MEASURING BIAS OF WEB-FILTERED TEXT DATASETS AND BIAS PROPAGATION THROUGH TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we investigate biases in pretraining datasets for large language models (LLMs) through dataset classification experiments. Building on prior work demonstrating the existence of biases in popular computer vision datasets, we analyze popular open-source pretraining text datasets derived from CommonCrawl including C4, RefinedWeb, DolmaCC, RedPajama-V2, FineWeb, DCLM-Baseline, and others. Despite those datasets being obtained with similar filtering and deduplication steps, neural networks can classify surprisingly well which dataset a single text sequence belongs to, significantly better than a human can. This indicates that popular pretraining datasets have their own unique biases or fingerprints. Those biases remain even when the text is rewritten with LLMs. We demonstrate that these biases propagate through training: Random sequences generated by models trained on those datasets can be classified well by a classifier trained on the original datasets.

1 INTRODUCTION

In 2011, Torralba & Efros (2011) proposed the dataset classification experiment to examine biases present in common computer vision dataset. The paper demonstrated that computer vision reserachers can relatively easily classify to which dataset an image from a computer vision dataset popular at the time (e.g., PASCAL, Caltech101, ImageNet,...) belongs to. Moreover, classifiers can be trained to relatively reliably classify which dataset an image belongs to. Torralba & Efros (2011) concluded that "despite the best efforts of their creators, the datasets appear to have a strong built-in bias". While some of the bias can be accounted for by isolating specific objects the different datasets focus on, Torralba & Efros (2011) found that the biases are still present in some form, even if those effects are isolated.

Recently, Liu & He (2024) revisited the dataset classification experiment in the current era of large-scale and diverse vision datasets like YFCC Thomee et al. (2016), DataComp Gadre et al. (2023), and LAION Schuhmann et al. (2022). Those datasets are large in scale and are collected to train generalizable representations, as opposed to datasets collected for a specific purpose (for example for urban scene understanding Cordts et al. (2016)). Liu & He (2024) found, perhaps surprisingly, that even for those large and diverse datasets, classifiers can relatively accurately assign a single image to belong to one of those datasets.

In this paper we study the bias of pretraining datasets for large language models (LLMs), as well as the propagation of this bias through training with dataset classification experiments.

We consider the most popular open web-based datasets for general purpose LLMs, specifically C4 Raffel et al. (2020), RefinedWeb Penedo et al. (2023), DolmaCC Soldaini et al. (2024), RedPajama-V2 Together Computer (2023a), FineWeb and FineWeb-edu Penedo et al. (2024), and DCLM-BASELINE Li et al. (2024). These datasets consist of sequences of text of average length ranging from 477 to 1235 tokens (see Appendix E), and are obtained by preprocessing and filtering CommonCrawl. These datasets are considered to be diverse and cover the large variety of text that is available in the internet, and are commonly used for pretraining and pretraining research of LLMs.

Our main findings are:

- We demonstrate that sequences from open pretraining datasets can be well classified to belong to a certain dataset, highlighting unique biases or fingerprints inherent in datasets. For example, the datasets C4, RefinedWeb, DolmaCC, RedPajama-V2, and FineWeb are all obtained from CommonCrawl using a very similar pipeline consisting of deduplication and heuristic quality filtering with similar rules. Yet an LLM trained to classify whether a single sequence is part of C4, RefinedWeb, and RedPajama-V2 achieves 80% accuracy, well above chance (33.3% accuracy), and significantly above the human accuracy.
- Even when rewriting sequences with LLMs to remove some of the formatting, sequences can be well distinguished by a classifier.

- The biases inherent in the pretrained data propagate through training: Sequences generated at random from LLMs pretrained on DCLM-Baseline, RefinedWeb, and FineWeb-Edu can be well distinguished (89.15% accuracy) with a classifier trained on the original training data.
- LLMs pretrained on several domains can generate random sequences that reflect the proportion of the domains in the pretraining mixture. By classifying the output sequences with a classifier trained to distinguish between the original domains, we can estimate the mixture proportions.

2 Related Work

This work is inspired by Torralba & Efros (2011)'s dataset classification experiment for vision datasets and Liu & He (2024)'s recent work that revisited the dataset classification experiment in the context of modern large scale dataset. Liu & He (2024) found, similar as we find for language datasets, that images from the large scale and diverse computer vision datasets YFCC Thomee et al. (2016), CC Changpinyo et al. (2021), and DataComp Gadre et al. (2023) can be accurately classified as belonging to one of those datasets.

A variety of works study the problem of classifying LLM generated text. Hans et al. (2024) and many phrase this as a classification problem Solaiman et al. (2019); Tian et al. (2024); Hu et al. (2023). Guo et al. (2023) demonstrate that ChatGPT generated answers can be well distinguished from human answers by a classifier, if the text is sufficiently long . In this work we focus on distinguishing popular pretraining text datasets with a classifier, not AI vs human generated text.

Shi et al. (2023) and Maini et al. (2024) consider the problem of detecting pretraining data based on blackbox access of LLMs; specifically given a text and blackbox acccess to an LLM, was the LLM trained on that text? In Section 5 we study the loosely related problem whether a classifier trained to distinguish training data can distinguish data generated by the LLMs trained with it.

Carlini et al. (2021) and Nasr et al. (2023) attempt to extract training data from LLMs. They show that an adversary can extract verbatim text sequences from the model's training data by querying the LLM with no previous information of the training set. In section 6 we assume the training domains are known, and estimate the proportion of each domain in the training mixture.

3 Setup and datasets considered

Throughout this paper, we perform dataset classification for language datasets as follows. Each dataset consists of a set of sequences, and a classifier is trained to distinguish the sequences from N such datasets. We measure performance on a test set consisting of an equal amount of sequences from each of the N datasets.

As classifier, we use a standard transformer model for next-word prediction that is fine-tuned on a training set of sequences to perform N-way classification. See Section 4 and Appendix A for the details of the model used, training details, and ablation studies with parameters of the classifier and other classifiers.

3.1 Distinguishing data from different sources

Language models are often pretrained on data from different sources, for example LLama 1's Hugo Touvron et al. (2023)'s pretraining data, and reproduction of the data, RedPajama-1T Together Computer (2023b), consists of the sources C4, CommonCrawl (CC), Arxiv, Github, Wikipedia, and Stack Exchange. Some of those sources are very easy for humans to distinguish, for example Github (containing code) and C4/CC (containing little code). Thus, it is perhaps not surprising that we find that 6 way classification of the Redpajama-1T sources (C4, CC, Arxiv, Github, Wikipedia, Stack Exchange) yields an accuracy of 98.25%.

3.2 Datasets considered

We consider seven of the largest and most popular open datasets for pretraining general-purpose LLMs based on web-filtered data. The datasets are based on web crawls from CommonCrawl, a nonprofit foundation that provides a publicly available web archive. Much of the text extracted by CommonCrawl consists of text that is not useful for training, like incomplete sentences, HTML artifacts, and duplicate texts. Thus all of the datasets we consider for LLM training are obtained by the dataset creators by extracting text from CommonCrawl data, filtering, cleaning, and deduplicating the text data.

All datasets are obtained by i) extracting text using parsers like `resiliparse` or using CommonCrawl's pre-extracted text and language filtering, ii) applying heuristic filtering (examples are removing very short texts, and removing texts with curly brackets { since those indicate code), iii) deduplication (for example, identical or nearly identical webpages are filtered out), and iv) machine learning based filtering (for example filtering based on a classifier trained to distinguish high-quality data from average data). The exact choices of those steps have a significant effect on the composition of the datasets are how good the models are that are trained on them.

We consider the following seven datasets:

C4: The Colossal Clean Crawled Corpus Raffel et al. (2020) is a popular dataset consisting of 360B Tokens obtained from CommonCrawl text extracted in April 2019, followed by i) language filtering, ii) heurisitc filtering, and iii) deduplication.

FineWeb: FineWeb Penedo et al. (2024) is a 15T token dataset extracted from CommonCrawl through i) language and ii) heuristic quality filtering and iii) deduplication. The heuristic filters and deduplication steps are carefully chosen based on ablation studies.

RefinedWeb: RefinedWeb Penedo et al. (2023) is a large scale (5T tokens, 600B publicly available) obtained from CommonCrawl by i) language and ii) heuristic filtering and iii) deduplication.

Dolma CC: Dolma Soldaini et al. (2024) is an open corpus of 3T tokens from different sources. The biggest proportion, about 2.4T tokens, is obtained from CommonCrawl. We consider the CommonCrawl part, which was obtained by first downloading about a quarter of the most recent CommonCrawl data in 2023 (i.e., data from 2020-05 to 2023-06), and was processed with i) language and ii) heuristic quality filtering and iii) deduplication. As a machine learning based quality filtering step, for each sequence the perplexity was computed with an LLM to measure Wikipedia-likeness (following the CCNet pipeline Wenzek et al. (2019)) and partitioned into into head, middle, and tail by perplexity; we consider the head and middle parts.

