Data, Data Everywhere: A Guide for Pretraining Dataset Construction

Anonymous ACL submission

Abstract

The impressive capabilities of recent language models can be largely attributed to the multi-trillion token pretraining datasets that they are trained on. However, model developers fail to disclose their construction methodology which has lead to a lack of open information on how to develop effective pretraining sets. To address this issue, we perform the first systematic study across the entire pipeline of pretraining set construction. First, we run ablations on existing techniques for pretraining set development to identify which methods translate to the largest gains in model accuracy on downstream evaluations. Then, we categorize the most widely used data source, web crawl snapshots, across the attributes of toxicity, quality, type of speech, and domain. Finally, we show how such attribute information can be used to further refine and improve the quality of a pretraining set. These findings constitute an actionable set of steps that practitioners can use to develop high quality pretraining sets.

1 Introduction

Recent language models (LMs) (OpenAI, 2024; Team, 2024b,a; Anthropic, 2024; Team et al., 2024) have shown very strong capabilities on a number of evaluation areas. In comparison to previously developed LMs (Brown et al., 2020; Radford et al., 2019; Smith et al., 2022a; Rae et al., 2022; BigScience, 2023), these newly released models generally follow the same architectural details, based on the transformer (Vaswani et al., 2017). Rather, with emphasis being placed on the size and quality of the pretraining dataset (Hoffmann et al., 2022; Longpre et al., 2023), the improved capabilities of LMs are largely due to self-supervised pretraining on ever larger, higher quality datasets. It is clear that the pretraining set is crucial to model success, but the question on how to effectively create one has yet to be openly answered.

Most leading models (OpenAI, 2024; Team, 2024b; Anthropic, 2024; Jiang et al., 2023) do not divulge what methods were used to go from raw data sources to a final pre-training set. Other models document only small sections of their process (Touvron et al., 2023b; Parmar et al., 2024; Bai et al., 2023; Team et al., 2024) and lack information on why or how the chosen decisions were made. The scarcity of open knowledge in this area hinders the general community from contributing to the advancement of model capabilities (Rogers, 2021).

The steps in pretraining set construction are shown in Figure 1: the pipeline starts with a collection of text data sources, removes ill-formed and duplicate documents during data curation, further filters out low-quality documents via data selection, and finally assigns sampling weights to determine the prevalence of each data source during training. Recent works (Longpre et al., 2023; Penedo et al., 2023; Soldaini et al., 2024; Penedo et al., 2024) have started to elucidate strategies for effective pretraining set development. However, they all focus solely on the step of data curation and analyze only a small number of mostly English sources.

In this paper, we provide insights across all steps of pretraining set development for a set of over 2T tokens composed of English, multilingual, and source code documents. We compare existing methods through a series of ablations at each step of the development pipeline in Figure 1 to quantify which techniques do and do not realize improvements in downstream evaluations. For the best identified method, we highlight various design decisions that impact performance.

Additionally, previous studies on web crawl are conducted across a small number of snapshots and are limited to the characteristics of toxicity and quality (Longpre et al., 2023). Despite web crawl documents constituting the majority of examples in pretraining sets (Almazrouei et al., 2023; Smith et al., 2022a; Gemma Team, 2024), we still do

Figure 1: Each step in the development process to go from a collection of data sources into a final pretraining set that produces a highly capable LM.

not thoroughly understand their composition. We close this gap by conducting a large-scale analysis on over 90 Common Crawl web snapshots for the attributes of domain, quality, toxicity, and type of speech. We then show how such data attributes can aid in pretraining set construction to further improve model capabilities.

By sharing this information, we provide an actionable series of steps that can be used to construct highly performant pretraining sets. Concretely, our contributions are as follows:

- Suggest a set of techniques to use for the data curation, selection, and sampling steps of pretraining set development for English, multilingual, and code data.
- Perform the first large-scale analysis of web crawled data across the attributes of quality, toxicity, type of speech, and domain.
- Demonstrate that attribute information can be used to enhance the performance of data sampling and data selection methods.

2 Methodology

We ablate a singular part of the development pipeline and train a LM on the resulting pretraining set to understand how various methods affect performance on downstream benchmarks. Our experimental setup is detailed below.

2.1 Data Sources

With current language models being trained on a wide range of data sources, an appropriate study on pretraining set construction must use a large, diverse set of data. Table 1 highlights the sources, along with the amount of tokens coming from each, included in the English, multilingual, and code data that we use in our experiments.

Experimenting on this broad set of data ensures our findings will be applicable in the development of large-scale pretraining sets. As current language models do not just pretrain on English-only data, we highlight the importance of including multilingual and code data within our study. However, while we run ablations for these domains, the majority of our experiments focus on the English set.

Data type	Data source	Tokens (B)
English	Web crawl	889
	Misc	109
	News	94
	Conversational	59
	Books	35
	Scientific	33
Multilingual	Web crawl	540
	Parallel corpora	56
Source Code	The Stack v1.2	212

Table 1: The data sources that are used in our ablation studies. Table 11, Table 12, Table 13, and Table 14 provide a more detailed breakdown of the English, multilingual, and source code datasets.

2.2 Data Curation

As dataset curation has been widely investigated, we do not run ablations to identify which specific techniques are beneficial, but rather compare the benefit when using these studied techniques versus not. We consider three phases of data curation: raw text, post deduplication, and post quality filtering. Our deduplication process is comprised of both exact deduplication where we compute a 128-bit hash for each document, group the documents by their hashes, and select one document per group in addition to fuzzy deduplication as described in (Smith et al., 2022b). In quality filtering, the deduplicated documents are filtered based on the perplexity of a KenLM model (Heafield, 2011) that was trained on a collection of high quality sources alongside a set of heuristic filters as described in (Rae et al., 2021; Raffel et al., 2020). Full details on the quality filter-

ing steps are shared in Table 15. When curating the source code datasets we formed repository-level contexts and filtered out low-quality documents by following the approach of (Li et al., 2023), which is outlined in Table 16.

2.3 Data Selection

In addition to filtering done during data curation, specialized methods have been developed for data selection (Albalak et al., 2024) to ensure that only the highest quality documents make it into pretraining corpora. Amongst the potential methods, we specifically investigate and run ablations with Domain Selection via Importance Resampling (DSIR) (Xie et al., 2023b) as it requires minimal compute overhead and is part of the set of techniques that stem from Moore-Lewis selection (Moore and Lewis, 2010), which accounts for most data selection methods. DSIR takes as input a raw dataset, along with a target dataset of known high quality examples, and then uses importance resampling to select examples from the raw dataset that are distributed like the target by utilizing a bag of hashed n-gram models to match the n-gram frequencies of the selected data and the target.

2.4 Data Sampling

During the construction of pretraining corpora, data weights $\{a_k\}_{k=1}^{N}$ such that $\sum_{k=1}^{N} a_k = 1$ are assigned to each of the N data sources to determine the sampling frequency of each source during pretraining. The value of data weights can greatly impact downstream accuracy as increasing the proportion of data from a given source decreases the cumulative weight on the others, potentially causing degradation on the domains that are now less represented. Specialized methods have been developed to identify appropriate sampling weights that endow the trained model with strong capabilities across a wide range of domains.

In our ablations, we consider two data sampling methods that use heuristics based on characteristics of the data sources to define weight distributions: alpha sampling (Arivazhagan et al., 2019; Shliazhko et al., 2022) and UniMax sampling (Chung et al., 2023), in addition to DoReMi (Xie et al., 2023a) which uses a learned model to identify the sampling weights. Both alpha and UniMax sampling use the number of tokens in each data source to define data weights. Alpha sampling proportionally weights data sources to a scaled factor, α, of their token counts while UniMax fits a uniform weight distribution subject to the constraint that no data sources sees more than a certain number of epochs at the given training token budget. Comparatively, DoReMi defines data weights by formulating the problem via group distributionally robust optimization (Sagawa et al., 2020) and minimizing the excess loss between a small proxy model and a pretrained reference model.

2.5 Data Attributes

We investigate attributes along the axes of toxicity, quality, domain, and type of speech for each document that comes from CC snapshots. Information from quality and toxicity labels can be used to categorize the potential utility of a given document while domain and type of speech labels characterize the types of documents that compose our pretraining set. We obtain these attribute labels by training a DeBERTaV3 (Liu et al., 2019) classifier on a small set of ground-truth labeled web crawled documents before obtaining predictions from each across our entire pretraining corpus. A full breakdown of the labels that each classifier outputs along with a more detailed description of the classifier training procedure can be found in Appendix B.

2.6 Evaluation

In experiments on English datasets, we use the LM-Evaluation Harness (Gao, 2021) to evaluate zero-shot accuracy on PIQA (Bisk et al., 2020), ARC-easy (Clark et al., 2018), Winogrande (Sakaguchi et al., 2020), Hellaswag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), and Race-H (Lai et al., 2017). We also evaluate on MMLU (Hendrycks et al., 2020) when the experimental setting allows for a non-random score. For our source code experiments, we evaluate on HumanEval and MultiPL-E (Chen et al., 2021; Cassano et al., 2023). In our multilingual experiments, we evaluate on the reasoning task of XCOPA (Ponti et al., 2020) and the question answering task of TyDiQA-GoldP (Clark et al., 2020).

