An Integrated Data Processing Framework for Pretraining Foundation Models

Yiding Sun[†]
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
emanual20.sun@gmail.com

Feng Wang[†]
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
rucwangfeng@gmail.com

Yutao Zhu
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
yutaozhu94@gmail.com

Wayne Xin Zhao
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
batmanfly@gmail.com

Jiaxin Mao[*]
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
maojiaxin@gmail.com

ABSTRACT

The ability of the foundation models heavily relies on large-scale, diverse, and high-quality pretraining data. In order to improve data quality, researchers and practitioners often have to manually curate datasets from difference sources and develop dedicated data cleansing pipeline for each data repository. Lacking a unified data processing framework, this process is repetitive and cumbersome. To mitigate this issue, we propose a data processing framework that integrates a Processing Module which consists of a series of operators at different granularity levels, and an Analyzing Module which supports probing and evaluation of the refined data. The proposed framework is easy to use and highly flexible. In this demo paper, we first introduce how to use this framework with some example use cases and then demonstrate its effectiveness in improving the data quality with an automated evaluation with ChatGPT and an end-to-end evaluation in pretraining the GPT-2 model. The code and demonstration video are accessible on GitHub[1].

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

Large Language Models; Data Quality; Data Processing

ACM Reference Format:
Yiding Sun[†], Feng Wang[†], Yutao Zhu, Wayne Xin Zhao, and Jiaxin Mao[*]. 2024. An Integrated Data Processing Framework for Pretraining Foundation Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3626772.3657671

[1]https://github.com/Emanual20/Yulan-GARDEN

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657671

1 INTRODUCTION

The exceptional performance of large language models (LLMs) in numerous downstream tasks stems from the extensive knowledge of the foundation models [24, 27, 28, 31, 34, 37]. According to the scaling law [14], the performance of these foundation models depends on the large-scale, high-quality preprocessed pretraining data. The inclusion of diverse large-scale pretraining data aids in enhancing the generalization ability [1, 5, 25], while high-quality data leads to improved training efficiency and effectiveness of LLMs [18, 29, 35].

However, preprocessing the pretraining data is a challenging and time-consuming task. First, data comes from various sources [6, 8, 16, 33], and different LLMs rely on different data recipes (*i.e.*, data mixture proportions) [5, 12, 25, 26, 32], each with its corresponding processing pipeline [30]. Reproducing such a process for each data repository becomes complicated and time-consuming. Additionally, there is a lack of standardized evaluation metrics for pretraining data. So the practitioners often have to use the performance in downstream tasks to measure the quality of the pre-training data. This further complicates the data collection and reprocessing for foundations models as training and fine-tuning these models need high computational costs, especially considering the current trend of exponential growth in model parameters [35].

To address the aforementioned issues, we propose an integrated data processing framework for pretraining data of foundation models. This framework consists of two main modules: **Processing Module** and **Analyzing Module**, as illustration in Figure 1. The Processing Module comprises data processing operators of varying granularity, primarily implemented by heuristic and language model-based rules. These operators preprocess data from granularities of document, paragraph, and sentence. The Analyzing Module consists of Evaluator, Retriever, and Debugger, which can assist

† Equal contribution.
* Corresponding author.

users in gaining an intuitive understanding of the dataset. Therefore with this framework, users can first probe the data to obtain preliminary insights with the Retriever and then customize data cleansing pipelines by flexibily combining and scheduling predefined operators provided in the Processing module, without coding manually. Additionally, users can utilize the Evaluator in Analyzing Module to reconfirm whether the refined dataset meets training requirements, enabling further iterations to enhance data quality.

We further conduct two experiments to validate the effectiveness of the proposed framework in improving the data quality. Through automated evaluation with ChatGPT, we observe a significant improvement in the quality of refined dataset when applied to OpenWebText2, Wikipedia, and HackerNews [8]. In the end-to-end evaluation, we train two GPT-2 models using CommonCrawl before and after processing, respectively. The model trained on the refined data demonstrates remarkable performance enhancement across downstream tasks of language modeling compared to the baseline.