RedPajama-V2: RedPajama-V2 Together Computer (2023a) is a corpus of 30T filtered and deduplicated tokens also processed with i) language and ii) heuristic quality filtering, and iii) deduplication. We consider the 20.5T token part of the corpus consisting of English speaking documents, as for all other datasets we also consider the English part only. The data contains a broad coverage of CommonCrawl, and comes with quality annotations that enables slicing and filtering the data. As a machine learning based quality filtering step, for each sequence the perplexity was computed with an LLM trained to identify Wikipedia-like documents and partitioned into head, middle, and tail, and head and middle was retained. We consider head and middle as for Dolma.

DCLM-BASELINE: DCLM-BASELINE Li et al. (2024) was obtained from CommonCrawl through i) text extraction with resiliparse and language and ii) heuristic quality filtering, deduplication, and iv) machine learning based quality filtering. All steps where chosen by ablation studies to obtain a dataset so that models trained on it perform well. The final, machine learning based filtering step is important and is trained to classify instruction-formatted data from OpenHermes 2.5 and high-scoring posts from the r/ExplainLikeImFive subreditt from RefinedWeb. Models trained on this dataset perform very well on common benchmarks.

FineWeb-Edu: FineWeb-Edu Penedo et al. (2024) is obtained from FineWeb through machine learning based quality filtering to obtain data with educational text, and consists of 1.3T tokens. Models trained on FineWeb-Edu perform very well on knowledge and reasoning benchmarks such as MMLU Hendrycks et al. (2021).

All datasets are based on web crawls, are large in scale, and are broad, i.e., not focused on a specific topic or area (such as Arxiv, Github, etc). Therefore it is perhaps surprising that sequences of these datasets can be relatively reliably distinguished, as we find in the next section.

4 Dataset classification experiments

We perform dataset classification experiments by default with a 160M standard autoregressive transformer model that we pretrain compute optimally Hoffmann et al. (2022) on 3.2B tokens. After pretraining, we remove the last layer in the transformer network and replace it by a classification head, similar to the reward model in RLHF Ouyang et al. (2022). Our head has N outputs, where N is the number of classes.

We group the seven datasets into three distinct categories based on their preprocessing techniques. Category 1 incorporates standard language processing, heuristic filtering, and deduplication, and consists of the C4, FineWeb, and RefinedWeb datasets. Category 2 performs the steps in Category 1 as well as additional light filtering based on Wikipedia perplexity scores, and includes the Dolma and RedPajama-V2 datasets. Category 3 also performs the Category 2 steps

# Classes	Category 1			Category 2		Category 3		Accuracy
	C4	FineWeb	RefinedWeb	DolmaCC	RedPajama-V2	DCLM	FineWeb-Edu	
	✗		✗		✗			80.50%
	✗	✗			✗			79.27%
			✗	✗	✗			77.99%
		✗		✗	✗			75.74%
	✗	✗	✗					74.76%
	✗			✗	✗			74.09%
		✗	✗	✗				73.04%
3	✗		✗	✗				72.90%
	✗	✗		✗				68.84%
		✗	✗		✗			67.55%
	✗					✗	✗	94.12%
				✗		✗	✗	92.94%
			✗			✗	✗	89.76%
		✗				✗	✗	85.16 %
					✗	✗	✗	84.55%
	✗	✗	✗		✗			70.31%
	✗		✗	✗	✗			68.98%
4		✗	✗	✗	✗			67.88%
	✗	✗		✗	✗			67.45%
	✗	✗	✗	✗				64.44%
5	✗	✗	✗	✗	✗			60.70%

Table 1: Classification accuracy across different combinations from the three dataset categories. Despite the similarity in the filtering techniques, high classification accuracies are observed, specially for category 3.

and in addition carefully selected machine learning-based text filtering techniques and consists of DCLM-BASELINE and FineWeb-Edu.

We conduct a comprehensive set of classification experiments, testing all possible combinations of 3-way, 4-way, and 5-way classification using the five datasets from categories 1 and 2. Additionally, we perform five 3-way classification experiments that pair the two datasets from category 3 with each of the five datasets from the other categories. We also report the results for all possible 2-way combinations in Table 5 in Appendix B. We use 160M training tokens per dataset, i.e., 480M for 3-way, 640M for 4-way, and 800M for 5-way classification. As a test set we take 8192 randomly sampled unseen sequences from every dataset.

As seen in Table 1, across all dataset combinations, the classifiers consistently achieve high accuracy. Particularly high accuracy is obtained when classifying sequences from the datasets DCLM-BASELINE and FineWeb-Edu vs the other datasets, which is perhaps not surprising since those sequences are relatively distinct, see Appendix D for examples.

However, it is perhaps surprising that sequences from the datasets processed with similar language and heurisitc filtering and deduplication steps are so easily distinguishable. Humans perform significantly worse in assigning text sequences to datasets than the neural networks as our study in Section 4.2 demonstrates.

4.1 Details of the experiments and ablation studies

In this section we perform ablation studies justifying the choice of our classifier. The ablation studies are performed on the 3-way classification of C4, FineWeb, and RefinedWeb. Unless stated otherwise, we use the default 160M model with 480M training tokens (160M per dataset) for every ablation study. Further ablation studies are in Appendix C.

We start by scaling the model size, pretraining data, and training data. We find that high accuracy is obtained with different model sizes and dataset set sizes.

Scaling model and pretraining data: The default model has 160M parameters and is pretrained on 3.2B tokens. We study the impact of the model size by considering the model sizes 25M, 87M, 160M, and 410M pretrained compute optimally on 0.5B, 1.7B, 3.2B, and 8.2B tokens, respectively. The finetuning set size is kept constant at 480M tokens.

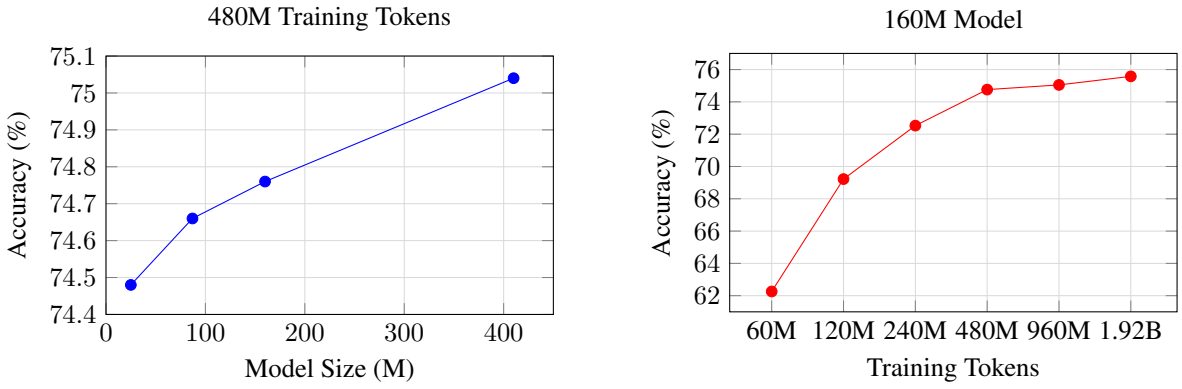

Figure 1: **Left:** Scaling model size and pretraining data with constant training data. **Right:** Scaling training data with constant model size and pretraining data. Scaling model size and pretraining data has a minimal effect on the accuracy, but the effect of the training data is more prominent.

The results are in Figure 1, left panel. For the model sizes considered, the model size and pretraining data amount play a relatively insignificant role; the difference in classification accuracy between the smallest and largest model is only 0.56%.

Scaling classification training data: The default training set size used to finetune the pretrained model is 480M tokens. In this study we consider the 160M model pretrained with 3.2B tokens, and finetune it for classification with training sets of different sizes. We start with a training set size of 60M tokens, and then double it up to 1.92B tokens, i.e., we consider the following sizes: 60M, 120M, 240M, 480M, 960M, and 1.92B.

The results are in Figure 1, right panel. The accuracy initially significantly increases with the training data, but saturates close to 480M, which is our default training set size. Quadrupling the training data from 480M to 1.92B tokens only gives a gain of 0.82% in accuracy.

Training without pretraining: All classification experiments are carried out by finetuning a model pretrained to predict the next token. To study the impact of pretraining for classification accuracy, we train a randomly initialized model (without any pretraining) directly for classification. This gives an accuracy of 71.59%, which is 3.17% less than the pretrained model (74.76%). Since pretraining improves performance by 3.17%, which is significantly more than increasing the model size, we choose to work with the pretrained model throughout all our experiments.