2.7 Model Specifications

To ensure that our results hold at various model scales, in our experiments we use either 2B or 8B decoder only transformer LMs trained with autoregressive language modeling at token horizons from 150B to 450B tokens. The configuration used for a given experiment is specified ahead of each reported result. Specifics on model architecture and hyperparameters are shared in Appendix C.

3 Data Curation Ablations

Findings

- Compared to raw text, deduplicated and quality filtered data improve model accuracy.
- In deduplication, it is better to priortize keeping samples from older sources than more recent ones.

All our data curation experiments use a 2B parameter model trained for 300B tokens. Table 2 shows that model accuracy improves after both deduplication and quality filtering, indicating the utility of effective data curation. The impact of data curation for code is shared in Appendix D.

Experiment	LM-Eval
Raw text	57.18
Post deduplication	58.93
Post quality filtering	**59.50**

Table 2: Impact of data curation steps on model accuracy. Per-task accuracies are shared in Table 18.

In fuzzy deduplication, it is possible to prioritize retaining documents from certain sources. As document age has been shown to impact model accuracies (Longpre et al., 2023), we run ablations with the following priortization of data sources: most recent to oldest, oldest to most recent, or at random. Table 3 indicates that prioritizing older documents leads to significantly better results.

Experiment	LM-Eval
Random	59.96
Recent-to-Old	58.93
Old-to-Recent	**60.47**

Table 3: The priortization of data sources in deduplication affects model accuracy. Per-task accuracies are shared in Table 19.

4 Data Selection Ablations

Findings

- DSIR improves the quality of web crawl snapshots.
- DSIR functions best when applied across each data source individually.
- DSIR is fairly sensitive to the composition of the target distribution.

We assess whether DSIR provides gains when used on data that has passed through a data curation pipeline. Through our ablations, we seek to answer: 1) how does naive application with the recommended settings of DSIR perform and 2) can we identify better settings for DSIR. In tackling question 2, we ablate whether selection should be done at the level of individual data sources instead of the entire pretraining corpus and altering the suggested percentage of data that should be selected. All our DSIR experiments train a 2B parameter model for 165B tokens on a training set of two CC snapshots.

Question	Experiment	LM-Eval
Q1	Original CC	54.30
	DSIR	**54.44**
Q2.1	Corpus DSIR	54.44
	Source DSIR	**54.71**
Q2.2	DSIR (80%)	54.55
	DSIR (87.5%)	54.25
	DSIR (95%)	**54.71**

Table 4: DSIR improves the quality of web crawl data. () refers to the percentage of examples that are selected by DSIR. Per-task accuracies are shared in Table 20.

As shown in Table 4, the naive application of DSIR leads to a slight improvement in accuracy compared to post curation CC data, 54.48 vs 54.30. We find that selecting at the level of individual sources improves upon the paper-recommended setting of selection across the entire corpus. The recommended 95% selection rate is optimal.

We ran an additional ablation to understand the sensitivity in performance of DSIR when the target set is altered. Table 5 illustrates that even small alterations to the target set, such as the addition of a high quality source like arXiv, causes fluctuations in model accuracy – indicating that the target set should be defined carefully.

Target Set	LM-Eval
Wikipedia, Books	**54.71**
Wikipedia, Books, arXiv, NIH	54.02
arXiv, NIH	53.71

Table 5: DSIR is impacted by target set composition. Per-task accuracies are shared in Table 21.

5 Data Sampling Ablations

Findings

- UniMax provides the best sampling weights for the English and multilingual domains.
- Alpha sampling, with a value of $\alpha = 1.3$, provides the best sampling weights for the code domain.
- DoReMi is unable to produce competitive sampling weights for any domain as it often gives the majority of the weight to a single source.

In our data sampling ablations, we study the domains of English, multilingual, and code individually as the inherent characteristics of each domain would likely change which data sampling method would be best suited for it. We use an 8B parameter model for the ablations and train on 150B tokens for the code domain and 300B tokens for the English and multilingual domains.

5.1 English

In our English ablations, we replace alpha sampling with preference based weighting, where the weights are hand tuned according to intuitive perceptions of quality, as it has been the most widely used sampling technique for English data (Touvron et al., 2023a; Gao et al., 2020a). With the weights returned by Unimax being dependent upon the number of epochs allowed for each data source, we additionally ablate across across varying values of this hyperparameter. The returned sampling weights and further details on each method can be found in Appendix E.

Method	LM-Eval	MMLU
Preference	65.85	27.20
UniMax (1e)	**67.14**	**28.30**
UniMax (2e)	66.50	28.00
UniMax (4e)	66.61	26.60
DoReMi	65.63	26.90

Table 6: UniMax sampling weights provide the best performance on English data. Ne means that UniMax can use a maximum of N epochs per dataset. Per-task accuracies are shared in in Table 22.

Table 6 shows that UniMax achieves substantially better accuracies on LM-Eval and MMLU compared to the next best method. We note that DoReMi attains the worst performance, which we believe to be a factor of its weight distribution being heavily skewed towards web crawl snapshots as detailed in Appendix E. Additionally, despite still outperforming both other methods, the performance of UniMax degrades as the maximum epoch hyperparameter increases. We hypothesize that as we have far more data tokens than the amount of training tokens, repeated epochs of data provide less utility than novel information. We suggest that practitioners choose the minimal value of the epoch hyperparameter that makes sense for their datasets and training budget.

5.2 Multilingual

It has been shown that models trained on a subset of multilingual languages from a given language family are able to transfer knowledge and capabilities to other languages in the family (K et al., 2020; Hu et al., 2020; Ye et al., 2023). This indicates that a sampling method which more evenly spreads weight so that all language families are well represented, like UniMax, should achieve better accuracy than one which places most of the weight on high resource languages, like alpha sampling. Table 7 confirms this intuition as UniMax slightly outperforms alpha sampling. As with the English ablations, DoReMi's returned weight distribution is heavily skewed, causing it to underperform both other methods. The sampling weights identified by each method are detailed in Appendix E.

Method	XCOPA	TyDiQA-GoldP
Alpha (α = 1.3)	58.11	17.86
UniMax (1e)	**58.24**	**18.11**
DoReMi	57.65	15.8

Table 7: UniMax slightly outperforms alpha sampling on multilingual data.

5.3 Code

We do not use the returned DoReMi sampling distribution in our code ablations as it placed over 80% of the weight on a single programming language, which does not allocate enough tokens to facilitate model learning during training for the remaining 42 languages. As shown in Table 8, we find that alpha sampling achieves better accuracies than UniMax. In our study, we did not find there to be a strong transfer ability between programming languages as has been seen for multilingual languages.

Given that we mainly evaluate on high resource languages, we find it natural that alpha sampling, which places high weight on high resource languages without dramatically undersampling low resource languages, performs best. Further details on this ablation can be found in Appendix E.

Method	MultiPL-E	HumanEval
Alpha (α = 1.3)	**19.72**	**20.73**
UniMax (1e)	19.33	20.12

Table 8: Alpha sampling outperforms UniMax on code data. Per-language accuracies for MultiPL-E are shared in Table 23.

6 Data Attributes

6.1 Attribute Analysis

Findings

- Website homepages, news articles, and blogs constitute the majority of web crawl documents. Conversational texts are sparsely contained.
- Technical domains like finance, law, and science are among the least represented in web crawl.
- Explanatory or news articles on science and health are the most likely to be high quality documents.
- Domains or types of speech that are generally of high quality may also exhibit high toxicity (i.e news articles on sensitive topics), explaining why previous toxicity based filtering has harmed model accuracy.

We perform the first large-scale study of web crawl snapshots by using our aforementioned attribute classifiers to analyze all available CC snapshots until August 2023, over 90 in total. This analysis provides new insights into the composition of web crawl documents and identifies areas of data shortage, both of which can be used to improve the quality of pretraining sets. We detail our key findings below and further analysis can be found in Appendix F.

Figure 2 quantifies the proportion of documents belonging to various types of speech. Three major document types constitute over 65% of all web crawl examples: websites (homepages for organizations, products, and individuals), news articles, and blogs. This potentially explains the vastly improved world knowledge of recent LMs (Touvron et al., 2023b; Jiang et al., 2023) as news and blogs contain information on a wide range of topics while homepages provide factual information on people, places, and items. The lack of conversational texts highlights why alignment is needed to greatly improve the chat ability of pretrained models.

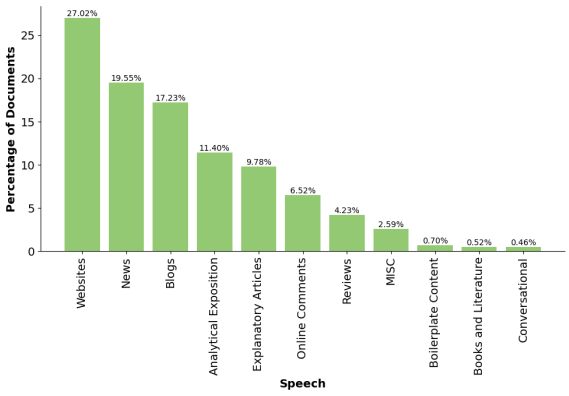

Figure 2: Distribution of document types in web crawl.

