Rethinking KenLM: Good and Bad Model Ensembles for Efficient Text Quality Filtering in Large Web Corpora

Anonymous ACL submission

Abstract

With the increasing demand for substantial amounts of high-quality data to train large language models (LLMs), efficiently filtering large web corpora has become a critical challenge. For this purpose, KenLM, a lightweight n-gram-based language model that operates on CPUs, is widely used. However, the traditional method of training KenLM utilizes only high-quality data and, consequently, does not *explicitly* learn the linguistic patterns of low-quality data. To address this issue, we propose an ensemble approach that leverages two contrasting KenLMs: (i) *Good KenLM*, trained on high-quality data; and (ii) *Bad KenLM*, trained on low-quality data. Experimental results demonstrate that our approach significantly reduces noisy content while preserving high-quality content compared to the traditional KenLM training method. This indicates that our method can be a practical solution with minimal computational overhead for resource-constrained environments.

1 Introduction

The advancement of large language models (LLMs) has accelerated as the *'scaling law'* (Kaplan et al., 2020), which states that the performance of LLMs directly correlates with data size, has been studied. Moreover, recent studies (Penedo et al., 2023; Gunasekar et al., 2023; Li et al., 2024; Penedo et al., 2024; Dubey et al., 2024) have shown that the performance of LLMs is largely determined by the quality of the training corpus. In other words, a vast amount of high-quality training corpus is necessary to enhance the performance of LLMs.

However, large web corpora often contain substantial amounts of low-quality data, making them difficult to use directly for training. In response to this challenge, various methods (Wettig et al., 2024; Kong et al., 2024) are employed to filter out low-quality data and select high-quality data. These methods typically require GPU resources, which makes them impractical, especially when processing data that exceeds trillions of tokens.

To *efficiently* filter large datasets, the most widely used method is KenLM (Heafield, 2011), a lightweight n-gram-based model that operates on *CPUs*. In many studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Laurençon et al., 2024), KenLM, trained on the high-quality Wikipedia dataset, is commonly used. It measures perplexity (PPL) to identify low-quality content. Note that higher PPL scores indicate lower-quality or out-of-domain text, while lower PPL scores suggest that the text closely resembles the linguistic patterns of the high-quality data used to train KenLM. Low-quality data with high PPL scores are then filtered out.

We argue that the traditional KenLM does not *explicitly* learn the linguistic patterns of low-quality data. Thus, while it assigns low PPL scores to data with high-quality linguistic patterns, it does not *consistently* assign high PPL scores to data with low-quality linguistic patterns. To address this issue, we propose an ensemble approach that utilizes the following two contrasting KenLMs: (i) *Good KenLM*, trained on high-quality data; and (ii) *Bad KenLM*, trained on noisy, low-quality data such as spam emails, hate speech, and informal social media text. Our empirical results show that this approach can be a practical solution with minimal computational overhead for resource-constrained environments, significantly reducing noisy content and preserving high-quality content compared to the traditional KenLM training method.

2 Related Work

As the demand for a vast amount of high-quality training corpus grows, it has become essential to *effectively* and *efficiently* filter large amounts of web corpus. Among various filtering methods, this paper focuses on model-based quality filtering,

which can be broadly divided into the following two categories: (i) perplexity-based filtering; and (ii) classifier-based filtering.

Perplexity-based filtering. Numerous studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Wei et al., 2023; Paster et al., 2023; Laurençon et al., 2024) use the perplexity (PPL) scores of KenLM (Heafield, 2011), an n-gram-based language model, to *efficiently* filter out low-quality data due to its lightweight architecture. It can operate on *CPUs*, making it a cost-efficient solution for handling large-scale text data. Despite its efficiency, there have been few efforts to improve its performance. Meanwhile, The Pile (Gao et al., 2020) used the perplexity of GPT-2 (Radford et al., 2019) and GPT-3 (Brown, 2020) to evaluate the quality of the dataset.

Classifier-based filtering. FastText (Joulin et al., 2016) is widely used to distinguish the quality of data (Computer, 2023; Wei et al., 2023; Li et al., 2024). Similar to KenLM, FastText is also an efficient model that operates on CPUs. However, as detailed in Section 4, KenLM demonstrated superior performance compared to FastText when both were trained on the same dataset. Furthermore, recent research (Gunasekar et al., 2023; Li et al., 2024; Penedo et al., 2024) has focused on fine-tuning pre-trained embedding models to serve as classifiers for quality filtering. Especially, Fineweb demonstrated that training relatively small-sized LLMs (1.82 billion parameters) on data filtered by a trained classifier (resulting in 350 billion tokens), rather than on unfiltered data, led to performance improvements across various benchmarks. However, these methods are impractical for processing large web corpora due to their high computational costs, which necessitate significant GPU resources.