4.2 Classification accuracy achieved by humans

Our experiments show that classifiers can accurately differentiate between datasets, even when the differences are subtle to human perception. This is particularly striking with datasets like C4 and FineWeb, which are obtained with similar filtering steps and appear nearly identical to humans. Two example sequences from each dataset are in Figure 2, more examples are in Appendix D.

We conducted a dataset classification experiment to measure human performance. The task is binary classification between C4 and FineWeb. We gave two machine learning researchers several sequences from each dataset for inspection. For testing, the researchers were given 50 unlabeled sequences from each set and had unlimited time to classify them. On average, it took them 1.5 hours to label the 100 sequences.

The researchers achieve an average accuracy of 63%, only 13% above random guessing. In contrast, the 160M sized classifier trained on 320M tokens performs significantly better attaining 88%, which highlights the model's ability to identify subtle patterns that are not easily distinguishable by humans.

4.3 Rewrite experiments

The datasets are difficult to distinguish for humans. To gain more insights into what makes the sequences distinguishable, we rewrite original data with an LLM and classify the rewritten texts. We rephrase the original datasets with OpenAI's GPT-4o-mini model prompted with the following three prompts:

C4

•Made it back, can I come inside for a change? Made of glass and falling fast all the way! Thanks for correcting Tokyo Police Club - Miserable lyrics!
•Jamie Oliver is a famous CHEF from the UK. Here you can learn how to make scramble eggs in three different ways: English, French and American way! ENJOY IT!

FineWeb

•Short-term and long-term changes in the strength of synapses in neural networks underlie working memory and long-term memory storage in the brain.
•Yesterday, we indulged in all the goodness of sweets, so I thought it only appropriate that we feature the other side of the coin: Salty. Now, I'm a girl who loves her potato chips.

Figure 2: Sample text sequences from C4 and FineWeb. From human perspective, it is hard to identify patterns to distinguish between the datasets.

Prompt	Original	Prompt 1	Prompt 2	Prompt 3
Accuracy	87.37%	83.19%	79.50%	66.02%
C4 av. length	425	436	408	371
FineWeb av. length	621	627	580	489

Table 2: Classification accuracy between C4 and FineWeb when rephrased with 3 different prompts. The accuracy drops as the prompts allow for more deviation from the original text. Average length is measured as the average number of tokens per sequence in the test set.

Prompt 1: "*Rewrite the following text sentence by sentence while preserving its length and the accuracy of its content. Maintain the overall format, structure, and flow of the text:*"

Prompt 2: "*Rewrite the following text while preserving its length and the accuracy of its content:*"

Prompt 3: "*Rewrite the following text while preserving its length and the accuracy of its content. Do not use newlines, new paragraphs, itemization, enumeration, and other formatting, unless it is important or appropriate for better readability:*"

The prompts encourage increasing degrees of deviation of the rephrased texts from the originals. This can be seen in Figure 7 in Appendix D, which shows an original text sequence from C4 rephrased with the three prompts. The rephrased text from Prompt 1 is the closest to the original, followed by Prompt 2, and then Prompt 3. Prompt 1 preserves the formatting and rephrases primarily through replacing a few words. Prompt 2 alters the format slightly, introducing changes such as line breaks. It also changes the text structure by making it more compact, for example, the final sentence in Prompt 2 (Figure 7) conveys the same meaning as the original text and Prompt 1 but in a more concise form, see table 2 for the average sequence lengths. Prompt 3 significantly alters both the structure and format of the original text.

We consider the binary classification task between C4 and FineWeb, i.e., we train a classifier to distinguish rephrased C4 from rephrased FineWeb. Using each of the prompts, we rephrase 160M training tokens and 8192 test sequences from every dataset. We use the default 160M transformer as the classifier, and report the results in Table 2.

Interestingly, while the rephrased text are more difficult to distinguish, when rewirtten with Prompt 1 and Prompt 2, the sequences are still easy to distinguish for a classifier. This suggests that the distiguishability of the texts does not overly rely on wording and formatting of the text.

4.4 Removing formatting and classifying based on word frequencies only

We have so far seen that biases exist within popular text datasets, and persist even when the text is rephrased. We next attempt to isolate which features are responsible for this bias.

Removing formatting. We remove structural formatting of C4 and FineWeb. Specifically, we remove all newlines, itemization and enumeration patterns, including numbers, bullet points and similar markers that commonly denote list elements, excessive spaces, and other special characters such as tabs and carriage returns with regular expressions (regex). The resulting text is a single continuous block of text. Essential punctuation markers, such as periods and commas, are not modified as they contribute to the text's meaning rather than just its formatting or display.

We train the 160M model to classify between regex preprocessed C4 and FineWeb. We use 320M training tokens (160M tokens per dataset), and 8192 test sequences. The accuracy is 72.42%, about 15% less than the accuracy on the original datasets (87.37%). This drop in accuracy suggests that models detect patterns in formatting that are important for classification. However, the fact that classification remains relatively accurate even after unifying the formatting, suggests that there are biases beyond format and structure present within the datasets.

Bag of Words (BoW) is a simple text classification method that represents text as a collection of unique words, disregarding format, grammar, word order, or context. Each text sequence is transformed into a vector with the frequency of each unique word within the text. For instance, BoW transforms the following two texts: "*I like apples but not bananas*" and "*I like bananas but not apples*" to the same exact vector representation.

We use BoW to distinguish between C4 and FineWeb, and achieve a classification accuracy of 63.45%. Classification with BoW is higher than a random guess despite reducing each text sequence to a vector with the frequency of words within it. BoW disregards any semantic relationship between the words, it is based solely on the vocabulary used, which suggests that the vocabulary distributions of C4 and FineWeb are different.

4.5 Dataset Categorization

To get a deeper understanding of the characteristics that differentiate the datasets, we obtain a random sample from each of the seven datasets, and categorize its text sequences into the 13 thematic categories shown in Figure 3. We use Chat-GPT-4o-mini's API by prompting it to classify the text to the most appropriate category. If none of the categories are appropriate, it chooses "Other".

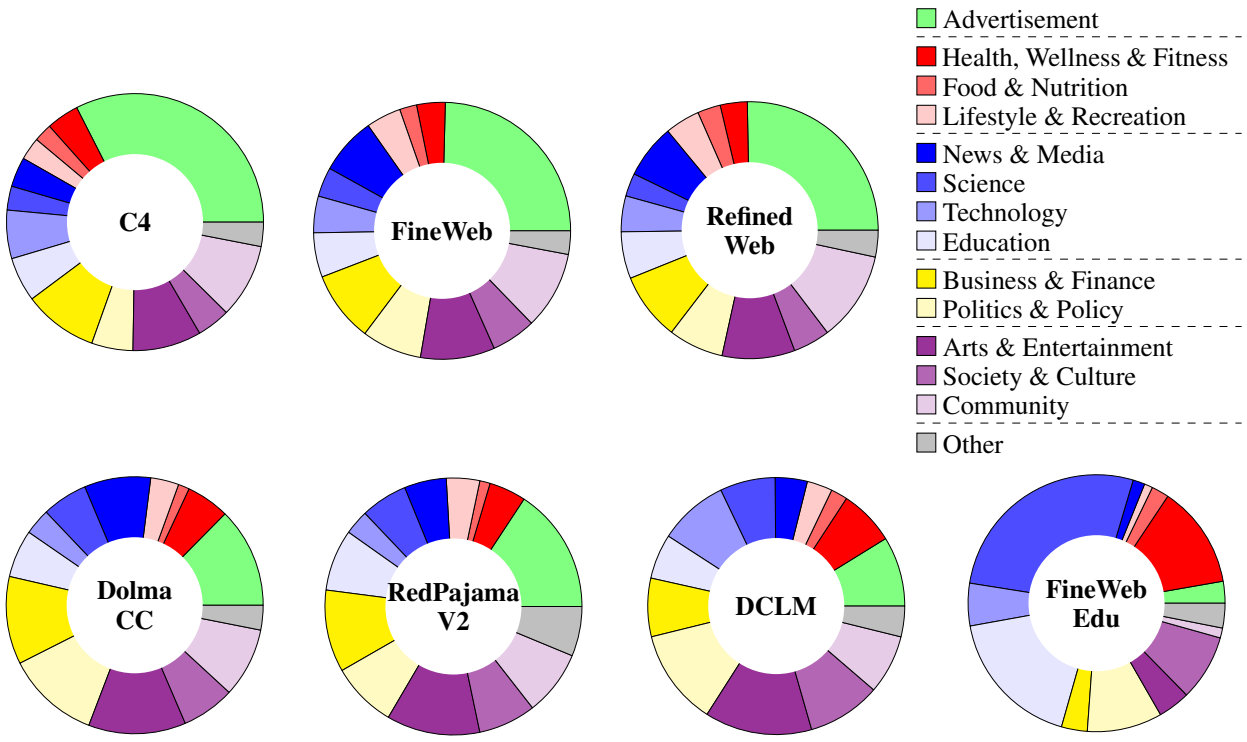

Figure 3: Categorization of datasets into 13 thematic categories. Similarly filtered datasets have comparable categorical distributions.