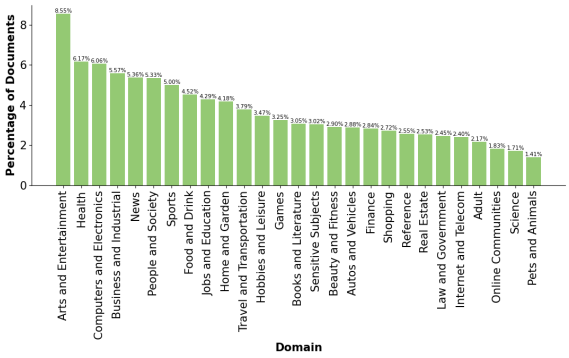

Figure 3: Distribution of content domains in web crawl.

Figure 3 illustrates the composition of content domains. The domains which are present in lower quantities are often technical in nature: finance, law, and science. To ensure that the model attains strong capabilities in these areas, it is pertinent to augment pretraining sets with data sources such as SEC filings (Wu et al., 2023), Court Listener (Henderson et al., 2022), and academic papers (Gao et al., 2020a; Touvron et al., 2023b).

We now examine how multiple data attributes vary with each other. Figure 4 shows the quality composition of each domain. As expected, technical domains like science, health and law contain the largest proportion of high quality content while adult and online communities are primarily of low quality. Surprisingly, sensitive subjects contains

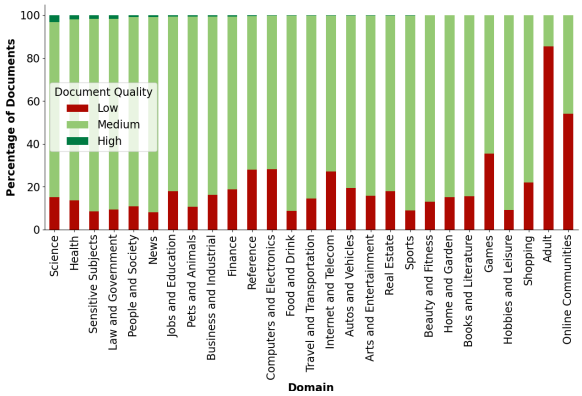

Figure 4: Domains sorted by descending order of percentage of high quality documents.

the third highest percentage of high quality examples. Looking at the distribution of domain by type of speech, which is detailed in Appendix F, the majority of sensitive subjects documents are news articles – indicating that these are well-written reports on topics such as war and protests.

Figure 5 shows the relationship between domain and toxicity. Sensitive subjects, likely due to the contained topics, is flagged for having high toxicity. This illustrates how toxicity based filtering can remove high quality documents and degrade LM quality as shown previously (Xu et al., 2021).

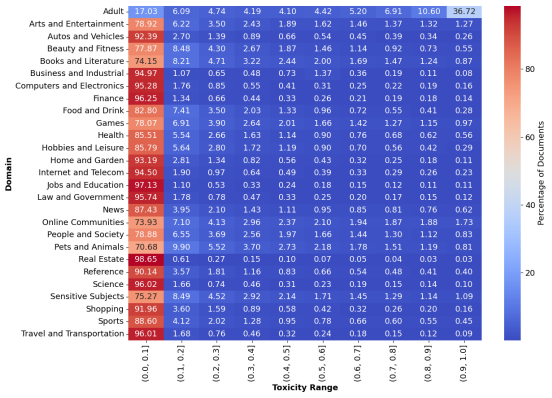

Figure 5: Heatmap of domains by probability of toxic content. Adult and online communities contain the highest percentage of toxic content.

6.2 Attributes in Sampling and Selection

Findings

- Buckets defined by data attributes substantially improve the performance of data sampling methods.
- Attributes compose more useful target sets for data selection.

Data attributes can refine pretraining set development as more exact target sets can be used in data selection and more informative buckets of data can be defined for which to assign weight distributions over during data sampling. We quantify the benefit of incorporating data attributes within both of these steps.

To use attribute information within data sampling, we define new buckets of examples based on the attributes. In one setting, which we term `fine-grained`, each existing data source is partitioned based on the attribute. A given CC snapshot CC-1 will now become $\{\text{CC-1-}X_i\}_{i=1}^{n}$ where each X_i is one of the n classes for the attribute. This means $\bigcup_{i=1}^{n} \text{CC-1-}X_i = \text{CC-1}$. An alternative setting, termed `grouped`, is to create attribute buckets across the entire corpus, C, such that $\bigcup_{i=1}^{n} X_i = C$, as each X_i consists of samples among all data sources with that given attribute label.

Experiment	LM-Eval
Baseline	56.81
Quality `fine-grained`	***57.88***
Quality `grouped`	*57.19*
Toxicity `fine-grained`	53.62
Toxicity `grouped`	54.99
Domain `fine-grained`	*57.34*
Domain `grouped`	*57.45*
Type of Speech `fine-grained`	56.69
Type of Speech `grouped`	*57.31*

Table 9: Sampling weights based on buckets of data attribute labels significantly improve upon baseline results. Italics indicate results that outperform the baseline. Per-task accuracies are shared in Table 24.

To assess the utility of attribute based data sampling, we train a 2B model for 165B tokens on a training set of 5 CC snapshots. Our baseline result is when attribute information is not included in data sampling. Further experimental details are shared in Appendix G. Table 9 highlights that all attributes aside from toxicity realize improved accuracy when used within data sampling. We note attributes which define broad classes of documents, like domain and type of speech, are more performant in the `grouped` setting while attributes that

assess a characteristic of a document, like quality, are better suited to the `fine-grained` setting.

Experiment	Target Set	LM-Eval
Original CC	N/A	54.90
DSIR	Wikipedia, Books	55.35
DSIR	Low Tox, High Qual	**55.63**

Table 10: Attribute information defines better target sets for data selection. Tox is Toxicity, Qual is Quality.

With data attributes, more precise target sets for data selection can be defined. For instance, one with examples that are of both low toxicity and high quality. Table 10 shows that using such a target set with DSIR outperforms the paper-recommended target set and enables toxicity based selection without accuracy degradation.

Additional angles where data attributes can refine pretraining sets would be through better selection of documents with information amenable for rephrasing (Maini et al., 2024) or seeding synthetic generation pipelines (Abdin et al., 2024).

7 Related Work

Data curation, which is the identification, organization, storage and cleaning of datasets (McLure et al., 2014; Freitas and Curry, 2016; Thirumuruganathan et al., 2020), has been the most well-studied aspect in pretraining set development. Early models, like BERT (Devlin et al., 2019) and XLNet (Yang et al., 2020), focused their data curation efforts on obtaining examples from high quality sources. In conjunction with the creation of larger collections of datasets such as C4 (Raffel et al., 2020), the Pile (Gao et al., 2020a), and BigScience ROOTS (Lachaux et al., 2020), heuristic and classifier based filters were used in data curation to remove ill-formed and useless documents (Rae et al., 2021; Chowdhery et al., 2022; Raffel et al., 2020). Additional studies within data curation found that data deduplication (Broder, 1997; Kandpal et al., 2022; Abbas et al., 2023) further improved model capabilities by preventing overtraining on a small set of similar examples.

Data selection and data sampling play major roles in pretraining set construction. Data selection methods (Moore and Lewis, 2010; Axelrod, 2017; Xie et al., 2023b; Engstrom et al., 2024) remove low quality documents to retain examples that more closely align with a predetermined high quality source. Moore-Lewis selection (Moore and Lewis, 2010) proposed the initial approach, with recent extensions by cynical data selection (Axelrod, 2017) and DSIR (Xie et al., 2023b) which both better estimate the probability that a given example belongs to a high quality domain. Data sampling techniques either use a learned model (Xie et al., 2023a; Albalak et al., 2023; Fan et al., 2024) or a heuristic function (Arivazhagan et al., 2019; Raffel et al., 2020; Chung et al., 2023) to define sampling weights for each data source. Learned techniques, such as DoReMi (Xie et al., 2023a), use the loss of a model across the data sources to define sampling weights while heuristic functions often use the size of a data source to explicitly define weights (Arivazhagan et al., 2019; Raffel et al., 2020) or fit a probability distribution (Chung et al., 2023).

The data attributes of toxicity and quality have been used to further refine pretraining sets (Gururangan et al., 2022; Meade et al., 2022). Toxicity classifiers (Welbl et al., 2021) that remove highly toxic examples reduce the number of toxic generations from LMs, but also negatively impact the model's other capabilities (Xu et al., 2021). Quality classifiers (Devlin et al., 2019; Raffel et al., 2020; Chowdhery et al., 2022) which remove documents such as machine generated texts (Dodge et al., 2021) or hate speech and sexually explicit content (Luccioni and Viviano, 2021) greatly improve model capabilities. (Longpre et al., 2023) extensively investigate the impact that toxicity, quality, and age of data have on model accuracy.