3 Proposed Method

In this paper, we aim to reduce noisy data while preserving high-quality data in a computationally efficient manner. To this end, we propose an ensemble approach using two contrasting KenLMs: (i) Good KenLM and (ii) Bad KenLM.

Good KenLM. The objective of Good KenLM is to assign low perplexity (PPL) scores to well-structured, high-quality text. Many previous studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Laurençon et al., 2024) have used a high-quality Wikipedia dataset for training, denoted as Wiki KenLM in this paper. However, with recent advancements in LLMs, several high-quality datasets (Soldaini et al., 2024; Penedo et al., 2024; Li et al., 2024) have emerged. In our experiments, as shown in Section 4, we found that the combination of S2ORC (Lo et al., 2020) and Textbooks-are-all-you-need-lite (SciPhi, 2023) as training data was more effective than utilizing Wikipedia. Thus, in this paper, we designate the KenLM trained on this combination of data as Good KenLM.

Bad KenLM. The rationale behind employing Bad KenLM alongside Good KenLM is that Good KenLM fails to detect unwanted content (*e.g.*, spam, advertising, and informal communication), which are generally considered poor for training LLMs, as it has not been *explicitly* trained on these types of content. For instance, if low-quality content shares superficial linguistic patterns with high-quality text, it may still score reasonably well under Good KenLM. Therefore, to detect a wider range of undesirable content, Bad KenLM is designed to assign low PPL scores to such content. To achieve this, we trained Bad KenLM using noisy, low-quality datasets, including hate speech, spam emails, and informal social media content. To the best of our knowledge, this is the first study to employ KenLM trained on noisy, low-quality datasets.

Ensemble. To leverage the complementary strengths of two contrasting KenLMs, we ensemble the models by integrating the PPL scores assigned by each. We perform Z-score standardization to align the scales of the two PPL scores assigned by each model, as they are trained on different datasets and therefore exhibit different distributions of PPL scores. Then, we compute the ensembled PPL score $P_{ens}(x_i)$, as follows:

$$P_{\text{ens}}(x_i) = \alpha \left(\frac{P_{\text{good}}(x_i) - \mu_{\text{good}}}{\sigma_{\text{good}}} \right) - (1-\alpha) \left(\frac{P_{\text{bad}}(x_i) - \mu_{\text{bad}}}{\sigma_{\text{bad}}} \right), \quad (1)$$

where $x_i \in \mathcal{X}$ denotes the i-th text data, $\mathcal{X}$ represents datasets, $P_{\text{good}}(x_i)$ (resp. $P_{\text{bad}}(x_i)$) indicates PPL score from Good (resp. Bad) KenLM for x_i, μ_{good} (resp. μ_{bad}) is the mean of the PPL scores from Good (resp. Bad) KenLM, σ_{good} (resp. σ_{bad}) is the standard deviation of the PPL scores from Good (resp. Bad) KenLM, and α denotes a parameter that balances the two PPL scores. Note that the coefficient for the term associated with Bad

KenLM is negative. This is because, in contrast to Good KenLM, which assigns low PPL scores to high-quality data, Bad KenLM assigns low PPL scores to low-quality data. Consequently, data with low ensembled PPL scores—obtained by appropriately subtracting two PPL scores—closely resemble the linguistic patterns of high-quality data and are distinctly separated from low-quality content.

4 Experiments

We designed our experiments to answer the following key research questions (RQs):

- **RQ1**: Does our ensemble approach outperform existing models in removing noisy content while preserving high-quality content?
- **RQ2**: Which data sources are effective for training the Bad KenLM?
- **RQ3**: How sensitive is the performance of our ensemble approach to hyperparameter α?
- **RQ4**: How much additional computational overhead does our ensemble approach introduce compared to a single KenLM?
- **RQ5**: What types of data does our ensemble approach effectively filter out?

4.1 Experimental Settings

Dataset and model details. As mentioned in Section 3, we randomly selected subsets of 300,000 samples each from S2ORC (Lo et al., 2020) and Textbooks-are-all-you-need-lite (SciPhi, 2023) as training data for Good KenLM. For the training data of Bad KenLM, we collected datasets that is likely to hinder the training of LLMs. Specifically, we used 1,000,000 pieces of social network service (SNS) data (Twitter) (mjw, 2022; fschatt, 2023; lcama, 2022; StephanAkkerman, 2023) and 776,142 pieces of spam message data (Metsis et al., 2006; thehamkercat, 2024; alissonpadua, 2024). During the training of both models, we configured the n-gram size to 6 and the vocabulary size to $65,536$. Also, we set the hyperparameter α to 0.7.