The visualizations in Figure 3 reveal that the content distribution is close for similarly filtered datasets. For instance C4, FineWeb, and RefinedWeb are filtered with standard heuristics and deduplication, and therefore have a comparable distribution. DolmaCC and RedPajama-V2 are additionally filtered with respect to Wikipedia perplexity and thus also exhibit similar distributions.

The machine learning filtered datasets (Dolma, RedPajama, DCLM, and FineWeb-Edu) have significantly less advertisement content than C4, FineWeb, and RefinedWeb. Also, FineWeb-Edu is filtered for educational content, and

therefore has the majority of its sequences categorized as "Science" followed by "Education". Such content differences across datasets provide a basis for their distinguishability.

4.6 Discussion

Our experiments suggest that format, vocabulary, and content are all characteristics that enable differentiating between the datasets. While some of these features are obvious, others are subtle and are only detected by neural networks.

The best example for features that are easily identifiable, is the formatting in DCLM. The DCLM team used resiliparse to extract text, which very frequently inserts new lines between the sentences (i.e., ends sentences with \n\n). This makes DCLM sequences particularly distinct, see Appendix D. This is also reflected in the high accuracy a model attains when classifying DCLM sequences as seen in Tables 1 and 5.

Another example is FineWeb-Edu. Most of the sequences are educational and scientific, and thus classifying educational and scientific sequences as FineWeb-Edu sequences can work relatively well, see Appendix D for examples.

For other datasets, it is harder to point to specific individual patterns that make them distinguishable, for example we found C4 and FineWeb sequences very hard to distinguish for a human, but easy for the model (see Section 4.2). This might be because the model sees millions of examples from which several subtle features can be picked up.

5 Bias Propagation

From our dataset classification experiments, we observe that each dataset exhibits inherent biases not apparent to humans but detectable by classifiers. We now explore how these biases propagate to text generated by LLMs trained on those datasets.

We consider the following three publicly available LLMs pre-trained on individual datasets from the seven datasets considered in this study:

- Falcon-7B: A 7B parameter model pretrained on 1.5 trillion tokens from the RefinedWeb dataset by TII (Technology Innovation Institute) Almazrouei et al. (2023).
- DCLM-7B: A 7B model pretrained on 2.5 trillion tokens from the DCLM-Baseline dataset by the DCLM team Li et al. (2024).
- FineWeb-Edu-1.8B: A 1.8B parameter model trained by Huggingface on 350 billion tokens from the FineWeb-Edu dataset. The smaller model size and fewer pretraining tokens for FineWeb-Edu are consistent with its focus on educational texts.

All LLMs are only pretrained on the respective datasets, and not finetuned in any way. We generate data with each of the LLMs by prompting the LLM with a single token, sampled from the distribution of tokens that appear as the first token in the sequences derived from original training data of the LLM. We generate 160M training tokens and 8192 test sequences from each LLM.

Original vs generated: By inspecting the generated data, we observe that the outputs of the LLMs resemble the data on which they are trained (see Appendix D for examples). We next measure a classifier's ability to differentiate between the original and generated data.

We train a 160M model on 320M tokens (160M original, 160M generated) for every dataset to distinguish the original from the generated data. The accuracies are as follows: RefinedWeb 89.64%, DCLM-Baseline 89.61%, and FineWeb-Edu 89.84%. This is not surprising, as it is well established that text generated with current LLM can be relatively well distinguished from human-written text if the text is sufficiently long Hans et al. (2024); Tian et al. (2024).

Generated vs generated: We next study how well we can distinguish between the generated data. We train a 160M model on 480M training tokens (160M per generated dataset) for the 3-way classification task of generated RefinedWeb, generated DCLM-Baseline, and generated FineWeb-Edu data.

The classifier achieves an accuracy of 95.59%, indicating that these generated datasets are easily distinguishable, even easier than the original datasets (89.76% as in Table 1). This is likely because the generated data comes from different LLMs, and each LLM introduces its biases in the data it generates.

Classifying generated data with a model trained to distinguish the original data: We next measure to what extent the bias in a dataset can be measured from sequences generated by a language model trained on this dataset. To this end, we measure whether a classifier trained on original data can effectively classify generated data.

Using the 3-way classifier trained on the original RefinedWeb, DCLM-Baseline, and FineWeb-Edu data (as described in Section 4), we classify the generated data. The classifier achieves 89.15% accuracy on the generated data, only 0.61% less than the accuracy on the original data (89.76% as in Table 1).

This indicates that the unique biases and fingerprints inherent in pretraining datasets propagate through training, and can be measured surprisingly well from the outputs of models trained on those datasets.

5.1 Instruction Finetuning

So far our analysis has focused on pretrained LLMs without additional finetuning. We next consider instruction finetuned models, and investigate to what extent finetuning influences the biases present in the models' outputs.

To this end, we utilize Falcon-7B-Instruct, provided by TII, and DCLM-7B-IT, by the DCLM team. Both are instruction-finetuned variants of Falcon-7B and DCLM-7B, respectively. We generate 8192 test sequences from each model.

We proceed with a binary classification task that distinguishes between RefinedWeb and DCLM datasets. We train a 160M model on 320M tokens from the original datasets of RefinedWeb and DCLM (160M tokens from each dataset).

We then evaluate the classifier's performance on three data types: the original datasets, generated data from the pretrained models (without finetuning), and generated data from the instruction-finetuned models. The results are summarized in Table 3.

	Original data	Generated data w/o finetuning	Generated data with finetuning
Accuracy	99.03%	97.39%	89.09%

Table 3: Accuracy of a binary classifier trained on original data, and tested on original data, generated data from a pretrained model, and generated data from an instruction finetuned model.

The results suggest that finetuning a model causes its outputs to diverge from the original data it was pretrained on. However, the inherent biases still persist, enabling the classifier to differentiate between the outputs.

6 Estimate Mixture Proportions

We have so far demonstrated that widely-used LLM pretraining datasets contain biases that make them distinguishable, and that these biases propagate through training. As a result, a classifier trained on the original data can accurately classify data generated by an LLM. Building on these findings, we can to some extent estimate the mixture proportions of pretraining datasets of an LLM, as we discuss next.

LLMs are typically pretrained on a mixture of datasets with certain mixture proportions. These proportions significantly impact model performance and are non-trivial to optimize Xie et al. (2023); Albalak et al. (2023); Ge et al. (2024). Several LLM developers disclose only the datasets used in training, but not the precise mixture proportions, such as GPT-NeoX Black et al. (2022), OPT Zhang et al. (2022), and Galactica Taylor et al. (2022).

We hypothesize that an LLM pretrained on multiple datasets, when prompted with a random token, will generate sequences that closely follow the proportions of its training mixture, since LLMs learn the underlying data distribution during training Deletang et al. (2024), and generate tokens by sampling from the probability distribution of the learned patterns. We therefore expect the model's outputs to statistically follow the original dataset proportions, producing sequences that reflect the frequency of each dataset in the mixture.

To verify this hypothesis, we utilize SlimPajama Shen et al. (2024), which is a further refined and deduplicated version of RedPajama-1T Together Computer (2023b). SlimPajama consists of 7 domains: Arxiv, Books, Github, C4, CC (Common Crawl), SE (Stack Exchange), and Wikipedia. The SlimPajama team provides two 1.3B LLMs trained on 330B tokens from the SlimPajama dataset. The first LLM is trained on only 4 domains: Books, Github, CC, and Wikipedia, and the second one is trained on all 7 domains. The mixture proportion of each domain is known.

We train a 4-way classifier on the original data from the 4 domains: Books, Github, CC, and Wikipedia, and another 7-way classifier on the original data from all 7 domains. We use the 160M model as a classifier, and 160M training tokens from each domain. We generate 2048 random sequences from each LLM by prompting the LLM with one random token, then classify the generated sequences using the the classifiers trained on the original data.