8 Conclusion

We present the first comprehensive study on pretraining set development conducted at the scale of modern day LMs and pretraining set sizes. Through a series of ablations, we identify helpful methods to use at each step of the pretraining development pipeline. We then analyze most currently available web crawl snapshots across the attribute labels of toxicity, quality, domain, and type of speech to better understand the composition of the most widely used data source in current pretraining corpora. These attribute labels are then shown to provide significant improvement in model abilities when incorporated within data selection and data sampling methods. We hope that the open transmission of this knowledge spurns more rapid advancements in the capabilities of LMs.

Limitations

While we designed our experimental setting to be as generally applicable as possible, we acknowledge that our findings are limited to the distribution of data sources, learning algorithm, and model configuration that we consider. Thus, when extrapolating our findings on pretraining set development to a setting with markedly different data sources or for usage in an alternate type of model, it may be that our results do not hold as strongly. In addition, we do not evaluate all possible techniques for each step of the pretreating pipeline so our results can not be thought of as the definitive rankings amongst all potential methods but rather as a set of strategies with which to create an effective, high-quality pretraining set. Lastly, although the use of synthetic data has recently garnered lots of attention, we did not include any such source of data within our studies and aspects relating to quality selection and sampling of synthetic data may be different than what our findings suggest.

References

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

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *Preprint*, arXiv:2402.16827.

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

Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, Logesh Kumar Umapathi, Carolyn Jane Anderson, Yangtian Zi, Joel Lamy Poirier, Hailey Schoelkopf, Sergey Troshin, Dmitry Abulkhanov, Manuel Romero, Michael Lappert, Francesco De Toni, Bernardo García del Río, Qian Liu, Shamik Bose, Urvashi Bhattacharyya, Terry Yue Zhuo, Ian Yu, Paulo Villegas, Marco Zocca, Sourab Mangrulkar, David Lansky, Huu Nguyen, Danish Contractor, Luis Villa, Jia Li, Dzmitry Bahdanau, Yacine Jernite, Sean Hughes, Daniel Fried, Arjun Guha, Harm de Vries, and Leandro von Werra. 2023. Santacoder: don't reach for the stars! *Preprint*, arXiv:2301.03988.

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. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *Preprint*, arXiv:1907.05019.

Amittai Axelrod. 2017. Cynical selection of language model training data. *Preprint*, arXiv:1709.02279.

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

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

BigScience. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *Preprint*, arXiv:2211.05100.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

A.Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 21–29.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg,

and Abhinav Jangda. 2023. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, pages 1–17.

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. 2021. Casino: A corpus of campsite negotiation dialogues for automatic negotiation systems. *arXiv preprint arXiv:2103.15721*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *Preprint*, arXiv:2304.09151.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *CoRR*, abs/2003.05002.

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*.

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

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *Preprint*, arXiv:2104.08758.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2019. Ccaligned: A massive collection of cross-lingual web-document pairs. *arXiv preprint arXiv:1911.06154*.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. 2024. Dsdm: Model-aware dataset selection with datamodels. *Preprint*, arXiv:2401.12926.

Miquel Esplà-Gomis, Mikel L Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. Paracrawl: Web-scale parallel corpora for the languages of the eu. In *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, pages 118–119.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. Doge: Domain reweighting with generalization estimation. *Preprint*, arXiv:2310.15393.

Oliver Ferschke, Iryna Gurevych, and Yevgen Chebotar. 2012. Behind the article: Recognizing dialog acts in wikipedia talk pages. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 777–786.

André Freitas and Edward Curry. 2016. *Big Data Curation*, pages 87–118. Springer International Publishing, Cham.

Leo Gao. 2021. An empirical exploration in quality filtering of text data. *arXiv preprint arXiv:2109.00698*.

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

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

Google DeepMind Gemma Team. 2024. Gemma: Open Models Based on Gemini Research and Technology.

Suchin Gururangan, Dallas Card, Sarah K. Dreier, Emily K. Gade, Leroy Z. Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A. Smith. 2022. Whose language counts as high quality? measuring language ideologies in text data selection. *Preprint*, arXiv:2201.10474.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Preprint*, arXiv:2207.00220.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *Preprint*, arXiv:2003.11080.

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

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. *Preprint*, arXiv:1912.07840.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. *Preprint*, arXiv:2202.06539.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Marie-Anne Lachaux, Baptiste Roziere, Lowik Chanussot, and Guillaume Lample. 2020. Unsupervised translation of programming languages. *Preprint*, arXiv:2006.03511.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. 2022. The bigscience roots corpus: A 1.6tb composite multilingual dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 31809–31826. Curran Associates, Inc.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *Preprint*, arXiv:2305.06161.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, toxicity. *Preprint*, arXiv:2305.13169.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Alexandra Sasha Luccioni and Joseph D. Viviano. 2021. What's in the box? a preliminary analysis of undesirable content in the common crawl corpus. *Preprint*, arXiv:2105.02732.

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. *Preprint*, arXiv:2401.16380.

Merinda McLure, Allison Level, Catherine Cranston, Beth Oehlerts, and Mike Culbertson. 2014. Data curation: A study of researcher practices and needs. *portal: Libraries and the Academy*, 14:139–164.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. *Preprint*, arXiv:2110.08527.

Benjamin S Meyers, Nuthan Munaiah, Emily Prud'hommeaux, Andrew Meneely, Josephine Wolff, Cecilia Ovesdotter Alm, and Pradeep Murukannaiah. 2018. A dataset for identifying actionable feedback in collaborative software development. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 126–131.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Dan Su, Chen Zhu, Deepak Narayanan, Aastha Jhunjhunwala, Ayush Dattagupta, Vibhu Jawa, Jiwei Liu, Ameya Mahabaleshwarkar, Osvald Nitski, Annika Brundyn, James Maki, Miguel Martinez, Jiaxuan You, John Kamalu, Patrick LeGresley, Denys Fridman, Jared Casper, Ashwath Aithal, Oleksii Kuchaiev, Mohammad Shoeybi, Jonathan Cohen, and Bryan Catanzaro. 2024. Nemotron-4 15b technical report. *Preprint*, arXiv:2402.16819.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Colin Raffel, Leandro Werra, and Thomas Wolf. 2024. FineWeb: decanting the web for the finest text data at scale.

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

Edoardo Maria Ponti, Goran Glavas, Olga Majewska, Qianchu Liu, Ivan Vulic, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. *CoRR*, abs/2005.00333.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *Preprint*, arXiv:2112.11446.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia

Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher.

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

Colin Raffel, Noam Shazeer, et al. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv*, abs/1910.10683.

Anna Rogers. 2021. Changing the world by changing the data. *Preprint*, arXiv:2105.13947.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *Preprint*, arXiv:1911.08731.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

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

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *Preprint*, arXiv:2204.07580.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022a. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *Preprint*, arXiv:2201.11990.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022b. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *CoRR*, abs/2201.11990.

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. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *Preprint*, arXiv:2402.00159.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Gemini Team. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Gemini Team. 2024b. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Reka Team, Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, Kaloyan Aleksiev, Lei Li, Matthew Henderson, Max Bain, Mikel Artetxe, Nishant Relan, Piotr Padlewski, Qi Liu, Ren Chen, Samuel Phua, Yazheng Yang, Yi Tay, Yuqi Wang, Zhongkai Zhu, and Zhihui Xie. 2024. Reka core, flash, and edge: A series of powerful multimodal language models. *Preprint*, arXiv:2404.12387.

Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. 2020. Data curation with deep learning. In *International Conference on Extending Database Technology*.

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. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *arXiv preprint arXiv:1906.06725*.

Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. 2021. Challenges in detoxifying language models. *arXiv preprint arXiv:2109.07445*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *Preprint*, arXiv:2303.17564.

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

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023b. Data selection for language models via importance resampling. *Preprint*, arXiv:2302.03169.

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, and Dan Klein. 2021. Detoxifying language models risks marginalizing minority voices. *Preprint*, arXiv:2104.06390.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding. *Preprint*, arXiv:1906.08237.

Jiacheng Ye, Xijia Tao, and Lingpeng Kong. 2023. Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability. *Preprint*, arXiv:2306.06688.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Ethan Zhou and Jinho D Choi. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34.

A Data Sources

A.1 English Data Sources

Table 11 shares the datasets which compose our English corpus. We share further detail on how we gathered the datasets from each category.

Data source	Dataset name	Tokens (B)
Web Crawl	CC 2022-40	284.3
	Re-crawled C4	174.8
	CC 2019-35	165.1
	CC 2020-50	141.9
	CC 2021-04	68.2
	Pile-CC	41.2
	OpenWebText2	14.0
News	CC NEWS	94.2
Misc	ROOTS	104.5
	Wikipedia	4.3
Conversational	Reddit + others	59.1
Books	Books3	25.1
	Stories	5.3
	Gutenberg	2.5
	BookCorpus2	1.5
Scientific	ArXiv	18.7
	StackExchange	9.8
	PubMed Abstracts	4.2
	NIH ExPorter	0.3

Table 11: Summary of each of the datasets that make up our English corpus

Web Crawl To acquire a significant amount of web-crawl data, we downloaded all Common Crawl web archive (WARC) files originating from the CC-2020-50, CC-2019-35, CC-2021-04 and CC-2022-40 snapshots. Additionally, we re-crawled all URLs provided by the documents within the C4 corpus (Raffel et al., 2019). While many of these URLs were no longer active, we were able to re-crawl approximately 1.7 TB of web pages contained within the C4 dataset. To add to our collected web-crawl corpus, we also used the pre-preprocessed documents available within Pile-CC (Gao et al., 2020b).

