Nemotron-CC: Transforming Common Crawl into a Refined Long-Horizon Pretraining Dataset

Dan Su*, Kezhi Kong*, Ying Lin*, Joseph Jennings, Brandon Norick,
Markus Kliegl†, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro

NVIDIA

*Equal contribution. †Correspondence to mkliegl@nvidia.com.

Abstract

Recent English Common Crawl datasets like FineWeb-Edu and DCLM achieved significant benchmark gains via aggressive model-based filtering, but at the cost of removing 90% of data. This limits their suitability for long token horizon training, such as 15T tokens for Llama 3.1. In this paper, we show how to achieve better trade-offs between accuracy and data quantity by a combination of classifier ensembling, synthetic data rephrasing, and reduced reliance on heuristic filters. When training 8B parameter models for 1T tokens, using a high-quality subset of our data improves MMLU by 5.6 over DCLM, demonstrating the efficacy of our methods for boosting accuracies over a relatively short token horizon. Furthermore, our full 6.3T token dataset matches DCLM on MMLU, but contains four times more unique real tokens than DCLM. This unlocks state-of-the-art training over a long token horizon: an 8B parameter model trained for 15T tokens, of which 7.2T came from our dataset, is better than the Llama 3.1 8B model: +5 on MMLU, +3.1 on ARC-Challenge, and +0.5 on average across ten diverse tasks. The dataset is available at `https://data.commoncrawl.org/contrib/Nemotron/Nemotron-CC/index.html`.

1 Introduction

Internet crawl is the largest source of unique tokens for training LLMs and can be seen as serving two main purposes: high-quality content and diversity. Recent English datasets derived from Common Crawl[1] such as FineWeb-Edu (Penedo et al., 2024) and DCLM (Li et al., 2024) have emphasized high-quality content that boosts benchmark accuracies over data quantity. They have demonstrated significant strides in achieving benchmark results competitive with some of the best closed models at a small scale (e.g., DCLM's 7B model trained over 2.6T tokens), primarily thanks to the use of model-based filters to extract high-quality educational and instructional content. However, this comes at the cost of data quantity: they remove around 90% of the data. Such aggressive pruning may not be the most effective strategy when training larger models over longer token horizons (e.g., Llama 3.1 includes 8B–405B parameter models, trained for 15T tokens (Dubey et al., 2024) and Gemma 2 27B was trained for 13T tokens (Team et al., 2024)). Both DCLM and FineWeb-Edu contain around 80% near-duplicates (1T and 0.2T unique tokens, respectively) (Ben Allal, 2024; Li et al., 2024) and to train on these datasets for many trillions of tokens implies seeing essentially the same samples many times during training. This could lead to inferior models, as Muennighoff et al.

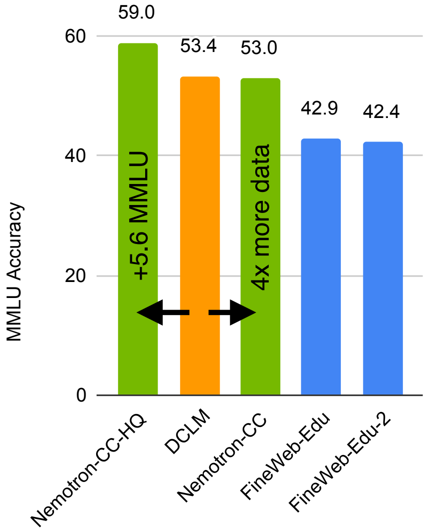

Figure 1: MMLU scores for 8B parameter models trained for 1T tokens. Compared to DCLM, our methods enable us to either create a 4× larger dataset of similar quality or increase the MMLU using a high quality subset of the tokens. Having a larger dataset, in the sense of unique real tokens, is crucial when training over long horizons such as 15T tokens.

[1] `https://commoncrawl.org/`

(2024) find there are diminishing returns after four epochs compared to training on more unique tokens.

In this paper, we show how to achieve a better trade-off between benchmark accuracy and data quantity with a combination of classifier ensembling, synthetic data generation, and reduced reliance on heuristic filters. Our main contributions are:

1. We propose a method for transforming English Common Crawl into a 6.3T token long-horizon pretraining dataset, consisting of 4.4T globally deduplicated original tokens and 1.9T synthetically generated tokens. We release the dataset[2] under the Common Crawl Terms of Use and a reference implementation as part of the Apache 2.0 open-source NeMo Curator library.[3] The quality classifier models have been released as well.[4]

2. We prove the effectiveness of this method by comparing to the state-of-the-art open English Common Crawl datasets DCLM and FineWeb-Edu (Figure 1).

   (a) A 1.1T-token high-quality subset of our data achieves a 5.6 MMLU improvement over DCLM, showing the superiority of our method over a relatively short token horizon.

   (b) Our full dataset performs on par with DCLM while having 4× as many unique real tokens.

   (c) This larger size enables state-of-the-art results over long token horizons: An 8B parameter model trained for 15T tokens using a weighted version of our dataset achieves higher overall accuracy than Llama 3.1 8B, and in particular MMLU 70.3 vs. Llama's 65.3. Note that Llama 3.1 8B was also trained on 15T tokens (Dubey et al., 2024).

3. We conduct ablation studies and find:

   (a) Ensembling different model-based classifiers can help select a larger and more diverse set of high quality tokens.

   (b) Rephrasing can effectively reduce noise and errors in low-quality data and produce diverse variants with fresh unique tokens from high-quality data, leading to better results in downstream tasks.

   (c) Disabling traditional non-learned heuristic filters for high-quality data can further boost high quality token yield without hurting accuracy.

Finally, we remark that our overall guiding principle is to shift from a static, non-learned, heuristic pipeline towards a more learned flywheel whose performance will naturally get better over time. As our data improves, so will the LLMs we train, and these improved LLMs will in turn improve our data as we use them to generate better synthetic data and quality classifications.

2 Methods

In this section we explain our efforts to build the best English Common Crawl pretraining dataset for LLMs. Our efforts can be split into three folds. First, we talk about our efforts in boosting token yield by utilizing text extractor and heuristic filters more properly in Section 2.1. Second, we introduce the model-based quality labeling pipeline methods in Section 2.2. Third, we introduce our synthetic data generation method to further improve the data quality in Section 2.3. For a schematic overview of our final pipeline, please see Figure 3 in Appendix A.

