, MMLU assesses the model's expertise over a wide range of specialized subjects, from professional domains to academia (Hendrycks et al., 2021).

General Understanding This subset contains datasets that test general cognitive skills, language understanding, and common sense reasoning.

- **ARC Easy (ARC-E)**: The Easy set of the AI2 Reasoning Challenge (Clark et al., 2018) features questions from the same source as ARC-C but are considered less challenging and do not require as advanced reasoning skills.
- **BoolQ**: A dataset consisting of boolean (yes/no) questions, focusing on reading comprehension and general understanding of natural language text (Clark et al., 2019).
- **Winogrande (Wino.)**: This dataset challenges models on common sense reasoning in a language context, focusing on pronoun disambiguation tasks (Sakaguchi et al., 2021).
- **PIQA**: Physical Interaction Question Answering tests the understanding of everyday physical processes, an aspect of practical common sense (Bisk et al., 2020).
- **HellaSwag**: This dataset evaluates a model's ability to complete scenarios in a contextually and logically coherent manner, requiring both language understanding and common sense reasoning (Zellers et al., 2019).
- **TruthfulQA**: Focused on the generation of truthful, accurate answers, this dataset challenges models on their ability to discern and reproduce factually correct information (Lin et al., 2021).
- **OpenBookQA (OBQA)**: OpenBookQA requires understanding a wide array of facts and concepts, thereby evaluating the model's broader knowledge and reasoning skills (Mihaylov et al., 2018).
- **LogiQA-2**: This dataset involves logical reasoning, testing the model's capability to understand and apply logical constructs and principles (Liu et al., 2023b).

Each dataset in these subsets is carefully selected to challenge and evaluate specific aspects of natural language processing models, ranging from domain-specific knowledge in science, medicine, and mathematics, to broader skills like common sense reasoning and general language understanding.

B Filtering Details for Synthetic Data

When generating synthetic paraphrases using language models, we occasionally encounter the challenge of extraneous introductions in the generated outputs. Such paraphrases might commence with phrases like "Here's a paraphrase...", "The following..." or even contain keywords such as "high-quality English". To mitigate this, we've developed a method to filter and refine the synthetic outputs.

B.1 Methodology

The primary function, `remove_unwanted_part`, starts by splitting the input data into individual sentences. If the first sentence contains delimiters

such as "\n\n" (indicating a new paragraph) or ":", the function checks the segment preceding the delimiter for the aforementioned unwanted elements. If these elements are detected, the preceding segment is removed. The entire revised content is then reconstructed and returned. In cases where no modifications are applicable, but we still have the flagged keywords, we remove the paraphrase completely. To achieve this:

1. Split the input data into individual sentences using the NLTK's sentence splitter function.
2. Examine the first sentence for the presence of delimiters.
3. If a delimiter is detected, check the preceding segment for unwanted elements.
4. If unwanted elements are found, discard the preceding segment (before an occurrence of `"\n\n"` or `":"`).
5. Modify and return the filtered paragraph.

Based on manual inspection we found that the error rate (occurrence of sentences with unwanted elements) after the modification is less than 0.1%.

C Properties of Synthetic Corpus

To understand the properties of synthetic data generated from the rephrase model that lead to better pre-training performance, we compare the semantic similarity, syntactic complexity, and diversity between synthetic data, C4 data, and data from the Pile. Our primary focus is answering the following questions about synthetic data: (i) Do models trained on synthetic data perform better due to information leakage from the rephrase model? (ii) Does the rephrase model accurately capture multiple styles? (iii) What attributes of synthetic data make it high quality? Our investigation helps address what data is beneficial for better generalization to specific domains, and quantify the importance of data variability and quality.

C.1 Experimental Setup

We take a subset of the first 1000 documents from each of the datasets. For synthetic comparisons with real C4 data, we take pairs of samples, while for Pile subsets, we take the first 1000 samples from the test subset. When computing dataset quality statistics, we remove outliers more than two standard deviations in metric value. When the number of samples from the Pile subset was fewer than 1000, we split samples. Figures with distributions use a Gaussian Kernel Density Estimator (KDE) to construct distributions for statistics from 1000 values.

C.2 Semantic Properties

In Section 6, we compared pairs of examples of synthetic and real data to confirm the performance gain is not attributed to knowledge leakage from the rephrase models using a pre-trained BERT model trained with SimCSE objective (Gao et al., 2021) for medium and qa prompts in Figure 5(a) and (b). We additionally compare the similarity of synthetic rephrases and actual rephrases using the MRPC corpus in Figure 6(c). We denote this additional comparison by RealP (real paraphrase), while maintaining comparison of splits of the sentence: R1 and R2. Synthetic rephrases have similar cosine similarity on average and lower spread compared with true rephrases according in the MRPC corpus.