News To curate our news dataset, we downloaded all Common Crawl News WARC files between 2016 and October 2022.

Conversational Our conversational dataset was constructed primarily from the Pushshift Reddit dataset (Baumgartner et al., 2020), with small amounts of other public datasets such as CaSiNo (Chawla et al., 2021), Wikipedia Talk Pages (Ferschke et al., 2012), Persuasion for good (Wang et al., 2019), Friends (Zhou and Choi, 2018), Chromium, (Meyers et al., 2018) and Ubuntu dialogue conversational datasets (Lowe et al., 2015).

The Reddit dataset was pre-processed to ensure that only the longest conversation thread is sampled per post to avoid duplicate text that can arise from sampling many or all possible conversation subtrees (Zhang et al., 2022). Reddit usernames are anonymized with random alphanumeric strings while preserving speaker information within the conversation. Given the prevalence of toxic and harmful content on Reddit, we filter out conversations that have a toxicity score $>= 0.5$ according to Perspective API[1].

Books Our books dataset consisted of documents originating from the Books3, Gutenberg (PG-19), BookCorpus2 (all provided by the Pile), as well as documents from the CC-Stories dataset (Trinh and Le, 2018).

Scientific We curated all scientific documents from sub-datasets contained within the Pile. Specifically, we used the StackExchange, PubMed Abstracts, NIH Exporter and ArXiv datasets.

Misc As a miscellaneous category, we lump together the Wikipedia and ROOTS (Laurençon et al., 2022) datasets.

A.2 Multilingual Data Sources

Our multilingual dataset consists of 52 languages, 50 of which were curated from the CC-2022-40 Common Crawl snapshot. For Chinese and Japanese, we used documents from the mC4 corpus (Xue et al., 2020). This was a consequence of the inability of our text extraction library to parse languages without spacing. Table 12 provides a summary of the multilingual web crawl data that made up our multingual corpus.

Additionally, we used an English-centric sentence-level parallel corpus of 32 languages (Details in Table.13). This was collected largely from data sources such as CC-Matrix (Schwenk et al., 2019), CC-Aligned (El-Kishky et al., 2019) and Paracrawl (Esplà-Gomis et al., 2019). Multiple examples are formatted into a document using few-shot templates with the number of in-context examples from 0-10 following an exponentially decaying probability of selection.

A.3 Code Data Sources

Our source code dataset was mainly constructed from a subset of the Stack v1.2 dataset (Kocetkov et al., 2022). Table 14 list the selected languages and their token counts. While the dataset is distributed with each file as a single document, we pre-process the data further to concatenate all files of a particular language from a repository into a single long document to allow the model to attend across files.

B Data Attribute Classifiers

We detail the training methodology and output labels for each of our data attribute classifiers.

B.1 Toxicity Classifier

Solutions, like Perspective API, exist for quantifying the toxicity of a given piece of text. However, due to low rate limits it would be intractable to scale across the billions of documents that exist across all CC snapshots. In developing our own toxicity classifier, we aim to recapitulate the performance of Perspective API and reliably mark text which contain obscene language, threats, insults, and identity-based hate speech as having high toxicity. As a training set, we use 320K examples sourced from the Jigsaw[2] and Jigsaw Unintended[3] datasets. We obtain our final toxicity classifier by fine-tuning a DeBERTaV3 base model for 1 epoch on this data. The output for our toxicity classifier is a probability from 0 to 1 on whether or not a given piece of text contains toxic content.

We evaluate our toxicity classifier by measuring its correlation with Perspective API scores on a set of 50k documents from CC. We find that the classifier obtains a Pearson correlation of 0.8 which indicates high agreement between the models. Additionally, we ask a set of human annotators to label 500 documents with toxicity scores. On this

[1] https://perspectiveapi.com/

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

[3] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification

ISO	Tokens (B)	ISO	Tokens (B)	ISO	Tokens (B)	ISO	Tokens (B)
RU	94.52	FA	6.59	HI	2.60	IS	0.38
JA	70.52	RO	6.58	SK	2.58	UR	0.37
DE	48.98	TR	6.46	HR	2.45	AZ	0.37
ES	46.50	EL	6.43	CA	2.12	MR	0.33
FR	44.30	SV	6.39	LT	1.69	KA	0.32
ZH	43.41	HU	5.89	HE	1.47	MK	0.32
IT	26.40	AR	5.74	SL	1.33	NE	0.31
NL	15.64	NO	5.61	SR	1.24	KK	0.30
VI	15.16	FI	4.11	ET	1.24	HY	0.29
PL	14.50	DA	3.79	BN	0.90	GL	0.29
PT	11.99	UK	3.63	LV	0.84	ML	0.25
ID	10.90	BG	3.37	TA	0.82	TE	0.24
CS	7.23	KO	3.05	SQ	0.49	KN	0.18

Table 12: Summary of our multilingual web crawl data consisting of 52 languages. All languages except for JA and ZH were curated from the 2022-40 CC snapshot. The JA and ZH data were curated from the mC4 corpus.

Language	Percentage (%)	Language	Percentage (%)	Language	Percentage (%)	Language	Percentage (%)
Spanish	12.84	Indonesian	3.12	Japanese	2.30	Lithuanian	1.39
French	10.52	Portuguese	2.90	Norwegian	2.19	Bulgarian	1.30
German	9.78	Polish	2.88	Hungarian	2.13	Hindi	1.17
Italian	5.48	Czech	2.74	Ukrainian	1.90	Slovak	0.99
Russian	5.25	Turkish	2.60	Finnish	1.84	Slovenian	0.91
Dutch	4.81	Vietnamese	2.54	Swedish	1.73	Estonian	0.81
Chinese	3.61	Greek	2.39	Korean	1.54	Latvian	0.76
Arabic	3.20	Romanian	2.32	Danish	1.53	Croatian	0.55

Table 13: The language composition of our parallel machine translation corpus.

Language	Tokens (B)	Language	Tokens (B)	Language	Tokens (B)
Javascript	21.12	Rust	2.81	Pascal	0.68
Markdown	20.27	Jupyter	2.58	Assembly	0.67
Java	19.84	Ruby	2.29	Fortran	0.65
Python	19.49	Swift	2.02	Makefile	0.54
PHP	18.87	JSON	1.78	Julia	0.52
C	18.26	TEX	1.76	Mathematica	0.51
C++	15.79	Scala	1.29	Visual Basic	0.42
C#	12.05	YAML	1.28	VHDL	0.42
Go	9.03	Shell	1.18	Common Lisp	0.24
HTML	8.97	Dart	1.08	Cuda	0.21
Typescript	8.16	Lua	1.00	System Verilog	0.16
SQL	5.31	reStructuredText	0.96	Docker	0.16
CSS	4.96	Perl	0.83	Omniverse	0.03
XML	2.97	Haskell	0.72		

Table 14: Summary of our source code corpus consisting of 41 different programming languages all of which, except for omniverse, were curated from the Stack v1.2 dataset.

Heuristic	Threshold	English Only
N-gram LM Perplexity	5000	Yes
Fraction of non-alpha-numeric characters	0.25	Yes
Fraction of words without alphabets	0.20	Yes
Fraction of numbers (in characters)	0.15	
Fraction of URLs (in characters)	0.20	
Fraction of lines starting with bullets	0.90	
Fraction of whitespaces (in characters)	0.25	
Fraction of parentheses (in characters)	0.10	
The ratio of symbols to words	0.10	
Contains a word >1000 characters	1.0 (Hard Constraint)	
Contains <50 or >100k words	1.0 (Hard Constraint)	
Contains less than 2 common English words	1.0 (Hard Constraint)	Yes
Mean word length <3 or >10 characters	1.0 (Hard Constraint)	
Fraction of boilerplate content (in characters)	0.40	
Duplicate line fraction	0.30	
Duplicate paragraph fraction	0.30	
Duplicate lines (by character fraction)	0.20	
Duplicate paragraph (by character fraction)	0.10	
Repeating top n-gram fraction	0.20	
Repeating duplicate n-gram fraction	0.20	
Fraction of lines that do not end with punctuation	0.85	
Fraction of lines that end with ellipsis	0.30	
Documents containing Pornographic content in URLs	1.00	

Table 15: A list of document-level data filtering heuristics and thresholds. Heuristics are borrowed or derived from Rae et al. (2021) and C4's cleaning heuristics (Raffel et al., 2020)

Heuristic	Min. Threshold	Max Threshold
Fraction of comments (in characters)	0.001	0.85
Number of lines of code	5	20,000
Ratio of characters to tokens	2	-

Table 16: A list of file-level data filtering heuristics and thresholds applied to the source code data. Heuristics follow those described in (Allal et al., 2023).

held-out test set, we find that our toxicity classifier achieves an AUC-ROC of 0.83 while Perspective API attains an AUC-ROC of 0.85.