2.1 HTML-to-text Extractor & Filter

Extracted texts from HTMLs are the foundation and major source of LLM pretraining dataset, so it is of great significance to analyze and understand the extraction tools for optimal data quality and token yield. Moreover, heuristic filters are often utilized to remove low-quality tokens with human-designed heuristics (Li et al., 2024; Parmar et al., 2024; Penedo et al., 2024; Dubey et al., 2024), which may also put good tokens at the risk of being removed. We carefully examine both aspects with the assist of the FineWeb-Edu classifier (Penedo et al., 2024), a model-based quality classifier that had shown effectiveness in identifying high-quality tokens that are significant in boosting the strength of LLMs.

[2] https://data.commoncrawl.org/contrib/Nemotron/Nemotron-CC/index.html

[3] https://github.com/NVIDIA/NeMo-Curator

[4] https://huggingface.co/nvidia/nemocurator-fineweb-nemotron-4-edu-classifier and https://huggingface.co/nvidia/nemocurator-fineweb-mixtral-edu-classifier

	#Tokens	#HQ tokens	#HQ +%
Trafilatura-filtered	994	80	-
Justext-filtered	1,380	104	28.6%
Justext	1,804	127	57.4%

Table 1: Extraction and filteration token count statistics (billion). Tokens counted after deduplication.

HTML-to-text Extraction We test two HTML-to-text extractors, Justext (Pomikálek, 2011) and Trafilatura (Barbaresi, 2021). Qualitatively, we view both extractors at the same level of quality. Quantitatively, we calculate token yields of both extractors on 13 selected snapshots of Common Crawl (see Appendix F). The statistics are reported in Table 1. We see that Justext can yield more tokens, notably more high-quality tokens (+28.6%) by the standard of Fineweb-Edu classifier (score 3, 4, and 5). We highlight that boosting unique token amount is of great importance when building long-horizon pretraining dataset, e.g., 15T tokens for Llama3.1. Even though there is a slight decline in the percentage of HQ tokens for Justext vs. Justext-filtered (7.0% vs. 7.5%), what we aim to maximize here is the absolute number of HQ tokens (127B vs. 104B). We will later sort the data into quality buckets, which enables exact control of the proportion of HQ vs. non-HQ data seen during training instead of reliance on the natural distribution for a particular extraction tool. After extraction, we apply filtering to keep only English text, as determined by pycld2[5] and the FastText lid176 language classifier[6] with threshold 0.3 (Joulin et al., 2016, 2017). We then apply global fuzzy deduplication as well as exact substring deduplication over eighths of snapshots (Lee et al., 2022), using the NeMo Curator library[7] and the deduplicate-text-datasets library,[8] respectively.

Filtering Conventionally, heuristic filters are leveraged to remove low-quality tokens from the pretraining dataset as a post-processing step (Li et al., 2024; Parmar et al., 2024; Penedo et al., 2024; Dubey et al., 2024). We revisit the filtering pipeline as in (Parmar et al., 2024). Such pipeline sequentially consists of a set of heuristic filters proposed by Raffel et al. (2020); Rae et al. (2021) and a perplexity filter based on a KenLM model (Heafield, 2011) trained on Wikipedia and books data (Wenzek et al., 2020). To quantitatively better understand the effectiveness of the filtering pipeline, we calculate the token yield and report the numbers in Table 1. We find the filtering pipeline removes a non-trivial portion of high-quality tokens (-18.1%) classified by FineWeb-Edu classifier from the dataset.

Given the impact that the heuristic filters have on the high-quality token yield, we propose to NOT apply such filters to the high-quality tokens distinguished by model-based quality classifers (described in the next section), but only use those on the low-quality splits. In the experiment section we empirically verify the impact of both the extractor and filter on pretraining data quality through downstream benchmarks. We refer readers to Section 3.3 for detailed results.

2.2 Model-based Quality Labeling

Recent work (Li et al., 2024; Penedo et al., 2024) use model-based classifiers to extract high-quality pretraining documents from English Common Crawl. However, both of the two quality classifiers have a limited recall (around 10%) of high-quality tokens (see Table 9), and this will become a bottleneck to train an LLM over a long horizon. Also, the quality labels assigned by the quality classifier are not necessarily aligned with LLM's downstream task performance. Therefore, we propose our ensemble-based quality labeling pipeline method. Specifically, we first build three quality classifiers, each of which has different high-quality preferences. Then, we ensemble the three classifiers to score all the documents, and split the crawl corpus into different quality buckets based on the quality score. Finally, we regroup the fine-grained document buckets into 5 different quality levels based on their corresponding performance on downstream task.

Quality Classifier Training Preparing pretraining documents with quality annotations is the first key step in building a quality classifier (Dubey et al., 2024; Abdin et al., 2024; Yang et al., 2024). Similar to the work (Penedo et al., 2024)[9], we constructed two versions of quality annotation data. We prompt Mistral 8x22B-instruct[10] and

[5] https://pypi.org/project/pycld2/

[6] https://fasttext.cc/docs/en/language-identification.html

[7] https://github.com/NVIDIA/NeMo-Curator

[8] https://github.com/google-research/deduplicate-text-datasets

[9] We use the same 460K document samples as in the FineWeb-Edu-Annotation dataset.

[10] https://mistral.ai/news/mixtral-8x22b/

Nemotron-340B-instruct (Adler et al., 2024), to score web documents from FineWeb based on their educational value on a scale from 0 to 5. We then fine-tune a linear regression model on top of the Snowflake-arctic-embed-m embedding model (Merrick et al., 2024) using the two different version of training sets. The two models have been trained for 20 epochs with a learning rate of 3e-4, with the embedding and encoder layers frozen, and we selected the checkpoint with the highest F1 score on the held-out validation set.

We also employ the DCLM classifier which is a fastText-based classifier released by Li et al. (2024). The DCLM classifier is trained on a combination of instruction-formatted data (Teknium, 2023) and high-scoring posts data from ELI5 subreddit (Fan et al., 2019), and has shown stronger performance in identifying high-quality pretraining tokens, compared to the FineWeb-Edu classifier (Penedo et al., 2024). The DCLM classifier will offer a new perspective in labeling high-quality pretraining documents, and will help increase the recall of high-quality tokens.

Quality Scoring and Bucketing First, we use each of the three classifiers to predict the quality scores for all the documents. Then based on the ranked quality score from each classifier, we rounded the model's output score to integers from 0 to 19. So that each score bucket will have around 5% of the documents, and bucket 19 will have the top 5% highest quality documents. We then assign the final quality score for each document by ensembling the three classifiers' integer score by a maximum operation. The number of documents distribution in each buckets will be skewed by the ensemble operation.