As the semantic information is similar between C4 and our synthetic data, we further investigate stylistic differences in the data. Figure 7(a) shows the Flesch–Kincaid reading levels for different rephrase styles, and the Pile. Our findings indicate that C4 is on the low end of reading level (7-8). In contrast, medium increases the reading level to 10, and qa synthetic variants further reduces the reading level to 6. Medium synthetic data matches the reading level of Wikipedia, and other high reading level datasets yielding better performance on these domains. On QA synthetic data, we observe reduced reading level. This is because we observed that sentences are typically split into question and answer leading to shorter setnences compared with in the original text and medium style rephrases. This leads to lower metric values for many of the metrics. For type token ratio, we note that the diversity is quite similar between medium and most subsets of the Pile. The QA dataset has particularly low TTR matching ubuntu, github, and math as these are more similar to QA format datasets and have heavy repetition of the Question, and Answer format.

C.3 Syntactic Properties

Finally, we compare the mean tree depth (measured by the mean over setences of the depth of the dependency tree), and mean dependency distance (measured as the average dependency distance of

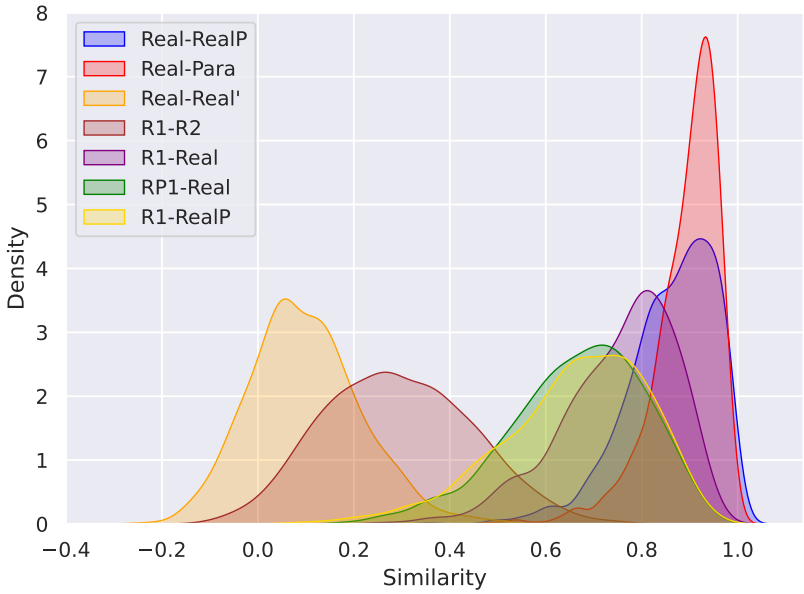
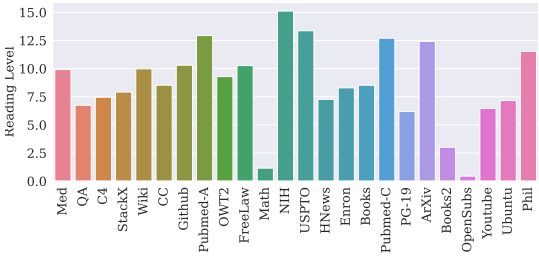
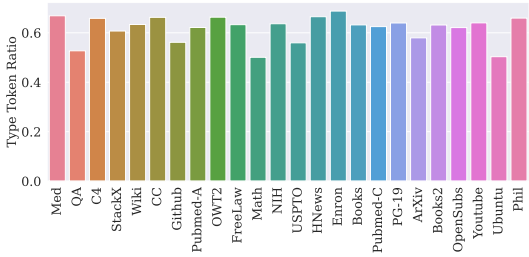


Figure 6: Cosine similarity medium synthetic MRPC rephrases



(a) Flesch-Kincaid Reading Level



(b) Type token ratio

Figure 7: Comparison of readability and diversity (ttr) of synthetic data compared with C4 and different subsets of the Pile.

any pair of words within a sentence) in Figure 8, which have been shown to be good measures of syntactic difficulty (Futrell et al., 2015; Gibson et al., 2000; Oya, 2021). We find similar trends as for reading level and TTR diversity where mediums tyle increase depth, mdd, and syntactic complexity in general. We find again that QA style reduces this complexity.

D Evaluation Metrics

The metric utilized for evaluation is the *macro token level perplexity*. Given a batch of encoded texts, the perplexity at the token level was computed as follows:

Given the accumulated loss over the entire dataset, denoted as L, and the total number of tokens, represented by T, the macro token-level perplexity, denoted as $\mathcal{P}$, is calculated as:

$$\mathcal{P} = \exp\left(\min\left(20, \frac{L}{T}\right)\right) \quad (3)$$

Where:

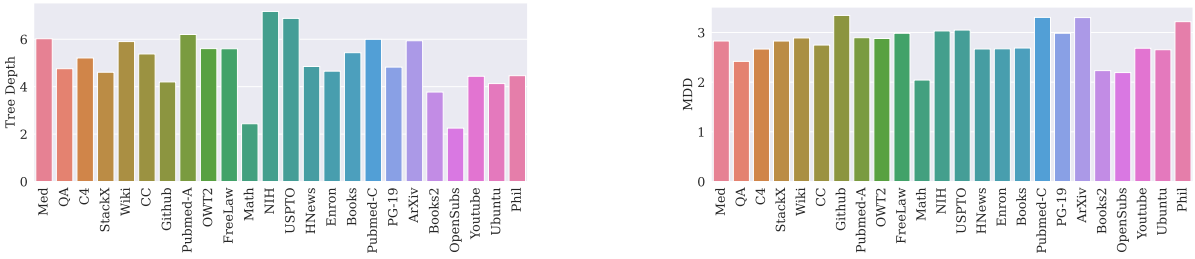
- exp is the exponential function.
- L is the cumulative loss over all shifted logits and labels in the dataset.
- T is the total number of tokens in the dataset.

The value of 20 acts as an upper limit to stabilize the metric in cases of high loss values.

E Limitations

E.1 Cost Analysis

Should you generate synthetic data, or just train longer on real data?



(a) Tree Depth

(b) Mean Dependency Distance

Figure 8: Comparison between synthetic and real data from the C4 corpus showing that synthetic data have higher syntactic complexity indicated by higher average tree depth, and higher mean dependency distance (MDD).

The applications of **WRAP** lies in both paradigms—(i) low-resourced data settings such as a language model for Finnish language (Luukkonen et al., 2023), and (ii) data-rich settings such as training on the common crawl. In the former, there is no alternative option of naively gathering more data, and hence, synthetic data is a natural solution that should outperform training on in-domain data alone. However, there is a significant interest in training language models on English, or more broadly, general web data. Is using synthetic data a viable option even in this paradigm?

Before, we dive into the feasibility of pre-training on synthetic data, we should acknowledge the results of Table 1. The TinyLlama model trained for 3 Trillion tokens also underperforms a model jointly trained on real and synthetic data. In fact, it performs quite comparably to the models that were trained for 300B tokens on just real data as well. This suggests that the ceiling for improvement by training for longer may not be that high (for a model of size 350M/1.3B parameters; larger models may benefit from training for longer).

To analyze this cost trade-off, we compare the cost of generating synthetic data, versus that of training a language model on extra data. For our synthetic data generation experiments, we use the vLLM (Kwon et al., 2023) library for fast generation. In particular, we are able to generate 3M tokens per hour on a single A100 when using the Mistral-7B. Generating 85B tokens (as in our work) accounts for about 25K GPU hours.

In comparison, on 64 A100s, we achieve a throughput of 0.5M tokens per second. Assuming training for 300B tokens, would mean 256 GPU days, accounting for about 6k GPU hours to train a single model. On the contrary, training a 13B model would take about 30K GPU hours. At the scale of training a 13B model, reducing the training cost by 3-10x can incorporate the cost overhead of training with synthetic data in a single run.

While the cost of generating high quality data is still relatively high, two important sources of improvement impact this cost analysis. First, if we use the Qwen-1.8B model (Bai et al., 2023b) for rephrasing, we are able to get a 3x higher token throughput. As seen in our preliminary results in Fig 3, the model pre-trained on rephrases generated by Qwen model performs comparably to that by the Mistral model. This reduces the cost of generation by 3x. More recent work in speculative decoding (Liu et al., 2023c) and optimized inference (Xia et al., 2024) suggest that we can leverage another 3-5x improvement in the generation cost. Hence, indeed, even at the scale of just 1.3B parameter model training, we can already improve upon the cost of pre-training using just real data.

Two additional important advantages of synthetic data generation that could not be accounted for in the discussion above:

1. The cost of synthetic data generation is a one-time investment, and we may train many models of varying scales once the data is generated.

2. Data generation is 100% parallelizable, whereas training requires the availability of a big cluster with fast inter-node connectivity. This is much more expensive. On the other hand, generation can be thought of as a side process that can fill in the empty GPUs in any large-scale compute cluster, and runs on single GPU machines.

E.2 Diversity of Synthetic Generations

Another limitation is enforcing the diversity in the generated data. This diversity comes from both the

"style" and the "knowledge" contained in the generated data. Recent works (Li et al., 2023b,c) used a selection of topics, or scenarios to seed the model to generate novel texts. Still, a recent study by Padmakumar et al. (2023) showed that using language models for AI-assisted writing tends to reduce content diversity, particularly when using instruction-tuned models. While we used the paradigm of rephrasing specifically to mitigate the issues pertaining to the diversity of novel content generation, it remains for future work to assess the presence (or lack of) and impact of content diversity in paraphrase models.