2 USE CASES

2.1 Integrated Data Processing Framework

The inherent characteristics of datasets essentially influence the behavior of machine learning models [9]. Consequently, as illustrated in Figure 1, before the data-refined processing pipeline, users can probe into the raw dataset to gain insights into the dataset distribution, retrieve specific texts, and determine the parameters of the configuration file by Analyzing Module. The Evaluator demonstrates the distribution of text length, perplexity, language, etc. While the Retriever handles users' search queries and retrieve documents, the Debugger presents matching cases of Cleaner and the relation between filter parameters and the filter ratio of Filter.

After fetching results from the Analyzing Module, users can customize configuration files to meet dedicated data processing needs, e.g. filter short texts, remove exact strings, and deduplicate. With the guidance of a user-defined configuration file, the submodules of the Processing Module collaborate to refine the raw dataset.

Finally, users can probe into the refined dataset by Analyzing Module to compare the difference with the raw dataset. If the refined dataset still fails to fulfil the criteria of pretraining foundation models, users can modify the configuration file based on existing deficiencies and repeat the above steps until the data meets users' requirements. For example, users can remove gibberish fragments without actual symantic by setting the parameter 'fil_ppl' as 3 to filter texts with perplexity over avg.3+s·std. If users still find gibberish in the refined dataset, they can adjust the parameters and reprocess the data until it meets the requirements for training.

2.2 Retriever Augmentation

In the pretraining stage, foundation models may suffer from hallucination, due to the inconsistent distribution of entities between pretraining data and users' real needs. Existing work introduced Retrieval Augmented Generation (RAG) to involve external knowledge to mitigate hallucination [4].

In our framework, the Retriever Module can help researchers detect whether specific entity-level or topic-level texts exist in pretraining datasets by retrieving relevant keywords. For example, when users' language model checkpoint has no knowledge about "Renmin University", to bridge the gap between the facts and the model's generation results, users can search information about Renmin University in the dataset by Retriever. If there is no relevant information about Renmin University in retrieved texts, users will need to gather relevant texts from other datasets by Retriever to get a specialized dataset of Renmin University.

3 IMPLEMENTATION

3.1 Processing Module

Processing Module can refine the raw datasets by unifying different formats, filtering and denosing irrelevant information, and deduplicating. The Processing Module consists of four components: Reformatter, Filter, Cleaner, and Deduplicator.

Reformatter. To accommodate datasets from diverse sources, we reformat the raw dataset into jsonlines, where each row is a JSON dictionary including a "text" field and other meta information, aligning with the format of Huggingface datasets.

Filter. Filter fixes the datasets by discarding contentless texts (e.g., crawled thin content webpage), non-target language texts, ambiguous texts, and toxic texts. Filter excludes the texts from two perspectives. First, it employs labels annotated by trained language models, such as language and perplexity. Second, Filter excludes texts according to some handcrafted features, such as the occurrence of dirty words, the proportion of Alphabet or Chinese characters, the proportion of short lines, and number of entities.

Cleaner. Due to the specific formatting of data sources, some datasets may include content with weak semantics or with personal privacy, such as HTML tags in crawled web data or reporter names in news data. Discarding such data directly would significantly reduce the scale of the training corpus, especially considering some rare data sources. Consequently, we design Cleaner for eliminating useless information while keeping useful texts. Cleaner fixes texts on the dimensions of strings, lines, and paragraphs using exact match and regular expression match. Additionally, Cleaner can convert the texts into another target language and extract contents from files of specific formats, such as HTML, epub, mobi, and PDF.

Deduplicator. The repetition in a dataset can potentially cause the strong double descent phenomenon [21], indicating repetition may lead to an increase in test loss midway through training. This phenomenon highlights the importance of discarding repeated data before the training process to prevent such degradation in performance. Therefore, Deduplicator utilizes the MinHash Locality Sensitive Hashing (MinHashLSH) [13, 15, 20] algorithm to remove duplicate texts from datasets. We split the texts into n-gram sequences, where n is set as 10. The similarity threshold is 0.7.