Quality Labeling In order to assign a quality label that is more aligned with their real performance on downstream tasks, we further group the fine-grained quality score predicted by three classifiers into 5 downstream quality categories. We used annealing to assess each data bucket's downstream task's quality. Specifically, we measure the quality of each bucket by continuous pretraining with 50B tokens on a 70% trained 8B models. We assign 66% of weight to the default data mix and 34% to the dataset that we are evaluating. By comparing the average performance of each bucket over 9 tasks, we group the 20 buckets into 5 big categories, with the final distribution shown in Table 2. For more details, please see Appendix C.

Quality Label	Buckets	# Tokens (B)	Token (%)
High	19	553	12.63
Medium-High	18	504	11.52
Medium	12-17	2,023	46.24
Medium-Low	7-11	894	20.43
Low	0-6	402	9.18

Table 2: Common Crawl quality labels statistics.

2.3 Synthetic Data Generation

Upon reviewing samples across the quality tiers, we observe that documents with lower scores tend to contain more noise and errors, while those scoring higher generally exhibit good writing and formatting. Therefore, we employ different strategies when generating data from low- and high-quality documents.

For low-quality data, our goal is to improve the quality by reducing noise and errors while preserving useful information, thereby decreasing training compute expenses. As shown by Maini et al. (2024), rephrasing web data using a medium-sized language model yields an enhanced parallel corpus of synthetic data, thereby reducing model perplexity and boosting its accuracy on downstream tasks. Unlike existing methods that create new content such as textbooks and short stories (Wang et al., 2023; Eldan and Li, 2023; Gunasekar et al., 2023), our rephrasing-based approach does not utilize the language model as a knowledge bank but focuses on transforming provided texts into another style, allowing it to operate with a lighter-weight model. We adopt the Wikipedia style prompt from (Maini et al., 2024) to rewrite low-quality documents (Prompt 5 in Appendix H), which effectively reduces errors and redundancies and improves formatting.

For high-quality data, we aim to obtain more unique tokens and condense essential knowledge. According to (Muennighoff et al., 2024), adding repeated tokens yields a diminishing return, especially after 4 epochs. For high-quality documents, we generate synthetic data using four additional prompts: (1) Diverse Question-Answer (QA) pairs: ask questions in various forms (e.g., yes/no question, open-ended question, multi-choice question) about factual information in the text and provide the correct answers; (2) Distill: rewrite the text into a concise and clear passage; (3) Extract knowledge: rewrite knowledge from the text and disregard uninformative content; (4) Knowledge list: extract key information from the text as an organized list.

We require the model to provide clear and concise responses while preserving factual information and concrete details such as numbers. The full prompts are shown in Appendix H.

As we increase the length of provided text, the model shows a tendency to produce over-simplified outputs with reduced detail. Therefore, we chunk each document into segments, each of which contains one or more complete lines and is shorter than a specific token limit.[11] Over-length lines exceeding the token limit are discarded.

```
Question: Which year did the United Nations
implement the 2030 agenda for SDGs?
Answer: January 1, 2016

Question: What are the three key dimensions of
sustainable development covered by the SDGs?
Answer: (a) economic growth, (b) social
inclusion, and (c) environmental protection

Question: Which of the following can flossing
prevent? A) Cavities B) Gum disease C) Both A and
B D) Neither A nor B
Answer: C) Both A and B

Question: Is flossing important even if you
brush your teeth twice a day?
Answer: Yes, flossing is important as it reaches
areas that brushing alone cannot.
```

Figure 2: Examples of generated question-answer pairs.

Our post-processing steps include removing incomplete results, eliminating specific Markdown formatting (e.g., double asterisks), stripping away prefixes of certain patterns (e.g., "*Here is a paraphrased version:*" and "*Paraphrased Text:*"), removing quotation marks enclosing the entire response, and filtering out under-length outputs (i.e., shorter than 50 tokens). For Wikipedia results, we concatenate passages generated from segments belonging to the same original document. For Diverse QA Pairs results, we shuffle the generated question and answer pairs, retain up to a number based on the length of the segment, and append the pairs to the end of the segment.

Using the instruct version of Mistral NeMo 12B[12] with FP8 inference, a top-p value of 0.9, and a sampling temperature of 0.5, we synthesize over 1.8T tokens as Table 3 shows, including 336.3B tokens from low-quality documents and 1.5T tokens from high-quality documents. We do not use medium-quality documents for synthetic data generation due to time and resource constraints. We employ TensorRT-LLM[13] and NeMo-Skills[14] to enable large-scale data synthesis.

Source	#Raw	Prompt	#Synthetic
Low	403.0	Wikipedia	336.3
High	451.3	Wikipedia	372.9
		Diverse QA Pairs	499.5
		Distill	157.6
		Extract Knowledge	303.6
		Knowledge List	203.2

Table 3: Synthetic data token count statistics (billion).

2.4 Putting It All Together

Dataset	Total	Unique	Synthetic
FineWebEdu-2	5.4	1.1	-
FineWebEdu	1.3	0.2	-
DCLM	3.8	1.0	-
Nemotron-CC	6.3	4.4	1.9
Nemotron-CC-HQ	1.1	0.6	0.5

Table 4: Dataset sizes in trillions of tokens. "Unique" shows the estimated number of tokens after global fuzzy deduplication of the real tokens.

Combining the techniques above to the 99 snapshots CC-MAIN-2013-20 through CC-MAIN-2024-30 of Common Crawl, we create a 6.3T token dataset (Nemotron-CC), consisting of 4.4T globally deduplicated tokens and 1.9T synthetically derived tokens. This dataset has roughly 4× more unique tokens than FineWebEdu-2 and DCLM, since both of those datasets only underwent a sharded form of approximate deduplication and contain roughly 80% fuzzy duplicates (Ben Allal, 2024; Li et al., 2024). To enable a fairer comparison over relatively short token horizons, we thus also consider a 1.1T token high quality subset of our data (Nemotron-CC-HQ), consisting of just the highest-scoring real and diverse QA pairs synthetic data. The size breakdown of the datasets is shown in Table 4.

3 Experiments

3.1 Experiment Setup

Training Setup We use the open source Megatron-LM library[15] (Shoeybi et al., 2019) to train standard 8B parameter transformer LLMs.

[11] The token limit is set to 512 for Wikipedia, 2,000 for Distill, 1,400 for Extract Knowledge and 1,000 for Diverse QA Pairs and Knowledge List, including tokens from the prompt and chat format.