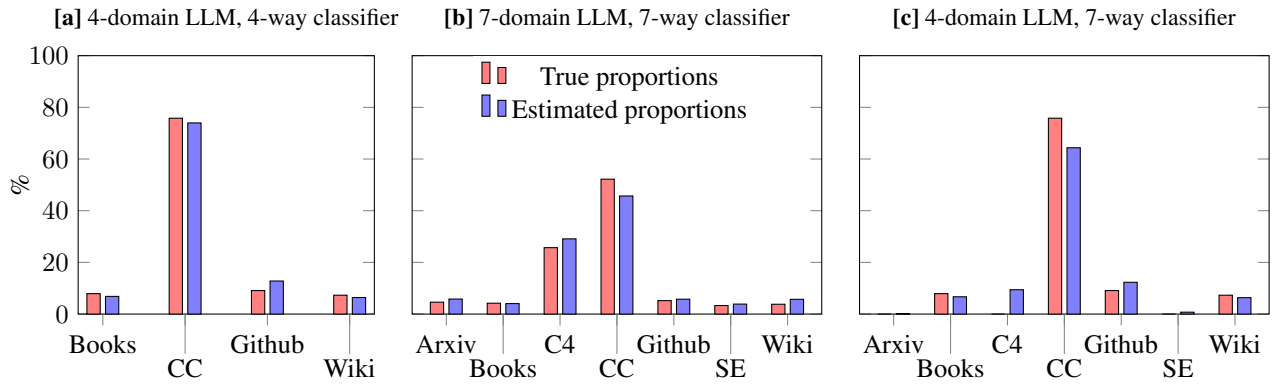

Figure 4: Percentage of generated sequences assigned to different domains by a classifier trained on original data. **[a]** Sequences generated by an LLM trained on 4 domains and classified by a classifier trained on the same 4 domains. **[b]** Same as [a] but 7 domains. **[c]** Sequences generated by an LLM trained on 4 domains and classified by a classifier trained on 7 domains.

We classify the sequences generated by the LLM trained on 4 and 7 domains using the 4-way and 7-way classifiers respectively, and report the percentage of sequences classified as belonging to one of the domains in Figure 4 **[a,b]**. The estimated proportions closely approximate the true proportions across most domains. However, the estimates of C4 and CC deviate from the true ones as seen in **[b]**, as some CC sequences are classified as C4. This is somewhat expected as C4 is a subset of CC.

In Figure 4 **[c]**, we use the 7-way classifier to classify the sequences generated by the LLM trained on the 4 domains, to verify if the classifier correctly refrains from assigning sequences to the 3 excluded domains: Arxiv, C4, and SE. Almost no sequences were classified as Arxiv or SE, confirming that the LLM has not been trained on any of them. As observed previously, a discrepancy appears with some CC sequences misclassified as C4.

7 Conclusion and Limitations

In this paper we demonstrated that popular pretraining text datasets contain inherent biases that propagate through training, enabling a classifier trained on original data to effectively classify generated data and, consequently estimate the pretraining mixture proportions. We showed that classification is possible under various conditions such as rephrasing and finetuning.

However, one case where classification accuracy is degraded is when datasets consist of the same domains but differ solely in their mixture proportions. Consider two perfectly distinguishable dataset domains, $\mathbf{A}$ and $\mathbf{B}$. Two datasets $\mathbf{X}$ and $\mathbf{Y}$ are constructed with different mixtures of $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{X}$ has a higher proportion of $\mathbf{A}$ than $\mathbf{Y}$, and $\mathbf{Y}$ has a higher proportion of $\mathbf{B}$ than $\mathbf{X}$. Sequences from $\mathbf{A}$ in $\mathbf{Y}$ may be misclassified as belonging to $\mathbf{X}$, since $\mathbf{X}$ has seen more sequences from $\mathbf{A}$. Similarly, sequences from $\mathbf{B}$ in $\mathbf{X}$ are likely to be misclassified as originating from $\mathbf{Y}$. This setup highlights how classification becomes unreliable when datasets differ only in domain proportions rather than content or filtering techniques.

Reproducibility Statement

This work is fully reproducible, as all resources and tools used are publicly available. We conduct the classification experiments using the OpenLM repository, which is a public repository designed for research on medium-sized language models (up to 7B parameters). All datasets considered in this study are publicly available on Huggingface. The rewriting experiments are performed with ChatGPT, and require an API key from OpenAI. For bias propagation, we employ pretrained LLMs publicly available on Huggingface. We will release all code, dataset and LLM download links, and reproduction instructions on our GitHub page.

REFERENCES

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *arXiv*, 2312.02406, 2023.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models. *arXiv*, 2311.16867, 2023.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. *arXiv*, 2204.06745, 2022.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. *arXiv*, 2012.07805, 2021.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. *arXiv*, 2102.08981, 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

Gregoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations (ICLR)*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 1810.04805, 2019.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. Bimix: Bivariate data mixing law for language model pretraining. *arXiv*, 2405.14908, 2024.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv*, 2301.07597, 2023.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. *arXiv*, 2401.12070, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *arXiv*, 2009.03300, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models. *arXiv*, 2203.15556, 2022.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. RADAR: Robust AI-Text Detection via Adversarial Learning. *arXiv*, 2307.03838, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv*, 2302.13971, 2023.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv*, 1607.01759, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*, 2015.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldani, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. *arXiv*, 2406.11794, 2024.

Zhuang Liu and Kaiming He. A Decade's Battle on Dataset Bias: Are We There Yet? *arXiv*, 2403.08632, 2024.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *arXiv*, 2406.06443, 2024.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv*, 2311.17035, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv*, 2203.02155, 2022.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv*, 2306.01116, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. *arXiv*, 2406.17557, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):140:5485–140:5551, 2020.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. Slimpajama-dc: Understanding data combinations for llm training. *arXiv*, 2309.10818, 2024.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting Pretraining Data from Large Language Models. *arXiv*, 2310.16789, 2023.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release Strategies and the Social Impacts of Language Models. *arXiv*, 1908.09203, 2019.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani,

Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv*, 2402.00159, 2024.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv*, 2211.09085, 2022.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM*, 59(2):64–73, 2016.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. Multiscale Positive-Unlabeled Detection of AI-Generated Texts. *arXiv*, 2305.18149, 2024.

Together Computer. RedPajama: An Open Dataset for Training Large Language Models, 2023a.

Together Computer. RedPajama: An Open Source Recipe to Reproduce LLaMA training dataset, 2023b.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528, 2011.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. *arXiv*, 1911.00359, 2019.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. *arXiv*, 2305.10429, 2023.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv*, 2205.01068, 2022.

A Model, Training Details, and Hyperparameters

In this section, we outline the architectures of the models employed in our experiments, as well as the training procedures and hyperparameters. For all experiments, we utilize the GPT-NeoX tokenizer Black et al. (2022), which has a vocabulary size of 50,432 tokens.

A.1 Model

Our primary model is a transformer with 160 million parameters (160M). Additionally, we conduct ablation studies using models of varying sizes, including 25M, 87M, and 410M parameters. All models are standard autoregressive transformers, with detailed specifications provided in Table 4.

To adapt the standard transformer architecture for classification tasks, we replace the final layer with a classification head. Specifically, the original last layer, a linear transformation that maps from the context length to the vocabulary size, is substituted with a classification head. This classification head is also a linear layer that maps from the context length to N, where N represents the number of classification classes.

A.2 Training Details

To prepare the training data, we follow the standard procedures for LLM pretraining. We first tokenize the text sequences using the GPT-NeoX tokenizer. We then construct input sequences of length 2048 tokens, corresponding to the model's context length, by appending sequences of the same dataset together.

An $< |\text{endoftext}| >$ token is added at the end of every sequence before concatenating it with the subsequent sequence. The resulting training sequences, each of length 2048, are partitioned into shards. Each shard contains 8192 sequences, resulting in a total of $8192 \times 2049 = 16.78$M tokens per shard.

We train the transformer with a classification head to classify which dataset a text sequence is coming from using the cross-entropy loss. The loss is computed at the token level, where the model classifies every sub-sequence within a

Model	25M	87M	160M	410M
Hidden dimension	192	488	768	1024
Num. heads	12	12	12	16
Num. layers	12	12	12	24
Context length	2048	2048	2048	2048
Vocab. size	50432	50432	50432	50432
MLP ratio	8/3	8/3	8/3	8/3
Activation	SwiGLU	SwiGLU	SwiGLU	SwiGLU
Weight tying	no	no	no	no

Table 4: Model specifications. All models have a similar architecture, and differ only in the hidden dimension, number of heads, and number of layers.

given sequence. For instance, a sequence of length 2048 tokens is seen by the model as a series of sub-sequences of lengths 1, 2, 3, ..., 2047, and 2048. Each sub-sequence is classified individually under the same class as the original sequence, ensuring that the model learns to predict the class consistently across all sub-sequence lengths.