F Additional Results for Smaller Model and Token Sizes

F.1 Results for 350M Models Trained for 75B Tokens

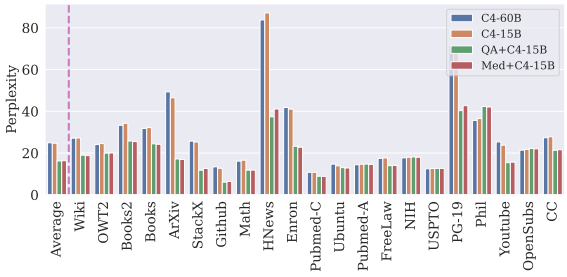


Figure 9: Perplexity across all domains of the Pile comparing combining multiple styles of synthetic data. Models are 350M parameters trained for a total of 75B tokens.

We train models at smaller scales and demonstrate improvement. In particular we train a 350M GPT-2-medium architecture for a total of 75B tokens. We show Pile perplexity averaged across the 21 domains is much lower than for that of the model trained only on C4 in Figure 9, and even lower than 1.3B models trained only on C4 in Figure 1c. We also show an increase of 2% across general understanding language tasks, and roughly $2-3\%$ on specialized knowledge tasks in Table 4 when adding QA rephrases. We also experimented with medium rephrases at this smaller scale. Our findings indicate that the high quality provided by medium rephrases improves performance over only C4, however matching the style as indicated by QA rephrase performance further improves performance.

F.2 Results for 1.3B Models Trained for 150B Tokens

We additionally train 1.3B GPT-2-XL models at 150B tokens, reducing the number of steps by half. We show Pile perplexity averaged across the 20 domains is much lower than for that of the model trained only on C4 in Figure 10, and even lower than 1.3B models trained only on C4 in Figure 1c for twice as long. We also show an increase of 2% across specialized knowledge tasks, and roughly 2% on general understanding tasks in Table 5 when adding QA rephrases. We also experimented with medium rephrases at this smaller scale, and report similar findings consistent with other small-scale experiments.

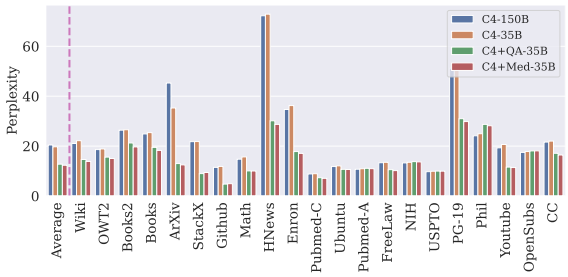


Figure 10: Perplexity across all domains of the Pile comparing combining multiple styles of synthetic data. Models are 350M parameters trained for a total of 75B tokens.

G Ablations

We further ask the following Research Questions (RQs) to investigate in a finer granularity for how to enhance performance with synthetic data.

G.1 Data Combination Analysis

RQ4: Does a combination of multiple synthetic datasets improve performance? We measure the impact of combining multiple synthetic styles with C4 for training. We consider two variants: combining in a 1:1 ratio meaning that there are two copies of C4 to match two synthetic styles (medium and QA), and 1:2 ratio which combines only one instance of the C4 dataset. For zero-shot QA tasks, our finding in Table 6 indicate lower performance than combining only QA and C4 data. Evaluations over the Pile are shown in Figure 11. We notice that both the 'Q/A' and 'Wikipedia' paraphrases help improve performance on certain domains. For example, 'Stackexchange', that has lots of question-answers benefits from the presence of synthetic

Dataset (Real Tok.)	ARC-E	BoolQ	Wino.	PIQA	HellaSwag	TruthfulQA	OBQA	LogiQA	Avg
C4-18B	43.7	52.8	53.0	70.1	43.9	37.8	29.6	26.5	44.7
C4-75B	43.9	53.4	51.6	71.2	44.2	39.0	29.6	25.7	44.8
QA+C4-18B	47.3	60.7	52.2	71.2	44.6	40.0	31.0	26.3	46.7
Med+C4-18B	43.6	57.3	53.6	70.8	43.4	36.9	31.0	26.2	45.4

Dataset (Real Tok.)	ARC-C	SciQ	PubMedQA	MathQA	MMLU	Avg
C4-18B	25.3	66.6	50.6	22.0	23.1	37.5
C4-75B	24.3	66.8	46.4	22.1	23.0	36.5
QA+C4-18B	27.3	69.8	56.0	21.4	22.9	39.5
Med+C4-18B	6.1	67.6	53.6	21.7	23.1	38.4

Table 4: **Medium Sized Model Evaluation:** (Left) Evaluation of 350M parameter LLMs trained for 75B tokens on General understanding Tasks. This table shows the performance across various datasets, focusing on general reasoning, language understanding, and common sense comparing training. (Right) Evaluation Specialized Knowledge Tasks. This table presents the performance on tasks that require specific domain knowledge such as science, medicine, mathematics, and logic.