[12] https://mistral.ai/news/mistral-nemo

[13] https://github.com/NVIDIA/TensorRT-LLM

[14] https://github.com/NVIDIA/NeMo-Skills

[15] https://github.com/NVIDIA/Megatron-LM

Dataset	ARC-E	ARC-C	H	W	RACE	PIQA	SIQA	CSQA	OBQA	MMLU	Avg
FineWebEdu-2	71.9	44.7	75.4	67.0	36.8	79.5	45.2	25.5	43.8	42.4	53.2
FineWebEdu	73.6	48.0	70.7	64.6	**38.0**	76.4	43.5	30.0	44.4	42.9	53.2
DCLM	74.7	47.0	76.3	69.1	36.5	79.7	45.6	44.1	44.0	53.4	57.0
Nemotron-CC	75.3	50.7	75.9	67.8	37.9	**80.5**	45.1	47.7	44.2	53.0	57.8
Nemotron-CC-HQ	**78.8**	**52.9**	**76.6**	**69.4**	36.4	80.1	**46.6**	**55.8**	**45.4**	**59.0**	**60.1**

Table 5: Results for 8B parameter models trained on 1T tokens (73% English Common Crawl from the tested dataset, 27% the same, fixed non-Crawl datasets). The models were evaluated on ARC-Easy, ARC-Challenge, Hellaswag, Winogrande, RACE, PIQA, Social IQA, Commonsense QA, Openbook QA, and MMLU.

Model	ARC-E	ARC-C	H	W	RACE	PIQA	SIQA	CSQA	OBQA	MMLU	Avg
Llama 3.1	82.4	55.0	79.3	**74.7**	**39.1**	**81.2**	**48.3**	**70.6**	**46.0**	65.3	64.2
Ours	**82.7**	**58.1**	**80.8**	73.8	37.8	81.1	47.4	69.9	45.4	**70.3**	**64.7**

Table 6: Comparison of our 8B parameter model vs Llama 3.1 8B. Both were trained for 15T tokens. The numbers for Llama 3.1 are from our own lm-evaluation-harness setup described in Section 3.1 and may not match Meta's publicly reported numbers, as Meta made various customizations to the benchmarks.

The hyperparameter details are shown in Appendix D.

Data Blend Unless otherwise noted, we train for 1T tokens on a blend of 73% English Common Crawl data and 27% a fixed mix of specialized code, papers, books, patents, and Wikipedia datasets (Adler et al., 2024). When comparing datasets, we vary only the 73% English Common Crawl portion. See Table 12 in Appendix D.

Evaluation Setup We use the open source LM Evaluation Harness library[16] (Gao et al., 2023) to evaluate on the following ten common sense and reasoning tasks (reported metric in parentheses): ARC-Easy and ARC-Challenge (normalized accuracy) (Clark et al., 2018), Hellaswag (normalized accuracy) (Zellers et al., 2019), Winogrande (accuracy) (Sakaguchi et al., 2021), RACE (accuracy) (Lai et al., 2017), PIQA (normalized accuracy) (Bisk et al., 2020), Social IQA (accuracy) (Sap et al., 2019), Commonsense QA (accuracy) (Talmor et al., 2019), Openbook QA (normalized accuracy) (Mihaylov et al., 2018), and MMLU (accuracy) (Hendrycks et al., 2021).

3.2 Main Results

Short Token Horizon (1T) To validate the quality of our datasets, we first train standard 8B parameter transformer LLMs over a relatively short 1T token horizon. The results are shown in Table 5. Our high quality dataset (Nemotron-CC-HQ) shows accuracy gains over DCLM and FineWeb-Edu on all tasks except RACE. In particular, there is a 5.6 MMLU and 3.1 average gain over DCLM. This shows the effectiveness of our classifier ensembling and synthetic data even in the non-data-constrained setting. Our complete 6.3T token dataset (Nemotron-CC) gives MMLU and average accuracies roughly on par with DCLM. But since this dataset contains 4× more unique real tokens, we expect it to be superior in data-constrained settings like 15T token training runs.

Long Token Horizon (15T) Our dataset contributed 7.2T of the tokens used to train an 8B model for 15T tokens. As shown in Table 6, our model achieves a higher average accuracy than Llama 3.1 8B, which was also trained for 15T tokens, including an MMLU score of 70.3 vs. Llama's 65.3. This shows that our dataset is indeed suitable for state-of-the-art training over long token horizons. For more details on this experiment, please see Appendix E.

3.3 Ablation Study

To further investigate the contribution and effect of each module in our method, we conducted thorough ablation studies.

Extractor & Filter Comparison As we have discussed in Section 2.1, by deploying Justext instead of Trafilatura and removing filter from the post-processing step, we can attain significantly 57.4% more high-quality tokens. We also conduct ablation studies to better understand the impact of the extractor selection and the removal of filter through

[16] https://github.com/EleutherAI/lm-evaluation-harness

downstream benchmarks. We carry out four 8B-1T experiments. We report the benchmark scores in Table 7. Beyond the token-yield benefit by leveraging Justext instead of Trafilatura and not using heuristic filters, we see that combining these two does not impact the downstream task accuracies with only marginal differences (comparing Trafilatura filtered vs. Justext unfiltered). Moreover, when we ONLY remove filter from high-quality tokens, the results get further improved (comparing Justext unfiltered vs. Justext HQ unfiltered). In particular, MMLU gets boosted by +2%. Note that, the motivation behind removing filter is to boost token yield, especially on high-quality tokens due to the notable scarcity of such. Given the experimental results and considering the overall growth in token yield, we opt to only remove filter from high-quality tokens.

Exp name	MMLU	Avg (non-MMLU)
Trafilatura filtered	55.4	60.6
Justext filtered	54.1	**60.9**
Justext unfiltered	55.5	60.3
Justext HQ-unfiltered	**57.5**	60.6

Table 7: Ablation studies on extractor and filter. HQ means high-quality data judged by FineWeb-Edu classifier (score 3,4,5). HQ-unfiltered means filtering is applied only to LQ data. See Appendix G for more details.

Classifiers Comparison Assembling different classifiers to label the document quality is one of the key steps in constructing our datasets, so we did thorough analysis and comparison of the component.