B.2 Domain Classifier

We train a domain classifier to label the content domain of a given piece of text into one of 27 potential classes: Adult, Arts and Entertainment, Autos and Vehicles, Beauty and Fitness, Books and Literature, Business and Industrial, Computers and Electronics, Finance, Food and Drink, Games, Health, Hobbies and Leisure, Home and Garden, Internet and Telecom, Jobs and Education, Law and Government, News, Online Communities, People and Society, Pets and Animals, Real Estate, Reference, Science, Sensitive Subjects, Shopping, Sports, and Travel and Transportation. The training data consists of 1 million CC documents which are labeled using Google Cloud's Natural Language API[4] and 500k Wikipedia articles that are curated using the Wikipedia-API [5]. We train a DeBERTaV3 on two epochs of this training set. We ask a set of human annotators to label 500 held-out CC documents

[4]https://cloud.google.com/natural-language/docs/classifying-text

[5]https://pypi.org/project/Wikipedia-API/

and evaluate both the Google API and our domain classifier on this test set. We find that our trained domain classifier matches the performance of the Google API as it achieves an accuracy of 77.9% while the Google API achieves 77.5%.

B.3 Quality Classifier

We train a quality classifier to label a given piece of text as either high, medium, or low quality. The training data consists of 25k CC examples that are labeled by 3 Surge AI[6] annotators. We ensure that all of these annotated documents had at least greater than 2 annotators in agreement on the quality label. In these annotations we provide the following definitions of each quality class to the annotators:

High Text which is grammatically correct, well-written, coherent between sentences and paragraphs, and without any missing punctuations or without any incomplete sentences. It also does not include any boilerplate text and has useful content.

Medium This text is mostly grammatically correct with minor errors. It may not be coherent throughout and can jump from topic to topic. It should not have many missing punctuations or incomplete sentences. It should not include a lot of boilerplate text and more than 50% of the text should be useful.

Low This category includes text which is not grammatical, not coherent at all, or contains a lot of missing punctuations, poor capitalization of words and incomplete sentences or abrupt paragraph breaks. If the text contains pornographic content, lewd or profane language or toxic content of any kind then it is de facto low quality. Text which has a lot of boilerplate content making more than 50% of the text useless should also be marked as "Low".

We train a DeBERTaV3 model on this training set and find that on a held-out test set of 23k additionally labeled examples, it achieves an accuracy of 83%.

B.4 Type of Speech Classifier

We train a type of speech classifier to label a given document into one of the following 11 document types: conversational, news, online comments, books and literature, blogs, analytical exposition (persuasive text), explanatory articles, reviews, product/company/organization/personal websites, boilerplate content, and miscellaneous. The training data consists of the same 25k CC examples labeled by 3 Surge AI annotators as the quality classifier training set. We ensure that all of these annotated documents had at least greater than 2 annotators in agreement on the type of speech label. In these annotations we provide the following definitions of each type of speech label to the annotators:

Conversational Is this text a conversation between two or more people? Does this piece of text sound like a response to something which is not mentioned in the document? If the answer to either of the questions is "Yes" then mark the document as belonging to this category. Conversations include podcast transcripts, talk show transcripts or if there is an exchange of thoughts, feelings, ideas or information between two or more people.

News News is a form of communication that informs the public of current events, issues, and trends in society.

Online Comments Comments are messages posted by users in reaction to social media or blog posts. They can take the shape of feedback, questions, praise, or even disagreements. Comment is a short-form type of content or message that gets published on social media platforms or other online communities. You may have to check the URL of the document to get a sense of the context of the text. This category encompasses social media comments, comments in online communities, and comments on an article or a blog.

Books and Literature Is the piece of text long and seems to span multiple pages? Does it have different chapters? If the response to either of the questions is "Yes" then mark the document as belonging to this category. This category also includes short stories that may be published on an online platform.

Blogs A blog (short for "weblog") is an online journal or informational website run by an individual, group, or corporation that offers regularly updated content (blog post) about a topic. It presents information in reverse chronological order and it's written in an informal or conversational style. You may have to look at the URL to check for this category. A blog typically has a title and addresses one topic throughout the text. Blogging has a highly

[6] https://www.surgehq.ai/

personal form of writing and authors demonstrate a connection with their blog content.

Analytical Exposition The social function of Analytical Exposition text is: To persuade the reader that there is an important and correct matter that, certainly, needs to get attention. Analytical exposition typically uses emotive words and simple present tense. This type of text contains ads for products, properties, items, companies etc. It may even be in the form of a blog persuading the reader to either buy a certain product or avail certain services. In such situations the text should be first marked as a "Blog" and then as "Analytical Exposition". This category includes persuasion, ads, and propaganda (text which is trying to sell the reader something or some idea).

Explanatory Article An explanatory article is a type of academic paper in which the author presents some point of view or opinion on a particular topic, subject, event or situation. Importantly, most of these articles provide references to the information presented in the text. This category includes Wikipedia articles, academic papers, abstracts of papers, Wiki How To articles or any piece of text plainly giving information for educational purposes. Note that any text that gives information is not an Explanatory Article. For example, in most cases ads also give information about a product but these should not be marked as Explanatory Articles. The purpose of Explanatory Articles is not to give information for selling something. These articles are also not written in conversation or informal format. They are written in a professional style and their sole purpose is to give information.

Reviews A review is a formal assessment or examination of something with the possibility or intention of instituting change if necessary. It is a critical article or report on a book, play, recital, movie, or an e-commerce product. A review typically provides a summary of the thing it is assessing, a reaction of the author and importantly a critical assessment of the thing.

Product/Company/Organization/Personal Websites Text that gives information about a product, company or organization falls into this category. The important thing is text in this category is authored and published by the same entity about which the information is given. For example a product website gives information about that product but a review website is written by someone else and will provide more than just the information about the product. Examples of this category are articles such as government websites giving information about their various programmes, organizations giving information about their services or products, schools giving information about courses, programmes, how to apply, jobs that are available etc.

Boilerplate Content Any written text (copy) that can be reused in new contexts or applications without significant changes to the original. Text and links in headers, footers, or sidebars are well-known examples. It could also be statements like "No search result" or email ids and addresses at the end of a website. Common examples of boilerplate are things like GDPR info about "cookies", "Google analytics" for websites. Things like "about info" at the bottom of websites etc. If there are any HTML artifacts remaining in the article, this should be marked as boilerplate. Examples of HTML artifacts are things like tables
, <tr>, <html>. Oftentimes, javascript needed to render the web page can be embedded into the text, this should also be marked as boilerplate.

Miscellaneous Other categories not covered here so far. If the text contains pornographic content, or toxic / lewd / profane language then by default you should mark it as "MISC".

We train a DeBERTaV3 model on this training set and find that on a held-out test set of 23k additionally labeled examples, it achieves an accuracy of 79.5%.

C Model Specifications

We detail the architecture and hyperparameters used for both the 2B and 8B models.

2B Model The architectural specifications include: 24 transformer layers, a hidden size of 2048, 16 attention heads, Rotary Position Embeddings (RoPE) (Su et al., 2023), SwiGLU (Shazeer, 2020) activations in the MLP layers, a SentencePiece (Kudo and Richardson, 2018) tokenizer with a vocabulary size of 256k, a context length of 4096, no bias terms, and untied input-output embeddings.

We train with a batch size of 256 and use a cosine learning rate schedule, with warmup over the first one percent of training tokens, to decay from a maximum learning rate of $2.0e$-4 to $2.0e$-5. We used the AdamW (Loshchilov and Hutter, 2019)

optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 0.1.

8B Model The architectural specifications include: 32 transformer layers, a hidden size of 4096, 32 attention heads, Rotary Position Embeddings (RoPE) (Su et al., 2023), SwiGLU (Shazeer, 2020) activations in the MLP layers, a SentencePiece (Kudo and Richardson, 2018) tokenizer with a vocabulary size of 256k, a context length of 4096, no bias terms, and untied input-output embeddings.

We train with a batch size of 1024 and use a cosine learning rate schedule, with warmup over the first one percent of training tokens, to decay from a maximum learning rate of $3.0e$-4 to $3.0e$-5. We used the AdamW (Loshchilov and Hutter, 2019) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 0.1.

D Data Curation Ablations

Table 17 illustrates that our specified steps of data curation for source code significantly improves evaluation performance, highlighting that data curation is a key component for all types of data.

Experiment	HumanEval	MultiPL-E
Raw source code	16.5	15.9
Post quality filtering	20.7	19.2

Table 17: Evaluation accuracies before and after data curation for our source code dataset. We train an 8B model for 150B tokens.

E Data Sampling Ablations

English We share the returned sampling weights for our English dataset across the three methods in Figure 6 and across the varying values of the UniMax maximum epoch hyperparameter in Figure 7. We clearly see that the returned weight distribution by DoReMi places too high of a weight on a single data source, which likely leads to its poor performance. Additionally, as the maximum epoch hyperparameter is increased in UniMax, the sampling distribution tends to a uniform one which likely begins to mitigate some of the utility gained from using the method.