Dataset (Real Tok.)	ARC-E	BoolQ	Wino.	PIQA	HellaSwag	TruthfulQA	OBQA	LogiQA	Avg
C4-35B	51.6	55.2	56.1	75.2	58.2	36.0	33.6	27.8	49.2
C4-150B	52.2	54.4	56.4	75.6	58.6	34.3	35.8	27.1	49.3
Med+C4-35B	53.2	57.0	55.7	74.8	57.6	36.5	33.2	26.2	49.3
QA+C4-35B	55.1	63.3	55.7	75.6	57.9	41.4	34.0	26.1	51.1

Dataset (Real Tok.)	ARC-C	SciQ	PubMedQA	MathQA	MMLU	Avg
C4-35B	28.0	74.3	55.0	22.0	24.3	40.7
C4-150B	28.5	75.8	55.4	22.6	25.4	41.5
Med+C4-35B	29.6	74.3	46.2	22.9	25.2	39.6
QA+C4-35B	30.3	76.8	62.2	23.0	26.1	43.7

Table 5: **Left)** Evaluation of $\sim$ 1.3B parameter LLMs trained for 150B tokens on General Understanding Tasks. This table shows the performance across various datasets, focusing on general reasoning, language understanding, and common sense comparing training . **Right** Evaluation on Specialized Knowledge Tasks. This table presents the performance on tasks that require specific domain knowledge such as science, medicine, mathematics, and logic.

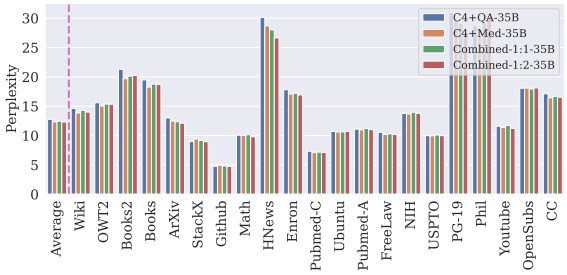


Figure 11: **Combining multiple styles:** Perplexity across all domains of the Pile comparing combining multiple styles of synthetic data. Models are 1.3B parameters trained for a total of 150B tokens. We see small perplexity improvements from combining multiple styles.

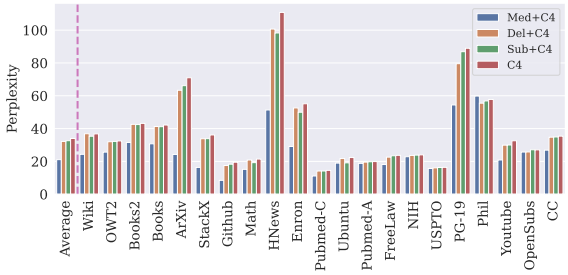


Figure 12: **Is re-phrasing same as any augmentation?** We compare perplexity on the Pile for different augmemntation strategies. 350M parameter models are trained for a total of 15B tokens. **WRAP** (Medium + C4) performs significantly better than traditional augmentations.

data in Q/A style. Overall, we note that there is a small improvement on the average perplexity on the Pile by combining multiple styles.

G.2 Method Ablations

RQ5: Does synthetic data improve over augmentations? Are the gains observed by pre-training on synthetic data the same as pre-training with augmentations? In order to test this, we consider two popular text augmentation baselines—synonym replacement and random deletion using the NL-Augmenter library (Dhole et al., 2021). We pre-train a 350M parameter model for 15B tokens in order to conduct this set of experiments. The total pool size is only about 1.5B tokens, meaning that the model would have to repeat data around 10 times during the pre-training phase, unless augmented over. As seen in the perplexity analysis in Figure 12, the models trained on augmented data perform significantly worse than those trained on combinations of real and synthetic data. This suggests that synthetic data enhances the learning process, and is not merely another form of augmentation.

RQ6: How does the style of synthetic data impact performance on specialized domains? We compare the performance of various models trained on different styles of synthetic data. In particular, we generate four styles of synthetic data (easy, medium, hard, and Q/A) and evaluate the performance of training on combinations of each style across Pile subsets. The prompts to generate these synthetic data styles are outlined in Appendix I. Results corresponding to generations from a Vicuna-v1.3 model, and for a 128M model trained for 3B tokens are summarized in Figure 13. We see that

Dataset (Real Tok.)	ARC-E	BoolQ	Wino.	PIQA	HellaSwag	TruthfulQA	OBQA	LogiQA	Avg
Med+C4-35B	53.2	57.0	55.7	74.8	57.6	36.5	33.2	26.2	49.3
QA+C4-35B	55.1	63.3	55.7	75.6	57.9	41.4	34.0	26.1	51.1
Combined-1:1-35B	53.7	60.2	57.7	75.1	56.9	40.2	34.2	26.3	50.5
Combined-1:2-35B	54.3	62.0	57.0	75.6	58.2	39.5	36.2	25.4	51.0

Dataset (Real Tok.)	ARC-C	SciQ	PubMedQA	MathQA	MMLU	Avg
Med+C4-35B	29.6	74.3	46.2	22.9	25.2	39.6
QA+C4-35B	30.3	76.8	62.2	23.0	26.1	43.7
Combined-1:1-35B	30.9	77.1	61.2	23.0	23.9	43.2
Combined-1:2-35B	29.6	76.7	57.4	23.6	23.1	42.1

Table 6: **Combining multiple styles:** (Left) Evaluation of $\sim$ 1.3B parameter LLMs trained for 150B tokens on General Understanding Tasks. (Right) Evaluation of the same model on Specialized Knowledge Tasks. Results suggest that combining rephrasing styles does not yield performance benefit on zero-shot tasks compared to just Q/A style.