At test time, the text sequences are tokenized and fed into the model in their original form, without concatenation. Consequently, the test sequences vary in length. If a sequence originally exceeds 2048 tokens, the model processes only the first 2048 tokens, as this is its maximum context length. Unlike the training phase, where sub-sequences are classified, the model classifies the entire sequence as a whole during testing.

A.3 Hyperparameters

In all experiments, we train each model for a single epoch, which means that each training token is seen by the model only once. We use a batch size of 16 and apply gradient clipping with a norm of 1 to stabilize training. The initial learning rate is set to 0.0003 and is decayed to zero using a cosine annealing scheduler, with a warm-up phase of 2000 steps.

The optimizer used is AdamW Kingma & Ba (2015) with hyperpameters: $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1 \times 10^{-8}$, and weight decay 0.2. We also use automatic mixed precision training with brain floating point 16 (bfloat16) to enhance computational efficiency throughout the training process.

B 2-way Classification

Our main results displayed in table 1 were for 3-, 4-, and 5-way classification between the main datasets considered in this study. In this section, we report the classification accuracies for all possible binary combinations between the seven datasets, i.e., $\binom{7}{2} = 21$ possible combinations.

As before, we use the 160M model with 160M training tokens and 8192 test sequences per dataset. Results are reported in Table 5.

	C4	FineWeb	RefinedWeb	DolmaCC	RedPajama-V2	DCLM	FineWeb-Edu
C4		87.37%	90.72%	69.42%	95.64%	98.85%	92.88%
FineWeb	87.37%		75.49%	82.70%	80.54%	99.15%	78.05%
RefinedWeb	90.72%	75.49%		88.32%	80.68%	99.03%	84.74%
DolmaCC	69.42%	82.70%	88.32%		90.91%	97.03%	91.08%
RedPajama-V2	95.64%	80.54%	80.68%	90.91%		99.05%	77.69%
DCLM	98.85%	99.15%	99.03%	97.03%	99.05%		98.54%
FineWeb-Edu	92.88%	78.05%	84.74%	91.08%	77.69%	98.54%	

Table 5: Classification accuracy for all possible 2-way combinations of the seven main datasets in this study.

C Further Ablation Studies

Accuracy vs Sequence Length: As seen in Appendix E, the sequence length varies a lot between the datasets, and even within the same dataset. To evaluate the impact of sequence length on classification accuracy, we analyze

sequences of lengths ranging from 0 to 2000 tokens, and divide them into intervals of 200 tokens (i.e., 0-200, 200-400, ..., 1800-2000). For each interval, we sample 1024 test sequences from each dataset.

The results, illustrated in Figure 5, show a steady improvement in classification accuracy as sequence length increases. This trend aligns with the expectation that longer sequences contain more information (2000 tokens is around 1500 words), which allows the classifier to identify more distinguishable patterns and improve classification performance. However, even short sequences can perhaps surprisingly be well classified.

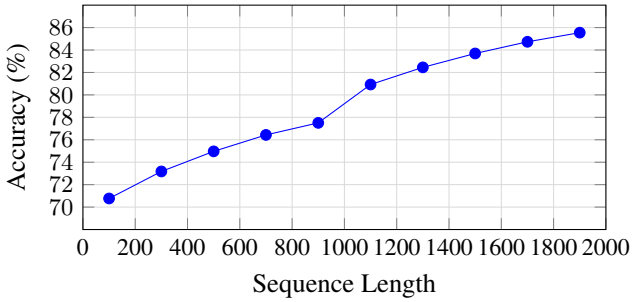

Figure 5: Accuracy vs sequence length. Longer sequences attain higher accuracies than shorter ones.

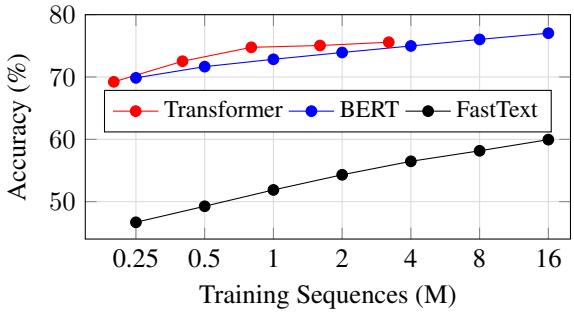

Figure 6: Classification accuracy of BERT and FastText classifier compared to an autoregressive transformer.

BERT: Unlike autoregressive transformers, BERT Devlin et al. (2019) is a bidirectional transformer model that captures contextual information from both preceding and succeeding tokens within a sequence, without the use of causal masks that limit attention to preceding tokens. As a result, BERT processes the entire sequence at once during training, rather than treating it as a series of subsequences. We therefore plot its performance as a function of the number of training sequences in Figure 6.

For reference, we also plot the performance of the autoregressive transformer relative to the training sequences instead of the training tokens (as in Figure 1 right panel). To obtain the number of sequences, we divide the number of tokens by the average sequence length of C4, FineWeb, and RefinedWeb (see Table 6).

The performance of BERT and the autoregressive transformer are closely comparable. BERT initially achieves slightly lower accuracy but eventually reaches a marginally higher accuracy. The behavior that BERT requires more training sequences is expected, as the autoregressive transformer processes each sequence as a series of subsequences, while BERT processes each sequence only once.

FastText Classifier: FastText Joulin et al. (2016) is an efficient text classification library designed to provide fast and scalable text classification tasks, particularly suitable for classification of large-scale datasets. FastText relies on a simple shallow neural network architecture that enables rapid training and inference. Similar to BERT, FastText processes each sequence as a whole.

We display FastText's performance with respect to the number of training sequences in Figure 6. The transformer-based classifier and BERT significantly outperform FastText, but are significantly slower, and require significantly more compute.

Majority Vote at Test Time: Unlike the training phase, where sub-sequences are classified, we classify the entire sequence as a whole during testing throughout the paper. In this ablation study, we classify all subsequences within one test sequence, and then determine the final prediction as the majority vote. For instance, a sequence of length $\mathtt{x}$ tokens will yield $\mathtt{x}$ predictions. The final predicted class is the most frequent among these individual predictions. Using majority voting reduces accuracy to 67.37%, which is a 7.39% decrease compared to the default whole sequence classification.

Aggregating sequences: Throughout the paper, we classify individual sequences. In this ablation study we combine the sequences of the same dataset together to form sequences of length 2048 tokens, aligning with the context length of our transformer model. This creates a uniform test set with sequences of equal length, were each sequence utilizes the entire attention span of the transformer.

The aggregation of sequences yields an impressive 95.18% classification accuracy, approximately 10% higher than the default sequence based testing with sequences of length 1800-2000 tokens as seen in Figure 5. This suggests that providing the classifier with multiple concatenated sequences simplifies the classification task, making it easier than classifying a single sequence of similar combined length.

Linear probing: Linear probing refers to training a linear classifier on fixed pretrained representations. It is often used as a simple evaluation metric as it offers a quick assessment of how well a pretrained model can classify data using only a linear classifier. We freeze the weights of the pretrained model, and train only the last linear layer, i.e., the classification head, which results in 33.18 %, equivalent to a random guess.

D SAMPLE TEXTS

In this section we show an original sequence paraphrased with the 3 prompts mentioned in section 4.3. We also provide example sequences from the datasets considered in this study. We display original sequences from C4, FineWeb, RefinedWeb, DolmaCC, RedPajama-V2, DCLM-Baseline, and FineWeb-Edu, and the generated sequences from the three pretrained LLMs: Falcon-7B, DCLM-7B, and FineWeb-Edu ablation model.

For clarity and ease of visualization, only short sequences are displayed here. Some sequences are considerably longer and span multiple pages. Therefore, the sequences shown here do not reflect the actual sequence length.

D.1 REPHRASED SEQUENCES

Original Text

What is the length of the coordinates in the two inertial systems S and S' at the velocity v, according to relativity - if the length of the coordinates are identical at v = zero?
1) Is the physical length of the coordinates greatest in S?
2) Is the physical length of the coordinates in S equal to the physical length of the coordinates in S'?
If the physical lengths are identical, will it then be possible to have any physical length contractions according to relativity?

Text rephrased with Prompt 1

What is the length of the coordinates in the two inertial systems S and S' moving at velocity v, given that the lengths of the coordinates are the same when v equals zero?
1) Is the physical length of the coordinates maximized in S?
2) Is the physical length of the coordinates in S the same as the physical length of the coordinates in S'?
If the physical lengths are the same, is it then feasible to have any physical length contractions as described by relativity?

Text rephrased with Prompt 2

What is the length of the coordinates in the two inertial reference frames, S and S', moving at velocity v, in accordance with relativity, if the coordinate lengths remain the same when v = 0?