We did a detailed comparison of two types of classifiers that we employ in our method: the FineWeb-Edu classifier which score document quality based on their educational-level, and the DCLM-based classifier which value the informativeness of the document. We compare the high-quality documents predicted by the two classifiers on a randomly selected Common Crawl Snapshot (CC-MAIN-2021-21). Table 8 shows the document statistics comparison. We can see that only 10% of the documents are predicted as high quality by both classifiers, while 35.4% documents are predicted as high quality by FineWeb-Edu classifier only, and 54.4% of documents are predicted as high-quality by DCLM classifier. Therefore, ensembling different classifiers can increase the recall of high-quality documents from Common Crawl.[17]

We further compare each of the classifiers with the ensembled method[18] by their downstream tasks' performances. We pretrain 8B parameters LLMs with 1T tokens, using the high-quality documents labeled by different classifiers on randomly selected 13 Common Crawl snapshots (see Appendix F). Table 9 shows the detailed comparison on different evaluation tasks. We can see that the ensembled method greatly boost the high-quality tokens percentage from 9% to 25%, while still achieving the highest general language understanding performance on average on all the tasks. The ensembled method also outperforms the FineWeb-Edu classifier and the DCLM classifier, in terms of the high-quality token percentage, and is on-par or slightly better on the 9 evaluation tasks. This is very important since more unique high-quality tokens is the key in pretraining larger LLMs on longer tokens horizons.

What	#Docs	Total unique(%)
Total unique in union	11,359,655	100.0%
In intersection	1,152,821	10.1%
In FineWeb-Edu only	4,022,294	35.4%
In DCLM only	6,184,540	54.4%

Table 8: High-quality documents overlap analysis.

Evaluating Synthetic Data As Table 10 shows, this ablation study aim to answer two questions: (1) Does rephrasing low-quality improve accuracies on downstream tasks? (2) Can synthetic data help offset the decreasing value of duplicated data reported in (Muennighoff et al., 2024)? To answer these questions, we train four 8B models with the same hyperparameters on different blends of 1T tokens: (1) LQ-Base: original Common Crawl data including low-quality documents; (2) LQ-Synthetic: an augmented version of LQ-Base where the low-quality documents are rephrased; (3) HQ-Base: a blend containing eightfold high-quality documents and less low- and medium-quality documents; (4) HQ-Synthetic: a variant of HQ-Base where 4 repetitions of the high-quality documents are swapped out for synthetic datasets.

By comparing the results between LQ-Base and LQ-Synthetic, we can see that rephrasing low-

[17] Detailed URL domain comparison can be found in Appendix B

[18] Note that we did not employ FineWeb-Edu classifier in our ensemble for license issue, since it is trained with annotations from Llama3.

Classifier	**HQ(%)**	**ARC-E**	**ARC-C**	**H**	**W**	**RACE**	**PIQA**	**SIQA**	**CSQA**	**OBQA**	**MMLU**	**Avg**
FineWeb-Edu	8%	77.7	50.1	74.9	67.3	**39.5**	78.8	45.8	53.6	43.0	55.4	59.0
DCLM	11%	76.0	49.2	**76.5**	**70.2**	38.2	**80.8**	33.9	55.2	45.8	56.0	58.4
Ours-mistral	9%	75.8	49.2	75.9	66.9	37.5	80.1	**46.2**	46.9	44.8	53.2	58.1
Ours-nemotron-340B	14%	76.3	**50.3**	75.6	67.5	37.8	80.2	34.3	54.0	**46.2**	54.9	58.0
Ours-ensembled	**25%**	**78.0**	49.7	75.3	67.1	37.2	79.6	45.7	**56.8**	44.8	**56.4**	**59.4**

Table 9: Different classifiers comparison. Our ensemble method includes the three classifiers: Ours-mistral, Ours-nemotron-340B and DCLM.

Blend	**ARC-E**	**ARC-C**	**H**	**W**	**RACE**	**PIQA**	**SIQA**	**CSQA**	**OBQA**	**MMLU**	**Avg**
LQ-Base	67.7	41.8	**75.2**	**67.1**	37.4	78.8	45.3	36.9	41.0	**48.2**	52.5
LQ-Synthetic	**71.3**	**45.2**	75.0	66.9	37.4	**79.4**	**46.2**	**41.6**	**42.8**	47.1	**54.0**
HQ-Base	74.2	47.7	**74.8**	66.9	37.3	78.2	**46.0**	47.3	43.6	53.4	55.8
HQ-Synthetic	**76.7**	**49.2**	74.5	**67.3**	**38.2**	**78.8**	45.2	**47.9**	**45.8**	**53.6**	**56.7**

Table 10: Impact of incorporating synthetic data.

quality data leads to 1.50 absolute gains on average score. We also observe noticeable boosts from 1.80% to 4.75% on ARC-Easy, ARC-Challenge, OpenbookQA, CommonsenseQA; however, we also encounter slight accuracy drops on some tasks, which may indicate potential misinformation introduced by data synthesis. Current practices typically utilize data curation approaches to detect and eliminate noisy examples. Due to time and resource constraints, we leave the detailed exploration of this issue for future efforts.

The comparison between HQ-Base and HQ-Synthetic shows that swapping 4 out of 8 epochs of high-quality data with a mix of synthetic datasets improves accuracy on most benchmarks. This improvement could potentially result from two factors: the incorporation of fresh unique tokens and styles that enable the model to learn specific abilities (e.g., question answering) or absorb knowledge more efficiently.

4 Related Work

The Phi series of models pioneered training on small amounts of very high quality data, including curated Web and synthetic data (Gunasekar et al., 2023; Li et al., 2023; Abdin et al., 2024). However, their focus is on shorter token horizons and they share limited details. FineWeb-Edu and DCLM are the main points of comparison for our paper (Li et al., 2024; Penedo et al., 2024). We build upon their core idea of model-based filtering, but show how to improve the filtering and data quantity through a combination of other techniques. Other English Common Crawl datasets such as C4, DOLMA, Gopher, RefinedWeb, TxT360 largely focus on extraction and non-learned heuristics (Penedo et al., 2023; Soldaini et al., 2024; Rae et al., 2021; Raffel et al., 2020; Tang et al., 2024). Just as for FineWeb-Edu and DCLM, the core pipeline we started from incorporates many of these ideas, but our paper describes how to modify and go beyond these non-learned techniques to achieve state-of-the-art accuracy and diversity. Concurrent work Zyda-2 shows how to filter, cross-deduplicate, and combine the FineWeb-Edu, DCLM, Zyda-1, and Dolma-CC datasets into a higher-accuracy and larger whole (Tokpanov et al., 2024). In contrast, we focus on techniques for the creation of a new English Common Crawl dataset rather than combinations or modifications of existing datasets. Finally, many works have focused on creating multilingual datasets (Xue et al., 2021; Brack et al., 2024; Abadji et al., 2022; Wenzek et al., 2020; Kudugunta et al., 2023). We leave extension of our ideas beyond English to the future.