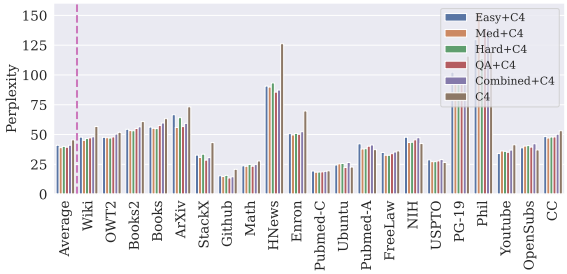


Figure 13: **Impact of style of synthetic rephrases:** Perplexity across all domains of the Pile comparing different styles of synthetic data. We train 128M parameter models for 3B tokens.

training with combinations of real C4 and synthetic data matching the style of the domain at evaluation improves performance. However, we find that no single synthetic data style performs the best across all domains, resulting in similar performance across training with combinations of real C4 data and each synthetic style variant. While knowing the best synthetic style to pre-train an LLM is impractical, an oracle that selects the best synthetic style across all domains will improve perplexity by 16%—indicating the importance of training with diverse data styles for LLM generalization, even when the underlying knowledge stays the same.

H LLM Leaderboard Few-shot Results

In our main experiments in Section 4 we demonstrate that LLMs trained with synthetic rephrases are a better backbone for zero-shot question-answering tasks as the model learns the question-answer format and style during pre-training. In this section, we show that improvements from pre-training on synthetic rephrases are still present even in few-shot settings where the model has access to test samples. To study few-shot performance, we evaluate on six tasks present in the OpenLLM-Leaderboard[2]:

1. ARC-Challenge (25 shot)
2. HellaSwag (10 shot)
3. MMLU (5 shot)
4. Truthful-QA (5 shot)
5. Winogrande (5 shot)
6. GSM8k (5 shot)

We evaluate two models trained for 300B and 350B tokens corresponding to roughly 85B and 100B unique C4 tokens respectively. Our findings show substantial improvements on the ARC-challenge benchmark, and Truthful-QA conssitent in the zero-shot settings and comparable performance across other datasets. Our models also perform better than the publicly released Falcon-1.3B model trained on the Refined Web dataset, and the Pythia-1.4B model, which was trained on Pile.

[2] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Dataset	ARC	Hellaswag	MMLU	TruthfulQA	WinoGrande	GSM8K	Avg
C4	31.7	62.1	26.7	33.4	57.9	0.9	35.5
Falcon-RW	35.1	63.6	25.3	36.0	62.0	0.5	37.1
Pythia-1.4b-Pile	32.7	55.0	25.6	38.7	57.3	0.8	35.0
TinLlama	33.9	60.3	26.0	37.3	59.5	1.4	36.4
QA+C4-85B (300K)	36.4	60.9	25.5	40.6	59.4	0.4	37.2
QA+C4-100B (350K)	35.5	60.5	26.8	40.6	61.3	0.3	37.5

Table 7: 1.3B 300K LLM Leaderboard Eval. Evaluation for **WRAP** is done on a single seed (1234).

I Rephrase Prompt Templates

We detail the prompts given to the Mistral-7B model to generate synthetic versions of the C4 dataset in specific styles. *Note: there are slight variations in the prompt that were used for other frozen LLMs, and no prompt was used for the T5 model.*

Easy Style

A style designed to generate content understandable by toddlers.

```
A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the questions. USER:
For the following paragraph give me a paraphrase of the same using a very small
vocabulary and extremely simple sentences that a toddler will understand:
```

Hard Style

A style designed to generate content comprehensible primarily to scholars using arcane language.

```
A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the questions. USER:
For the following paragraph give me a paraphrase of the same using very terse
and abstruse language that only an erudite scholar will understand. Replace
simple words and phrases with rare and complex ones:
```

Medium Style

A style designed to generate content comparable to standard encyclopedic entries.

```
A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the questions. USER:
For the following paragraph give me a diverse paraphrase of the same in high
quality English language as in sentences on Wikipedia:
```

Q/A Style

A style intended to convert narratives into a conversational format.

```
A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the questions. USER:
Convert the following paragraph into a conversational format with multiple tags
of "Question:" followed by "Answer:":
```