Multilingual In our multilingual ablations, we first ran a series of experiments to identify the optimal α value to use in alpha sampling. We found that $\alpha = 1.3$ achieved the best downstream accuracies. We share the returned sampling distribution from each method in Figure 8.

Code Like in our multilingual ablations, we found that $\alpha = 1.3$ achieved the best downstream accuracies for alpha sampling in the code domain. We share the returned sampling distribution from each method in Figure 9. The DoReMi identified sampling distribution is not useful as it places over 80% of the weight on markdown.

F Data Attribute Analysis

Figure 10 illustrates that the vast majority of web crawl documents are of medium quality; however, there does exist a significant chunk of low quality documents which should be appropriately considered when creating pretraining sets. Additionally, Figure 11 highlights that a large proportion of web crawl documents are unlikely to contain toxic content (defined as having a toxicity score lower than 0.3). These two factors combined assure us that web crawl snapshots provide positive utility during langauge model pretraining.

Next, we examine the overlap between the output of the developed quality classifier and the perplexity scores of the KenLM model which we used to filter low quality documents during data curation. Figure 12 shows that the two models have high agreement on documents which they classify as high or low quality. This indicates that such model based filtering during data curation is able to reliably remove low quality texts.

In examining the quality composition of various types of speech categories, as shown in Figure 13, we find that explanatory and news articles are the document types which tend to contain the highest proportion of high quality texts. Additionally, we see that the boilerplate content and miscellaneous categories by far have the largest proportion of low quality documents, indicating that it likely would be best to completely filter out web domains which contain high proportions of documents of these types. This analysis allows for the appropriate prioritization of document types within web crawl snapshots as we now understand which sorts of texts are likely to be of the highest quality.

Lastly, Figure 14 highlights the distribution of domain by type of speech. We find that a lot of the technical domains, such as science, law, and health, are primarily composed of high quality types of speech, such as news and explanatory articles. This highlights that when prioritizing certain websites in

Experiment	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag
Raw text	55.6	57.2	39.9	73.9	57.6	58.9
Post deduplication	57.8	59.1	39.9	76.6	56.9	63.3
Post quality filtering	58.3	60.2	41.0	75.4	58.7	63.5

Table 18: Per-task evaluation accuracies of the experiments detailed in Table 2.

Experiment	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag
Random	59.8	59.4	41.9	75.6	59.9	63.1
Recent-to-Old	57.8	59.1	39.9	76.6	56.9	63.3
Old-to-Recent	59.4	60.8	41.3	76.0	61.7	63.5

Table 19: Per-task evaluation accuracies of the experiments detailed in Table 3.

Question	Experiment	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag
Q1	Original CC	51.3	53.6	37.1	73.6	54.3	55.9
	DSIR CC	53.1	55.0	37.2	73.2	54.4	53.8
Q2.1	Corpus DSIR	53.1	55.0	37.2	73.2	54.4	53.8
	Source DSIR	51.5	54.0	37.5	73.5	56.7	55.9
Q2.2	DSIR (80%)	53.3	5.40	37.4	72.5	56.5	53.6
	DSIR (87.5%)	53.5	53.1	37.9	72.0	55.0	54.0
	DSIR (95%)	51.5	54.0	37.5	73.5	56.7	55.9

Table 20: Per-task evaluation accuracies of the experiments detailed in Table 4.

Target Set	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag
Wikipedia, Books	51.5	54.0	37.5	73.5	56.7	55.9
Wikipedia, Books, arXiv, NIH	46.9	53.6	38.2	74.3	55.6	55.6
arXiv, NIH	47.2	54.2	36.3	73.9	56.5	55.3

Table 21: Per-task evaluation accuracies of the experiments detailed in Table 5.

Method	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag	MMLU
Preference	67.7	68.6	42.11	79.2	66.0	72.6	27.2
UniMax 1e	70.1	69.8	42.8	79.1	68.0	73.1	28.3
UniMax 2e	70.7	67.6	42.9	78.9	66.3	72.6	28
UniMax 4e	70.5	67.7	43.0	78.9	67.3	72.4	26.6
DoReMi	68.3	68.6	41.2	78.9	65.0	72.0	26.9

Table 22: Per-task evaluation accuracies of the experiments shared in 6.

Method	HumanEval	MP-Python	MP-Java	MP-JS	MP-CPP	MP-Lua
Alpha	20.72	20.5	23.4	20.5	19.3	14.5
UniMax	20.12	19.3	20.9	19.9	19.3	17.4

Table 23: Per-task evaluation accuracies for the experiments detailed in Table 8. MP stands for MultiPL-E.

future web crawls, it likely would be most fruitful to focus on those surrounding such domains. Additionally, the domain of sensitive subjects, which we identified as being primarily composed of high quality documents, is in fact made up mostly by news articles. This would indicate that this domain likely covers investigative reports on subjects such as war and protests. We also note that the

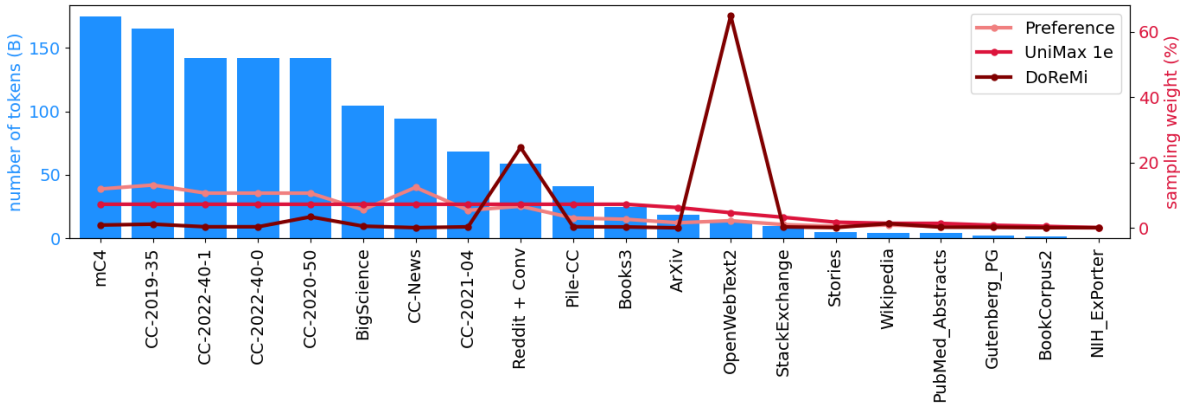

Figure 6: Returned samplings weights for the English dataset.

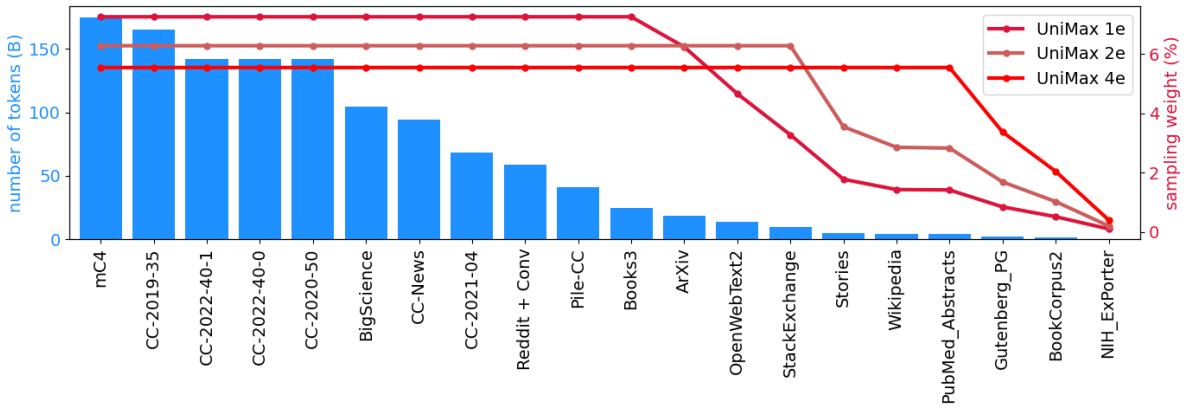

Figure 7: Effect of increasing the maximum epoch hyperparamter in UniMax on the returned sampling weights.

categories which we expect to have high overlap, like the domain and type of speech of news or the adult domain and the miscellaneous type of speech category, do in fact have a high degree of overlap. This confirms the efficacy of both our classifiers in providing accurate analysis.

G Data Attributes in Sampling and Selection

In this set of experiments, our baseline data sampling method is to proportionally weight each of the 5 CC snapshots by their token counts. We found that this sampling method performed better than UniMax. As the CC snapshots are all of relatively large token counts compared to our training token budget, 165B, UniMax ends up assigning a uniform distribution across each of the snapshots. As different CC snapshots have different utility, as indicated by (Penedo et al., 2024), a uniform distribution is suboptimal to one which weights snapshots differently.