Synthetic datasets have been widely used in language model pre-training and post-training. In (Cheng et al., 2024), instruction-response pairs are synthesized for pre-training. In (Eldan and Li, 2023), the authors show that smaller or simpler models trained on a synthetic dataset of short stories are capable of generating fluent and consistent stories. Similarly, smaller models trained using high-quality synthetic textbook and exercise datasets can achieve impressive high accuracy on coding benchmarks (Gunasekar et al., 2023; Li et al., 2023). These approaches typically require a powerful language model, such as GPT-3.5 and GPT-4 in (Eldan and Li, 2023), to synthesize new contents. Instead, (Maini et al., 2024) shows that compact models such as Qwen-1.8B and Mistral-7B are adequate to rephrase web data. This approach generates diverse, high-quality synthetic data that effectively lowers model perplexity and boosts performance across benchmarks. We adopt this main idea, but explore more prompts and show how to specialize them for low and high quality data.

5 Conclusion

For producing long-horizon pretraining tokens for LLMs from English Common Crawl data, we showed how to improve upon the state of the art and achieve better trade-offs between benchmark accuracy and data quantity, as measured by number of unique real tokens. Specifically, we showed the efficacy of ensembling model-based quality filters, rephrasing low and high quality documents, and reducing the reliance on non-learned heuristics. The dataset is public and split by quality level and type (actual data vs. different types of synthetic data), enabling the community to do further experiments on quality vs. diversity and how to build effective short and long horizon curricula.

6 Limitations

Some of the key limitations of our work are as follows. For the model-based filter ensembling and quality bucketing, we only had time and resources to try a single strategy. Though it is effective, it is possible this could be improved upon in future work, especially to improve the sensitivity at the higher-quality end of the spectrum. For the rephrased data, we did not verify the factual accuracy or fidelity to the original contents. More work is required to understand the risks of hallucinations or loss of content diversity in this setting and how to mitigate them. We also only looked at rephrasing low and high quality data. It could be interesting to explore how to best rephrase medium quality data as well. We did not do ablations on all parts of the pipeline. There is probably room for improvement with, for example, the language identification. Overall, we tried our methods only on English text. More work is needed to adapt our methods to other languages.

Finally, we did not decontaminate the dataset, as there is not yet a strong consensus on how to best do this and the impact is uncertain and debated, especially for large models trained over large token horizons. We note that the datasets we compare against (FineWeb-Edu, DCLM) were released without decontamination, and the model we compare against (Meta Llama 3.1) was also trained on contaminated data. DCLM reports some contamination analysis, but the findings suggest contamination is not a key factor: e.g., MMLU actually increases after decontamination, and DCLM does better than FineWeb on MMLU, even though FineWeb has more MMLU contamination (see Section 4.6 and Appendix N in Li et al. (2024)). Still, it would be interesting to better understand the impact of contamination for different model sizes and different token horizons, and we hope the community can explore such questions on this public dataset.

Acknowledgments

We thank the Common Crawl Foundation for hosting the dataset. We thank Pedro Ortiz Suarez for valuable feedback that improved the paper and Greg Lindahl for help with improving the data formatting and layout. Furthermore, we thank Ayush Dattagupta, Randy Gelhausen, Smita Ithape, Vibhu Jawa, Nirmal Kumar Juluru, Vineeth Kalluru, Mehran Maghoumi, Arham Mehta, Ranjit Rajan, Janaki Vamaraju, Ryan Wolf, and Sarah Yurick for help with NeMo Curator and for open sourcing Nemotron-CC recipes and classifiers as part of NeMo Curator.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 4895–4901.

Adrien Barbaresi. 2021. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131. Association for Computational Linguistics.

Loubna Ben Allal. 2024. Most of the data is duplicated? `https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu/discussions/7`. Accessed: October 24, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Manuel Brack, Malte Ostendorff, Pedro Ortiz Suarez, José Javier Saiz, Iñaki Lacunza Castilla, Jorge Palomar-Giner, Alexander Shvets, Patrick Schramowski, Georg Rehm, Marta Villegas, and Kristian Kersting. 2024. Community OSCAR: A community effort for multilingual web data. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 232–235, Miami, Florida, USA. Association for Computational Linguistics.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pretraining: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

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 preprint arXiv:1803.05457*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining. *Preprint*, arXiv:2412.15285.

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.

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*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36:67284–67296.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. Datacomp-LM: In search of the next generation of training sets for language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *arXiv preprint arXiv:2405.05374*.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Data, data everywhere: A guide for pretraining dataset construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10695.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36:79155–79172.

Jan Pomikálek. 2011. Removing boilerplate and duplicate content from web corpora. *Disertacnı práce, Masarykova univerzita, Fakulta informatiky*.

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. *Journal of machine learning research*, 21(140):1–67.

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.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. 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 4463–4473.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Cun Mu, Victor Miller, Xuezhe Ma, Yue Peng, Zhengzhong Liu, and Eric P. Xing. 2024. Txt360: A top-quality llm pre-training dataset requires the perfect blend. `https://huggingface.co/spaces/LLM360/TxT360`. Accessed: October 24, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak

Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants. https://huggingface.co/datasets/teknium/OpenHermes-2.5. Accessed: October 24, 2024.

Yury Tokpanov, Paolo Glorioso, Quentin Anthony, and Beren Millidge. 2024. Zyda-2: a 5 trillion token high-quality dataset. *Preprint*, arXiv:2411.06068.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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*, pages 4791–4800.

A Pipeline Overview

An overview of the pipeline is shown in Figure 3.

B Comparison of FineWeb-Edu and DCLM Classifier

Different classifiers have different standards for high-quality documents. Thus, ensemble multiple classifiers will help increase the recall of high-quality documents. We did a detailed comparison of two of the classifiers that we employ in our method: the FineWeb-Edu classifier which score document quality based on their educational-level, and the DCLM based classifier which value the informativeness of the document.

We compare the high-quality documents predicted by the two classifiers on one Common Crawl snapshot (dated 2021-21). Table 8 show the document statistics comparison. We further show the detailed URL domains comparison between the two classifiers' predictions in Table 11. We can see that each classifier has their own high-quality domain preferences. Among the top 1k domains, only 368 domains are in the intersection. Therefore, ensemble of different classifiers can help increase retrieving more high-quality documents from Common Crawl.