J Rephrase Examples from the MRPC Corpus

Original

```
The stock rose $2.11, or about 11 percent, to close Friday at $21.51 on the
New York Stock Exchange.

Revenue in the first quarter of the year dropped 15 percent from the
same period a year earlier.
```

Medium Style

```
The stock experienced an increase of approximately 11 percent, closing at
$21.51 on the New York Stock Exchange on Friday, with a rise of $2.11.

During the initial three months of the current year, there was a 15
percent decrease in revenue compared to the corresponding quarter of the
previous year.
```

Q/A Style

```
Question: What was the stock's closing price on Friday? Answer: $21.51
Question: How much did the stock rise on Friday? Answer: $2.11 or about 11
percent.

Question: What was the revenue drop in the first quarter compared to
the same period last year? Answer: The revenue dropped 15 percent.
```

K Rephrase Examples from C4 Corpus

Original

First round on stress at work survey. Answering the questionnaire is voluntary and all answers will be saved anonymously. Please fill in this questionnaire only if you have some work experience, part-or full time. Otherwise, you will not be able to answer some of the questions! Here is a the link to all language version.

Not that there's a thing wrong with frozen burgers. The key here is the meat seasonings, which are pretty strong and spicy and just GOOD, something else I think is really necessary in a turkey burger because ground turkey otherwise can be kind of flavorless. You'll need ground turkey, onion powder, chili powder, salt, pepper, and cayenne pepper for the burgers. Then the mayo takes garlic and onion. Then we need buns, clearly, swiss cheese, lettuce, and onion. I LOVE tomatoes but sometimes find that they get in the way of other flavors, so I left them off of this burger. Add them if you'd like to your array of toppings! First, we'll make the mayo. Grate the garlic directly into the mayo, add a pinch of salt, and squeeze in the lemon juice. Stir. Done! I love this. Then, we'll work on the burgers. Preheat a large skillet to medium-high heat with some olive oil, preheat the broiler to high, then add all the spices to the ground turkey.

Whether you like your velvet crushed, vibrant or head-to-toe, there's really no denying the sheer luxe and elegance of this timeless textile. Not only is it super stylish, it can actually be so wearable for day-to-day wear. Yes, really! This year it's all about embracing fun gem-toned velvety pieces. Long gone are the days when velvet was solely associated with dark moody shades of navy and black. Below we've rounded up the most covetable velvet pieces on the high street right now. We're already coming up with outfit ideas! Are you completely obsessed or beyond bored of it?

Save up to $8,086 on one of 1,258 Chrysler 200s near you. Find your perfect car with Edmunds expert and consumer car reviews, dealer reviews, car comparisons and pricing tools. We have 4,850,420. Research 2015 Chrysler 200 Sedan 4D 200C I4 prices, used values & 200 Sedan 4D 200C I4 pricing, specs and more. Many years ago, we wrote about the stalling problem with the 2011 Chrysler 200, and believe it or not, we still receive an occasional call regarding the problem.However, a much larger issue has monopolized the phone lines as of late 2015 Chrysler 200 transmission problems leaving drivers with check engine lights, harsh shifting, and the occasional loss of power. The 2015 Chrysler 200 can fetch a premium for its style and its horsepower—but rear-seat room and handling are better bargains elsewhere. Find out why the 2015 Chrysler 200 is rated 8.4 by The. Don't know where to find the perfect rims for your 2015 Chrysler 200 CARiD.com stores a massive selection of 2015 Chrysler 200 wheels offered in myriads of design and finish options, including chrome, black, silver, and so much more.

Medium Style

In the initial stage of the survey on work-related stress, participation is optional, and all responses will be kept confidential. It is important to note that only those with work experience, whether part-time or full-time, will be able to accurately respond to certain questions. To complete the questionnaire, please use the following link, which is available in multiple languages.

Frozen burgers are not inherently flawed, but the meat seasonings are a crucial element that elevates the flavor of the dish. Ground turkey can be quite bland without the right blend of spices, so it is essential to add onion powder, chili powder, salt, pepper, and cayenne pepper to the mixture. Additionally, garlic and onion should be used in the mayonnaise, which is necessary for the burger's topping. To make the mayo, grate the garlic directly into the mixture, add a pinch of salt, and squeeze in lemon juice. Mix well. The mayonnaise is a delicious addition to the burger topping. Now, let's work on the burgers. Heat a large skillet over medium-high heat with some olive oil, and preheat the broiler to high. Add the seasonings to the ground turkey and mix well. The burgers should be cooked on the skillet for about 5 minutes on each side, then transferred to the broiler to cook for an additional 2-3 minutes. The burgers are ready to be assembled on buns, along with swiss cheese, lettuce, and onion. Tomatoes can be added as a topping, but they may overpower other flavors. It is up to the individual to decide if they want to include them in their burger. In conclusion, frozen burgers can be tasty, but the meat seasonings are essential to add depth and flavor to the dish. The mayonnaise and toppings can also be customized to suit individual preferences.