In defining the sampling weights over both the `Fine-Grained` and `Grouped` settings of the attribute based buckets, we use UniMax with the maximum epoch hyperparameter set to 2.

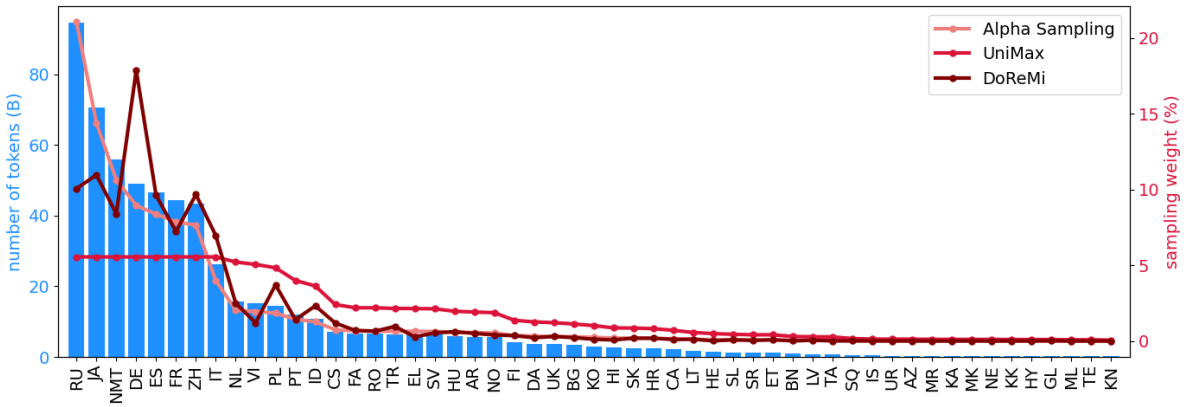

Figure 8: Returned samplings weights for the Multilingual dataset.

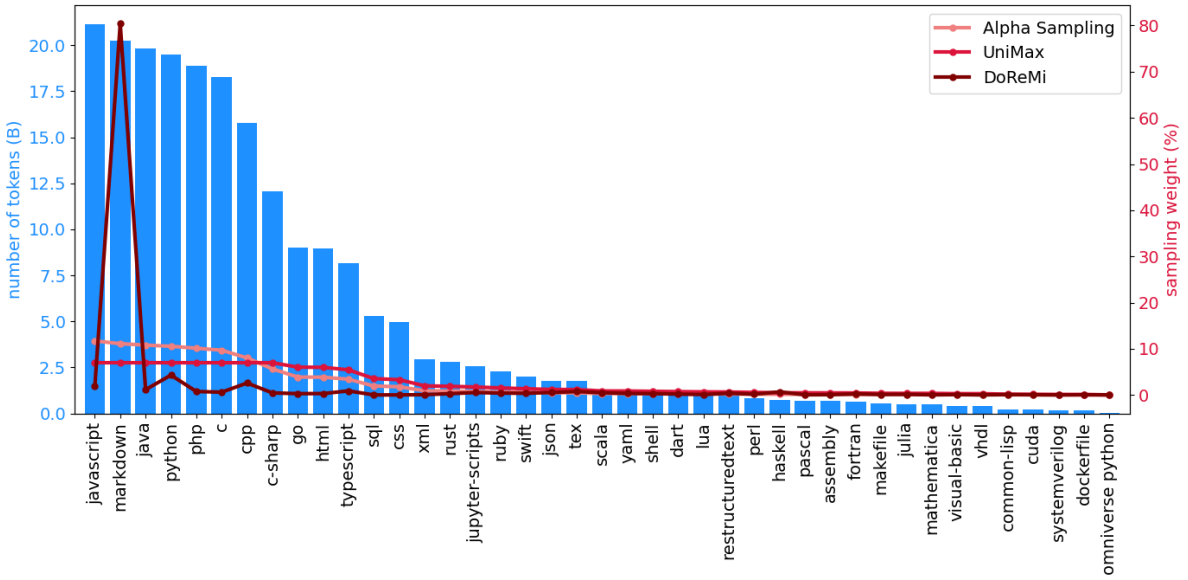

Figure 9: Returned samplings weights for the Code dataset.

Experiment	LAMBADA	ARC-easy	Race-H	PIQA	Winogrande	Hellaswag
Baseline	54.1	56.5	38.9	75.1	57.8	58.9
Quality `Fine-Grained`	57.3	57.7	39.7	75.0	57.6	60.0
Quality `Grouped`	56.2	56.6	38.7	74.2	56.8	58.3
Toxicity `Fine-Grained`	46.1	57.6	36.9	71.3	55.5	46.2
Toxicity `Grouped`	55.0	56.1	37.3	72.7	54.5	54.2
Domain `Fine-Grained`	57.0	60.7	39.5	73.3	56.5	57.0
Domain `Grouped`	54.6	59.7	40.2	73.9	59.2	57.1
Type of Speech `Fine-Grained`	53.4	59.2	37.5	74.3	56.2	59.5
Type of Speech `Grouped`	53.9	59.8	37.5	74.3	58.7	59.6

Table 24: Per-task evaluation accuracies of the experiments detailed in 9.

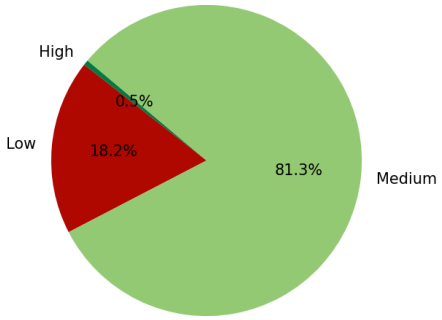

Figure 10: Breakdown of document quality across web crawl snapshots.

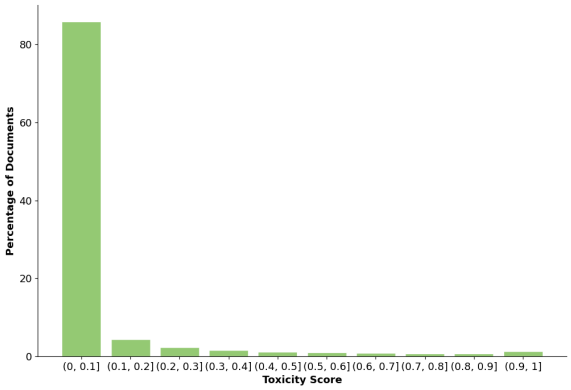

Figure 11: Breakdown of document toxicity across web crawl snapshots.

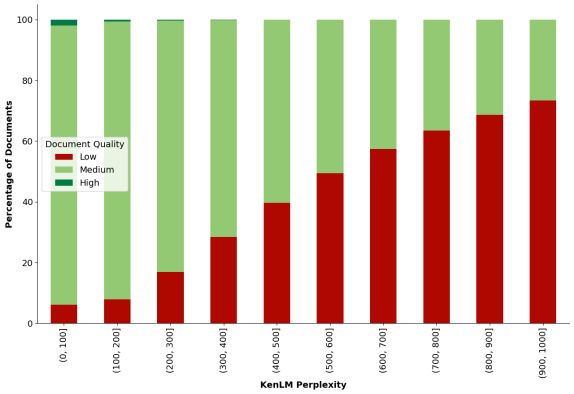

Figure 12: There is high correlation between the quality classifier and the perplexity of a KenLM model used for quality filtering during data curation.

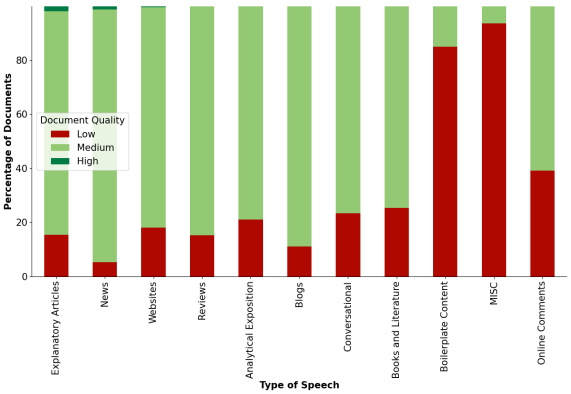

Figure 13: Types of speech sorted by descending order of percentage of high quality documents.

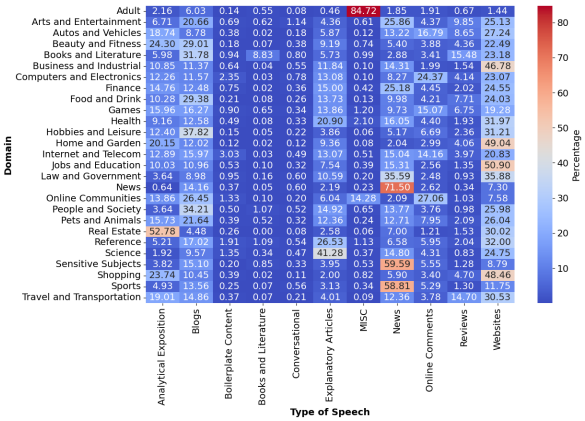

Figure 14: Heatmap of domains by types of speech.