Evaluation details. To evaluate the effectiveness of our ensemble approach, we measured perplexity (PPL) scores for the CC-MAIN-2024-10 dump (211 million samples) from Fineweb-edu (Penedo et al., 2024). Following Wenzek et al. (2019); Computer (2023), we then filtered the data based on the 30th and 60th percentiles of PPL scores. Subsequently, we measured the proportion of data with an *educational score* of 2.5 or higher that was included. In other words, we treated data with an educational score of 2.5 or higher as the ground truth and measured the recall value. Note that the educational scores are annotated using extensive GPU resources, and it has been demonstrated that training LLMs with data possessing high educational scores leads to performance improvements.

Models	Recall@30	Recall@60	Average Recall
Wiki KenLM	0.5530	0.8513	0.7022
Good KenLM	0.7059	0.9195	0.8127
Bad KenLM	0.3403	0.7031	0.5217
FastText(Wiki, Bad)	0.6453	0.8878	0.7665
FastText(Good, Bad)	0.7462	0.9412	0.8437
Ens(Good, Bad)	**0.8190**	**0.9647**	**0.8919**
Ens(Good, Wiki)	0.6312	0.8898	0.7605

Table 1: Performance comparison of our approach with existing models, and an ablation study on our design choices.

4.2 Main Results

We highlight the best results in bold and the second-best results with an underline in the tables.

RQ1: Comparison of existing models. As shown in Table 1, our Good KenLM significantly outperformed the widely used Wiki KenLM. Although Bad KenLM alone showed poor performance, our strategy of ensembling it with Good KenLM outperformed even FastText trained on the same data, improving Recall@30 and Recall@60 by 9.76% and 2.50%, respectively.

Moreover, to validate the effectiveness of Bad KenLM within our ensemble framework, we conducted a comparative experiment where Good KenLM and Wiki KenLM were ensembled in place of Bad KenLM, denoted as *Ens(Good, Wiki)*. The performance of *Ens(Good, Wiki)* was lower than that of Good KenLM alone. This is likely due to the relatively lower quality of the Wikipedia dataset compared to the training data used for Good KenLM, which negatively impacts its overall performance. This result also highlight the importance of incorporating Bad KenLM into the ensemble, as it successfully identifies undesirable content that Good KenLM may overlook.

RQ2: Impact of data sources on training Bad KenLM. The training dataset for Bad KenLM is diverse, including SNS, spam mail, and toxic datasets (Davidson et al., 2017; de Gibert et al., 2018; Kennedy et al., 2020; Mathew et al., 2021; Vidgen et al., 2021; Pavlopoulos et al., 2022) containing hate speech and profanity. We conducted

Training Dataset of Bad KenLM	Metrics Recall@30	Recall@60	Average Recall
Spam	0.8059	0.9576	0.8818
Twitter	0.8131	**0.9651**	0.8891
Toxic	0.7320	0.9402	0.8361
Spam + Twitter	**0.8190**	0.9647	**0.8919**
Spam + Toxic	0.7885	0.9545	0.8715
Twitter + Toxic	0.7973	0.9602	0.8788
Spam + Twitter + Toxic	0.7906	0.9533	0.8720

Table 2: The effect of data sources on Bad KenLM training.

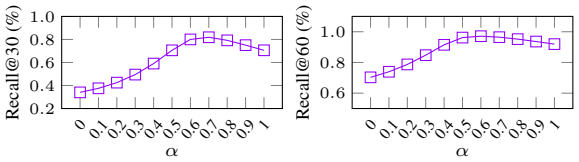

Figure 1: The effect of α on the performance of our ensemble approach.

experiments to determine which of these data sources are effective for training Bad KenLM. In this experiment, we ensembled our Good KenLM with various Bad KenLMs, each trained on different combinations of datasets.

As shown in Table 2, SNS data (Twitter) proved to be the most effective for training Bad KenLM, which is designed to filter out noisy content unsuitable for LLM training. Interestingly, toxic datasets led to a decrease in the performance of Bad KenLM. Unlike SNS data or spam mail, which share similar distributions with web data, toxic datasets contain a large proportion of highly offensive language, resulting in a substantial distributional difference. This discrepancy seems to adversely affect the training process of Bad KenLM.

RQ3: Hyperparameter sensitivity analysis. The parameter α in Eq. (1) adjusts the balance between the PPL scores of Good KenLM and Bad KenLM. We analyze how the performance of our ensemble approach varies with changes in α in terms of Recall@30 and Recall@60.