C Bucket Comparison

To better understand the quality of data in each of our 20 data buckets, we carry out ablation studies to test their benchmark accuracies. For each study, we take a 900B-token checkpoint and continue the pretraining for 50B more tokens. For 34% of the 50B tokens we used the bucket data being tested, while we fixed the other 66% as the same data distribution of the 900B pretraining process to make sure the distribution did not shift too much. See Figure 4 for the results. The average accuracy is calculated across 13 downstream tasks. Note that Bucket 19 greatly outperforms all other buckets and the differences within bucket 12-18 are marginal. We used the results here as a reference when designing the quality labels in Table 2.

D Training Details: Ablations

As mentioned in Section 3.1, we use the open source Megatron-LM library[19] (Shoeybi et al., 2019) to train 8B parameter transformer LLMs for 1T tokens. The key hyperparameters are

[19] https://github.com/NVIDIA/Megatron-LM

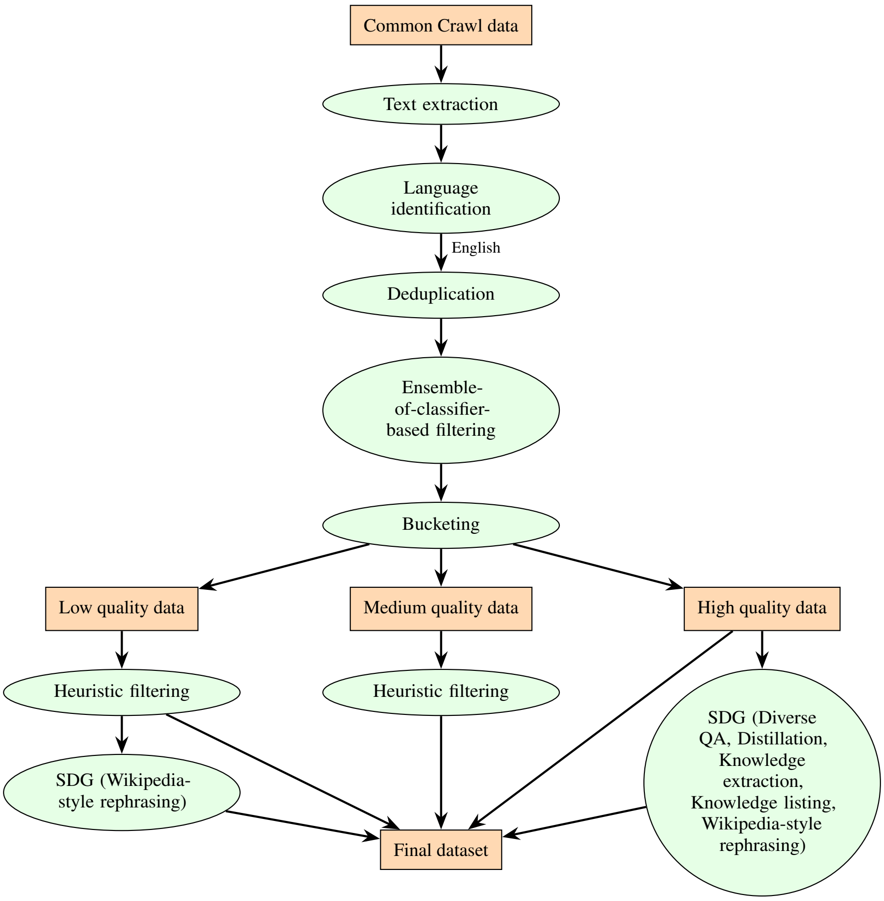

Figure 3: Pipeline overview

Top domains and domain overlap analysis =>368 domains are in top 1k domains of both						
FineWeb-Edu Top Domains	**Count**	**DCLM Top Domains**	**Count**	**Top 1k Domains**		
				Intersection (368)	In FineWeb-Edu only	In DCLM only
wordpress.com	39228	wordpress.com	85378	123helpme.com	111papers.com	4archive.org
thefreedictionary.com	20420	stackexchange.com	64831	24houranswers.com	3dprint.com	4channel.org
stackexchange.com	17853	livejournal.com	36521	abc.net.au	aafp.org	4hw.com.cn
britannica.com	14761	medium.com	27347	abovetopsecret.com	aappublications.org	5winebar.com
ipl.org	13132	fandom.com	13986	academickids.com	abs.gov.au	aawsat.com
medium.com	11539	ipl.org	12282	adafruit.com	accessgenealogy.com	abc11.com
nih.gov	10624	answers.com	10790	adobe.com	achrnews.com	abc30.com
igi-global.com	9136	nih.gov	9091	alchetron.com	acm.org	abc7chicago.com
slideplayer.com	8460	typepad.com	8078	aljazeera.com	adidasshoesoutletwholesale.com	able2know.org
answers.com	8103	commonsensemedia.org	7772	allegancountyedc.com	adslspeedtest.net	aceshowbiz.com
wikipedia.org	6867	wsj.com	7652	allinterview.com	aero-net.org	activerain.com
dictionary.com	6763	imdb.com	7263	amazon.com	agwired.com	addicted2success.com
en-academic.com	5292	theatlantic.com	7008	americanbar.org	ahdictionary.com	additudemag.com
sciencemag.org	5254	yahoo.com	5921	angelfire.com	ajol.info	agingcare.com
brainscape.com	5129	fanfiction.net	5499	answers.com	akjournals.com	agnostic.com
encyclopedia.com	4698	huffpost.com	5471	antiessays.com	aleteia.org	airmilescalculator.com
nasa.gov	4615	adobe.com	5182	apple.com	alison.com	airportia.com
slideserve.com	4538	scribd.com	4948	archive.org	all-creatures.org	alarabiya.net
scribd.com	4430	thefreedictionary.com	4847	arduino.cc	allaboutheaven.org	alex-in-wonderland.com
kiddle.co	4323	mathworks.com	4655	arstechnica.com	allthatsinteresting.com	alexa-gueguen.com

Table 11: High Quality Documents Domains Comparison. 368 Top Domains are in the intersection.