1) Is the physical length of the coordinates longest in S?
2) Are the physical lengths of the coordinates in S and S' equal?

If the physical lengths are equal, can length contractions according to relativity even occur?

Text rephrased with Prompt 3

What is the coordinate length in the two inertial systems S and S' at velocity v, given that the lengths are the same when v equals zero? 1) Is the physical length of the coordinates larger in S? 2) Is the physical length of the coordinates in S equivalent to that of the coordinates in S'? If the physical lengths are the same, can there be any length contractions as described by relativity?

Figure 7: A sequence from C4 rephrased with the 3 different prompts described in section 4.3. The rephrased text shows increasing divergence in structure and format from the original one.

D.2 ORIGINAL SEQUENCES

C4

• Beginners BBQ Class Taking Place in Missoula!
Do you want to get better at making delicious BBQ? You will have the opportunity, put this on your calendar now. Thursday, September 22nd join World Class BBQ Champion, Tony Balay from Lonestar Smoke Rangers. He will be teaching a beginner level class for everyone who wants to get better with their culinary skills.
He will teach you everything you need to know to compete in a KCBS BBQ competition, including techniques, recipes, timelines, meat selection and trimming, plus smoker and fire information.
The cost to be in the class is $35 per person, and for spectators it is free. Included in the cost will be either a t-shirt or apron and you will be tasting samples of each meat that is prepared.

• Hurrah! A cooperative worldwide effort to rescue Thailand children trapped in a flooded cave rescued them all in less than 3 weeks from the time they entered the cave to the time of their rescue.
It should be much easier, shouldn't even take a heroic effort, to rescue children trapped in separation from their families at the Mexican border. These things are possible, but this week, the administration did not even meet the first deadline to get all the children below 5 years old reunited with their families.
It should even be logistically possible with a cooperative world wide effort to develop economic systems that could rescue all the hungry children everywhere living in poverty.
In the U.S. alone, 1 in 5 children live in poverty, according to a recently released United Nations report.

FineWeb

• Originally Posted by bradhs
The only thing you can do is create a Search and Save it with a shortcut key. I do this when I only want to see my corporate email.
1. Go into your Messages and select Search.
2. Set the Service option to the Enterprise Email.
3. Save the Search. Give it a name and a shortcut key.
Use the shortcut key to restrict the email list to only Enterprise email.
Hm. This isn't working for me... When I initiate the search, it comes back with no messages, and I do have some that it should show... The service option choices are: All Services, my pop email address, and Desktop. I selected Desktop. That right?

• I have just updated my TV and Blu ray player but not my amp.
I didn't want to update my Sony STR-DG820 amp because it works so well, but I did want to keep my options open for playing 3D discs so I got the Panasonic DMP-BDT310 because it has two HDMI ports and could route sound through the amp and picture to the TV.
I've gone through every setup and I'm not getting DTS-HD or TRUE-HD. The manual doesn't help at all and this is becoming a little silly. Could some one go through a step by step guide in the setup.

RefinedWeb

• A huge thank you goes to those who helped with the hedge cutting on the road side of the churchyard recently. This was a very much needed task, the pathway is nice and clear now. We are also very grateful to whoever donated the funds to provide the skip, again this was a much needed requirement.
If anyone is interested in helping to maintain the churchyard, please contact Mr Mike McCrea on 01283 214473. Any assistance will be gratefully received.

• Free US shipping on orders over $50!
Pumpkin dominates the fall fragrance scene! This best seller combines brown sugar, molasses, vanilla, and classic holiday baking spices to make an aroma that is simply irresistible!
—-!
Amy's review:
"I love these candles. So clever that they're in a coconut shell! The scents fill my house and they have a long burn time! I've purchased from them twice and will continue to support this business! Can't wait to go home and try my fall scents!"

DolmaCC

• Wowed by the lights and prospects of city life, Loveness leaves her small mining town in search of a new life in Harare. She imagines herself falling for a hot-shot city man becoming his wife and spending her life in luxury while tending to her city children. The man she considers the love of her life is anything but a hot shot, and he is abusive and uncaring. To top all this off, he his HIV positive. Loveness is at a crossroads. She must consider her choices.
Although, Waste Not Your Tears does not shy away from misfortune, it is also a novel of forgiveness and hope. Loveness is an unlikely heroine on a stage set during the crisis of HIV/AIDS in Zimbabwe. She lives, however, amongst us, and reading this sensitive and thoughtful novel provides insights into the challenges of making the wrong choices, but having the strength to move forward.

• The Avon Lake Sports Hall of Fame's purpose is to give lasting recognition to the outstanding sports figures and/or teams of Avon Lake who have demonstrated outstanding athletic ability at the high school, college, amateur or professional sports levels.
We strive to recognize those individuals who have contributed greatly to the promotion of sports through leadership, sponsoring, coaching or providing assistance to athletes of athletic programs.
It is our utmost desire to promote more interest in the athletic programs of Avon Lake.

DCLM-Baseline

• Economic Indicators for Libertarians 101

Why Ron Paul is Unique? (Galvanizers and Diplomats)

Ron Paul is a unique figure in libertarianism, able to not only be a diplomat and figure that people outside of libertarianism can empathize with, but also a diehard who can galvanize the most radical of libertarians. It's very rare a figure like him can exist, and let's be glad he does.

• Tuesday, 26 April 2011

tea parties, wonderland, high tea, garden party

I want to hold a cute girly tea party and everyone has to wear their sunday best. I just love the idea. of pretty pastel colours, cupcakes, cooking for your girls and everyone looking pretty

1. This is a beautiful Idea, Ive always wanted to host a tea party and these photos have inspired me to actually go through with it.

2. Beautiful! Could you tell me where you got your cart from? I'm trying to create something similar and they're deceivingly hard to find...

Thankyou for commenting! x

RedPajama-V2

• Updaty posty thingy
Sooo....
Chainmaille was a disaster. I need someone to show me how to construct. It was moot anyway, as I had an anxiety attack and barely made it in to the con. I am so embarased, but glad it wasn't a long term thing.
I am still working on getting the chaim maille done. Maybe it will look fine. I don't know. I also need to work on the scale spoon maille.
Right now, though, my main focus is finding a job. I thought I had extended unemployment until I was done wtih school. Turns out that was not entirely true, and now I am kind of up a creek. I have been saving money, so right now I have been paying bills with my savings. However that is also about to run out. I have applied for abawd.

• Brooklyn Man Who Stabbed 75-Year-Old Woman and Left Her for Dead Sentenced to 75 Years in Prison
Brooklyn Man Who Stabbed 75-Year-Old Woman and
Left Her for Dead Sentenced to 75 Years in Prison
Defendant, a Friend of the Victim's Grandson, Forced His Way into Apartment
Brooklyn District Attorney Ken Thompson today announced that a Brownsville man has been sentenced to 75 years in prison following his conviction on second-degree attempted murder and other charges for stabbing an elderly woman repeatedly and leaving her seriously injured on her apartment floor.
District Attorney Thompson said, "This defendant savagely stabbed a defenseless 75-year-old woman all over her body, robbed her of what little money she had and then left her to die. He deserves every day of his 75-year prison sentence."

FineWeb-Edu

• A "magic" herb, Carissa Edulis, that drew thousands of people to a remote Loliondo village in Tanzania was identified by Kenyan scientists a few years ago as a cure for a drug-resistant strain of a sexually transmitted disease, gonorrhoea. This herb also is believed to cure many other diseases besides gonorrhoea. The Kamba refer to as mukawa or mutote and use it for chest pains, while the Nandi boil the leaves and bark to treat breast cancer, headache and chest pains.
Researchers discovered the plant could be used for the treatment of the herpes virus. Led by Dr Festus M Tolo of the Kenya Medical Research Institute (Kemri), the team from the University of Nairobi and the National Museums of Kenya found the herb could provide an alternative remedy for herpes infections.
"An extract preparation from the roots of Carissa edulis, a medicinal plant locally growing in Kenya, has exhibited remarkable anti-herpes virus activity for both wild type and drug resistant strains," they reported in the Journal of Ethnopharmacology.

• Dinosaurs' active lifestyles suggest they were warm-blooded
H. Pontzer, V. Allen, J.R. Hutchinson/PLoS ONE
Whether dinosaurs were warm-blooded or cold-blooded has been a long-standing question in paleobiology. Now, new research on how two-legged dinosaurs walked and ran adds new evidence to the argument for warm-bloodedness, and suggests that even the earliest dinosaurs may have been warm-blooded.
Warm-blooded (or endothermic) dinosaurs — able to regulate their own body temperatures — would have been more active and could have inhabited colder climates than cold-blooded (or ectothermic) dinos, which would have functioned more like modern reptiles — animals that become animated only as temperatures warm. Endothermic dinosaurs would have also required more energy to maintain their higher metabolic rates.