As depicted in Figure 1, Recall@30 and Recall@60 continuously improve as α increases to 0.7 and 0.6, respectively, and then gradually decrease. These results suggest that when α is too small, the influence of Bad KenLM becomes overly dominant, resulting in poor preservation of high-quality content. Conversely, when α is too large, the influence of Good KenLM prevails, leading to the inclusion of some low-quality content. These results indicate that appropriately determining the value of α is critical for effectively removing noisy content while preserving high-quality content.

RQ4: Degree of computational overhead. To assess the computational overhead of our approach, we measured the processing time and estimated cost[1] for the CC-MAIN-2024-10 dump on a machine with 128-core CPUs. As presented in Table 3, our approach increased the processing time from 2,234 to 3,928 seconds, with an additional cost of $1.08. These increases are justified by the recall improvement from 81.27% to 89.19%, as high-quality data is crucial for effective LLM training.

Models	Processing Time	Estimated Cost	Throughput	Avg. Recall
Good KenLM	2,234s	$1.42	94.4k docs/s	0.8127
Ens(Good, Bad)	3,928s	$2.50	53.7k docs/s	0.8919

Table 3: Comparison of computational overhead and performance for the CC-MAIN-2024-10 dump between Good KenLM and our ensemble approach.

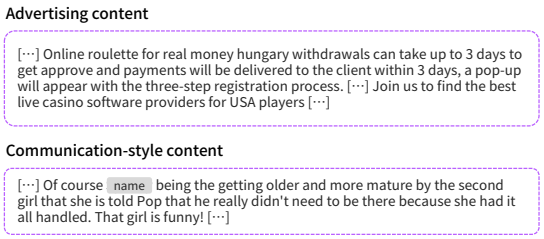

Figure 2: Visualization of examples that are not filtered by Good KenLM but are successfully removed by our ensemble approach.

RQ5: Case study on the effectiveness of our approach. To demonstrate the effectiveness of our ensemble approach, we present examples that are not filtered by Good KenLM but are successfully removed by our ensemble approach. As illustrated in Figure 2, our approach effectively filters advertising and communication-style content, which are generally unsuitable for LLM training. Since advertising content is usually written politely, Good KenLM, trained only on high-quality datasets, struggles to detect it. Conversely, Bad KenLM, trained on spam mail and SNS data, successfully identifies such content as well as communication-style content. Therefore, our ensemble approach more effectively filters these types of content.

5 Conclusion

In this paper, we propose an ensemble approach using Good KenLM and Bad KenLM for effective text filtering. By integrating perplexity scores, we successfully filter out noisy data, such as spam and informal content, while preserving high-quality text. Empirical results suggest that our approach could be a practical solution for filtering large-scale datasets in resource-constrained environments.

[1]It was measured using an AWS r6a.32xlarge (Amazon Web Services, 2022) spot instance.

Limitations

While the proposed method using Good KenLM and Bad KenLM offers effective filtering of large-scale datasets, it has the following limitations: (i) although our method has demonstrated effectiveness through extensive experiments using Fineweb-edu, we have not been able to measure its direct impact on LLMs training due to computational cost constraints; and (ii) the model relies heavily on predefined training datasets, and its performance may degrade when applied to content that significantly differs from the training corpora.

Ethics Statement

The experiments conducted in this paper were carried out objectively and fairly. No biases were introduced during the data selection or evaluation process. All datasets used in this research are publicly available, and the methods were rigorously tested to ensure the reliability and validity of the results.

References

alissonpadua. 2024. Ham spam scam toxic.

Amazon Web Services. 2022. Amazon ec2 r6a instances. Accessed: September 14, 2024.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

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

fschatt. 2023. Trump tweets.

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

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

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

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.

Xiang Kong, Tom Gunter, and Ruoming Pang. 2024. Large language model-guided document selection. *arXiv preprint arXiv:2406.04638*.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2024. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36.

lcama. 2022. Elon tweets.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes - which naive bayes?

mjw. 2022. sotck market tweets.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*.

John Pavlopoulos, Léo Laugier, Alexandros Xenos, Jeffrey Sorensen, and Ion Androutsopoulos. 2022. From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022).*, Dublin, Ireland. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

SciPhi. 2023. Textbooks are all you need : A sciphi collection.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

StephanAkkerman. 2023. Crypto stock tweets.

thehamkercat. 2024. Telegram spam ham.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. *Preprint*, arXiv:2012.15761.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. Polylm: An open source polyglot large language model. *arXiv preprint arXiv:2307.06018*.

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

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52915–52971. PMLR.