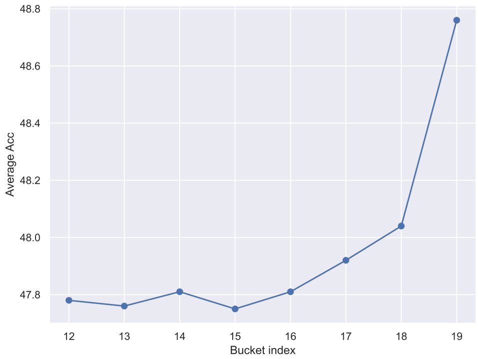

Figure 4: Ablation study on the buckets.

Category	Blend %
English Common Crawl	73
Books and patents	9
Papers	9
Code	5
Conversational	3
Wikipedia	1

Table 12: Data blend for the experiments with 8B parameter transformer LLMs trained for 1T tokens. Experiments in this paper varied only the 73% English Common Crawl portion.

as follows: We use 32 transformer layers with hidden dimension 4096, 32 attention heads, and SwiGLU activations (Shazeer, 2020). For the attention, we use grouped query attention with 8 query groups (Ainslie et al., 2023). We use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 1e{-}8$, weight decay 0.1, and the cosine learning rate schedule with peak learning rate at 3e-4 and minimum learning rate at 3e-6. A single training run takes about 40 hours using 1024 NVIDIA H100 GPUs.

The data blend breakdown for these experiments is shown in Table 12. Experiments in this paper varied only the 73% English Common Crawl portion.

E Long-Horizon Curriculum Details

For the 15T token training run, a two-phase curriculum was employed that is described in more detail in Feng et al. (2024). The first phase of 9T tokens used 59% English Common Crawl data (5.31T) and the second phase of 6T tokens used 31% (1.86T), for a combined total of 47.8% (7.17T). In the first phase, we used medium, medium-high, and high quality data (real and synthetic), and in the second phase we used only high quality data (real and synthetic).

F Common Crawl Snapshots

For the main datasets, we used the 99 snapshots CC-MAIN-2013-20 through CC-MAIN-2024-30.

The thirteen Common Crawl snapshots we use in some of the analysis and 1T token experiments are CC-MAIN-2023-23, CC-MAIN-2023-14, CC-MAIN-2023-06, CC-MAIN-2022-49, CC-MAIN-2022-27, CC-MAIN-2022-05, CC-MAIN-2021-43, CC-MAIN-2021-21, CC-MAIN-2021-04, CC-MAIN-2020-45, CC-MAIN-2020-29, CC-MAIN-2020-05, CC-MAIN-2019-35.

G Extractor & Filter Ablation

The Avg tasks include ARC-Easy, ARC-Challenge, Hellaswag, Winogrande, RACE, PIQA, Commonsense QA, Openbook QA.

Note that we only use FineWeb-Edu classifier for the quality labels of this ablation study and analysis. We do not use it in the final preparation of our dataset. See Section 2.2 for the details of our classifiers being used eventually to prepare the data.

H Prompt Templates

Prompts 1-5 show the prompt templates we use for synthetic data generation.

```
Task: Read the text, ask questions and answer them.

Follow these instructions:
1. Ask diverse questions that require different cognitive skills or cover different aspects of the
text.
2. Ask questions in various forms such as:
  - Yes/No questions that require determining whether a statement is true or false.
  - Open-ended questions that begin with words like what, how, when, where, why and who.
  - Multi-choice questions that offers two or more options to choose from. Include the options in the
   question.
  - Comparison questions that compare two quantities or objects and determine the relationship
  between them.
  - Reading comprehension questions that test the ability to understand and analyze the text.
  - Problem-solving questions that test the ability to solve mathematical, physical, or logical
  problems.
3. Focus on asking questions about factual information, important knowledge, or concrete details in
the text.
4. Write questions and answers using clear and concise language.
5. Use plain text. Do not use Markdown.
6. Each question and answer pair should be on a separate line. Tag the question with ”Question:” and
the answer with ”Answer:”.

Text:
[DOCUMENT SEGMENT]

Task:
After reading the above text, ask up to 8 questions and provide the correct answers following the
instructions. Give your response in this format:

Here are the questions and answers based on the provided text:
- Question: [first question] Answer: [first answer]
- Question: [second question] Answer: [second answer]
....
```

Prompt 1: Prompt template: Diverse QA pairs

```
Your task is to read and paraphrase the provided text following these instructions:
- Aim to create a condensed but accurate and informative version of the original text, not a
simplistic summary.
- Capture and preserve the crucial information, key concepts, important values, and factual details
in the original text, while making it more readable and accessible.
- Retain technical terms, specialized vocabulary, and complex concepts.
- Retain examples, explanations of reasoning processes, and supporting evidence to maintain the text'
s depth and context.
- Only include information that is present in the original text. Do not adding new or unsubstantiated
 claims.
- Write in plain text.

Here is the text:
[DOCUMENT SEGMENT]

Task:
After thoroughly reading the above text, paraphrase it in high-quality and clear English following
the instructions.
```

Prompt 2: Prompt template: Distill.

```
Review the text and extract the key information. Follow these instructions:
- Carefully read the above text and provide a concise and organized list of factual information,
concrete details, key concepts, and important numbers and statistics extracted from the text.
- Ensure each point is clear, specific, and supported by the original text.
- Ensure the extract text is information-dense and easier to learn from.
- Do not add titles or headings.

Text:
[DOCUMENT SEGMENT]

Task:
Extract the factual information, concrete details, and key concepts from the above text following the
 instructions.
```

Prompt 3: Prompt template: Knowledge list.

```
Your task is to rewrite knowledge from the provided text following these instructions:
- Rewrite the text as a passage or passages using easy-to-understand and high-quality English like
sentences in textbooks and Wikipedia.
- Focus on content in disciplines such as humanities, social sciences, natural sciences, technology,
engineering, math, law and legal, business, management, art, education, agricultural sciences,
politics, and history.
- Disregard content that does not contain useful facts or knowledge.
- Retain examples, explanations of reasoning processes, and supporting evidence to maintain the text'
s depth and context.
- Do not add or alter details. Only restate what is already in the text.
- Write in plain text.
- Do not add titles, subtitles, note, or comment.

Text:
[DOCUMENT SEGMENT]

Task:
Rewrite facts and knowledge from the above text as a passage or passages following the instructions.
```

Prompt 4: Prompt template: Extract knowledge.

```
For the following paragraph give me a diverse paraphrase of the same in high quality English language
 as in sentences on Wikipedia. Begin your answer on a separate line with "Here is a paraphrased
version:".

Text: [DOCUMENT SEGMENT]
```

Prompt 5: Prompt template: Wikipedia-style rephrasing (Maini et al., 2024).