D.3 GENERATED SEQUENCES

DCLM-7B

• Exercise 1: I've been trying to do this exercise for a couple of weeks, and have failed each time. I was able to get the hang of it on this week's attempt, but still couldn't finish before the timer ran out.

Here is what I did: 1) Find the equation for the line that goes through (1,0) and (2,4). This turned out to be y=2x-1.

2) Set up a table for my function g(x) = 2x-1

3) Plot g(x)

4) Put both g(x) and f(x) in a scatter plot

5) Solve for x so that y=0. That means there will be no intersection when the function is less than 0. For my function, x=-1/2, which was easy enough. For my line, since y=2x-1, x=y+1/2

The result looks something like this (if it displays correctly):

•Aluminium chloride is not a compound that you would normally make in the lab, but it can be made by reacting aluminium with chlorine or hydrochloric acid. It reacts violently with water and so if you do get any on your hands, wash them thoroughly under a tap. The reaction of these two elements produces hydrogen gas and sodium chloride as products. Aluminum Chloride Formula - AlCl 3; Molecular Weight: 133.34 g/mol; Structural Formula Drawing (BF4)3-Lewis Dot Structure for BF4- Boron trifluoride has three pairs of non-bonding electrons around it.

The other product is hydrogen chloride, HCl(g). When aluminum metal is exposed to air, it forms a thin protective layer of alumina, which protects the metal from further corrosion. In this experiment, an aqueous solution of aluminum chloride will react with potassium carbonate to produce solid aluminum hydroxide and aqueous potassium chloride.

FineWeb-Edu ablation model

• The number of individuals with this condition increased in the United States between the years 2000 and 2010, according to the National Institute of Neurological Disorders and Stroke.
The most common causes of acquired cerebral arteriovenous malformations include:
- Injury to blood vessels or arteries
- High blood pressure
- Brain tumors
Acquired cerebral arteriovenous malformations are usually congenital, meaning that they were present at birth. The exact cause of these malformations is unknown.
In some cases, the cause of an acquired cerebral arteriovenous malformation is known. This can occur when a specific event (such as bleeding) occurred near the malformation.

• A series of experiments was conducted at the University of Texas, Austin to determine the effects of an electromagnetic field on the rate and amount of protein binding between DNA and a transcription factor. The study found that when DNA was placed in an external magnetic field, the rate of DNA binding by the transcription factor increased threefold. When DNA and a transcription factor are bound together, it is often referred to as a complex.
In this study, the researchers found that the rate of protein binding could be enhanced up to five times in some instances by exposing the DNA to a magnetic field. This finding has been linked to the possibility that the magnetic field could act as a trigger for the transcription factors to bind together more easily.

Falcon-7B

• I have sold a property at 2001 208A ST in Langley.
Welcome to this well maintained rancher located in one of Langley's most desirable areas. This home has a large open concept kitchen with a breakfast nook, a formal dining room, a gas fireplace in the living room and a large master bedroom with a 4 piece ensuite. The home has had numerous upgrades over the years including the roof, windows, furnace, hot water tank and newer appliances. Enjoy the spacious covered patio, fully fenced back yard and a single garage with additional parking in the driveway. Close to schools, shopping, restaurants and all other amenities. This is a great home in a great neighbourhood! Call today for a viewing!

• Deploying 3D Models in the Web
The Web has long been a source for sharing 2D content, from photos and text to music and video. But sharing 3D content on the Web is not so common, due to bandwidth limitations, browser support and more.
However, 3D models can be used in the Web in various ways: as 3D printing instructions, as part of Augmented Reality or Virtual Reality applications, as part of an interactive online exhibit, as part of a video game, and more.
We present the most common ways to share 3D models on the Web and describe how to deploy models using Web technologies, such as WebGL and 3D CSS.

E Dataset Statistics

The datasets we consider in this paper are comprised of millions to billions of sequences with varying lengths. In this section, we present a statistical analysis on the sequence lengths of the seven datasets. To obtain representative statistics, we randomly sample 100,000 sequences from each dataset and tokenize them with the GPT-NeoX tokenizer. The statistics of the lengths of the tokenized sequences are summarized in Table 6 and histograms are displayed in Figure 8.

Dataset	Mean	St. Deviation	Mode	Median	Range
C4	477	823	58	253	31188
FineWeb	700	1540	129	410	118422
RefinedWeb	624	1549	82	314	137104
DolmaCC	825	1647	96	451	132310
RedPajama-V2	1137	3191	12	603	274814
DCLM-Baseline	1235	2600	101	665	153768
FineWeb-Edu	1059	1993	261	597	120240

Table 6: Statistics of the sequence lengths (in number of tokens) of the seven main datasets considered in this paper.

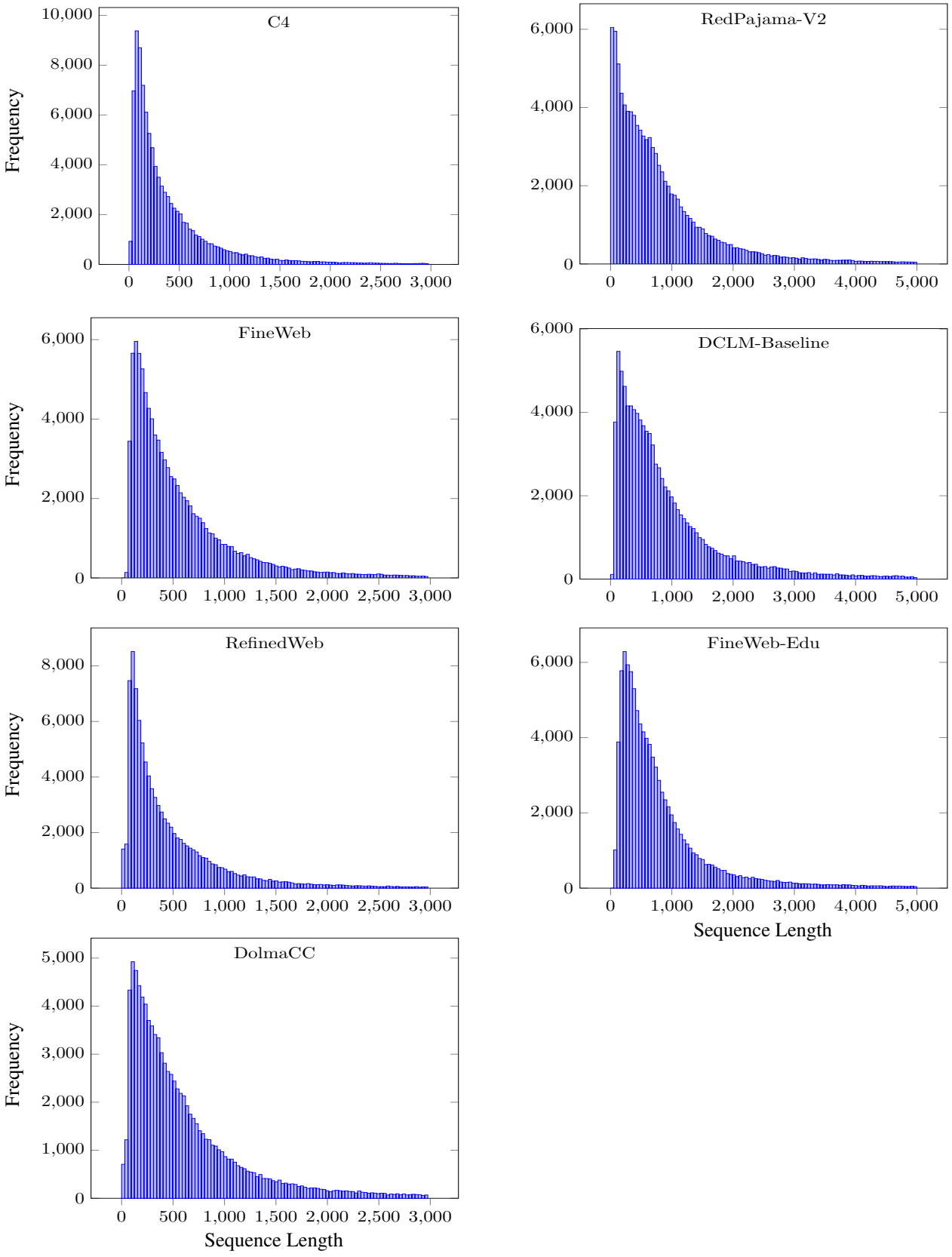

Figure 8: Histograms of the sequence lengths of the main datasets considered. Lengths exceeding 3000 and 5000 tokens are omitted for ease of visualization.