The luxurious and elegant nature of velvet cannot be denied, whether it is crushed, vibrant, or head-to-toe. This timeless textile is not only fashionable but also wearable for daily wear. In recent years, there has been a shift towards embracing fun gem-toned velvety pieces, breaking away from the traditional association of velvet with dark moody shades of navy and black. Below are some of the most desirable velvet pieces available on the high street today, and we are already envisioning outfits around them. Are you completely enamored with velvet or simply beyond bored with it?

Discover savings up to $8,086 on one of 1,258 Chrysler 200s near you. Get expert and consumer car reviews, dealer reviews, car comparisons, and pricing tools from Edmunds. We have 4,850,420 listings for 2015 Chrysler 200 Sedan 4D 200C I4 prices, used values, and more. While we wrote about the stalling problem with the 2011 Chrysler 200, we still receive occasional calls regarding the issue. However, a more significant issue has taken over the phone lines recently - 2015 Chrysler 200 transmission problems that leave drivers with check engine lights, harsh shifting, and occasional loss of power. Despite the premium price for its style and horsepower, rear-seat room and handling are better bargains elsewhere. The 2015 Chrysler 200 is rated 8.4 by The Car Connection. If you're looking for the perfect rims for your 2015 Chrysler 200 CARiD.com offers a vast selection of 2015 Chrysler 200 wheels in various design and finish options, including chrome, black, silver, and more.

Q/A Style

```
Question: Is participating in the stress at work survey mandatory? Answer: No, it is voluntary. Question: Will all answers be saved anonymously? Answer: Yes, they will. Question: Who is eligible to take the questionnaire? Answer: Only individuals with work experience, whether part-time or full-time. Question: What happens if someone is not eligible to take the questionnaire? Answer: They will not be able to answer some of the questions. Question: Can you provide a link to all language versions of the questionnaire? Answer: Yes, here is the link.
```

```
Question: What makes these frozen burgers so good? Answer: The meat seasonings are strong and spicy, and they give the burgers a good flavor. Question: Do you think ground turkey can be flavorless? Answer: Yes, ground turkey can be flavorless, but adding spices and seasonings can help. Question: What do you need to make these turkey burgers? Answer: Ground turkey, onion powder, chili powder, salt, pepper, and cayenne pepper for the burgers. Then the mayo takes garlic and onion. Buns, swiss cheese, lettuce, and onion are also needed. Question: What about toppings? Answer: Add tomatoes if you'd like to your array of toppings. Question: How do you make the mayo? Answer: Grate the garlic directly into the mayo, add a pinch of salt, and squeeze in the lemon juice. Stir. Done! Question: How do you work on the burgers? Answer: Preheat a large skillet to medium-high heat with some olive oil, preheat the broiler to high, then add all the spices to the ground turkey.
```

```
Question: What do you think of velvet? Answer: Whether you like your velvet crushed, vibrant or head-to-toe, there's really no denying the sheer luxe and elegance of this timeless textile. Question: Is velvet super stylish? Answer: Yes, really! This year it's all about embracing fun gem-toned velvety pieces. Question: Was velvet solely associated with dark moody shades of navy and black? Answer: Long gone are the days when velvet was solely associated with dark moody shades of navy and black. Question: Do you have any covetable velvet pieces on the high street right now? Answer: Below we've rounded up the most covetable velvet pieces on the high street right now. Question: Are you completely obsessed or beyond bored of it? Answer: We're already coming up with outfit ideas!
```

Question: How can I save money on a Chrysler 200? Answer: Save up to $8,086 on one of 1,258 Chrysler 200s near you. Find your perfect car with Edmunds expert and consumer car reviews, dealer reviews, car comparisons and pricing tools. Question: What are the problems with the 2015 Chrysler 200? Answer: We have 4,850,420. Research 2015 Chrysler 200 Sedan 4D 200C I4 prices, used values & 200 Sedan 4D 200C I4 pricing, specs and more. Many years ago, we wrote about the stalling problem with the 2011 Chrysler 200, and believe it or not, we still receive an occasional call regarding the problem. However, a much larger issue has monopolized the phone lines as of late 2015 Chrysler 200 transmission problems leaving drivers with check engine lights, harsh shifting, and the occasional loss of power. Question: What are the benefits of buying a 2015 Chrysler 200? Answer: The 2015 Chrysler 200 can fetch a premium for its style and its horsepower—but rear-seat room and handling are better bargains elsewhere. Question: How is the 2015 Chrysler 200 rated? Answer: It’s rated 8.4 by The. Question: Where can I find the perfect rims for my 2015 Chrysler 200? Answer: CARiD.com stores a massive selection of 2015 Chrysler 200 wheels offered in myriads of design and finish options, including chrome, black, silver, and so much more.