3.2 Analyzing Module

Analyzing Module helps users facilitate a more profound comprehension of datasets through statistics analysis, specific domain knowledge retrieval, and parameter analysis of Filter and match cases of Cleaner. There are three components in the Analyzing Module, including Evaluator, Retriever, and Debugger.

Evaluator. To conduct the analysis of statistical features of both the raw dataset and refined dataset, Evaluator employs a visualization approach to present the data distribution (text length, perplexity, language, etc.) in a user-friendly manner.

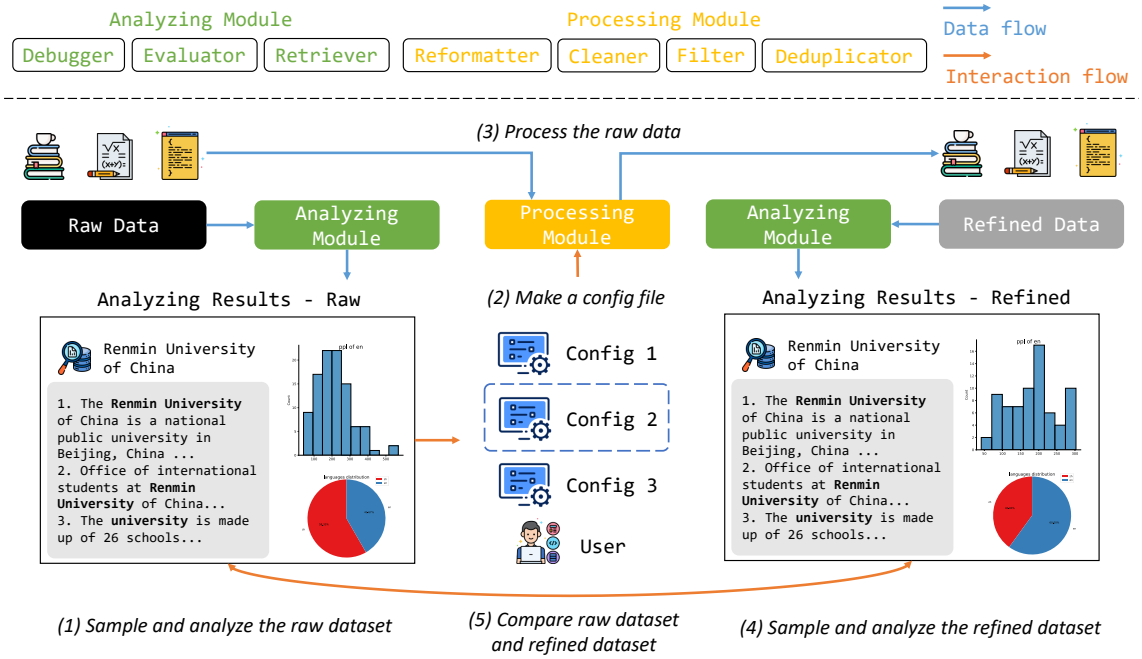

Figure 1: Overview of our data processing framework.

Retriever. During the training process, checkpoints of foundation models may face the issue of being clueless about an entity, leading to the need to supplement specific entity or topic-level knowledge. Therefore, Retriever integrates a simple search engine implemented by ElasticSearch to retrieve entity-level or topic-level texts, which can help users determine whether the dataset contains texts related to the field. The similarity of text is modeled by BM25. The index is divided into 20 shards for distributed storage and processing across multiple nodes.

Debugger. Many parameters in the Filter and Cleaner cannot be directly determined before the refining process. Consequently, we design Debugger to demonstrate the filtering or cleansing effects induced by different parameters. To help users confirm the parameters of Filter and Cleaner, it samples the raw dataset, provides the relation between the parameters of Filter operators and the filtering ratio, and presents the match cases of Cleaner operators.

4 EXPERIMENTS

We further evaluate the effectiveness of our data processing framework. Specifically, we test whether there is an enhancement in data quality and whether the enhancement contributes to improved performance of the training process of foundation models. Therefore, we conduct automated ChatGPT pairwise comparisons and end-to-end model training experiments to answer aforementioned research questions.

4.1 Automated Evaluation

Implementation Details. We adopt ChatGPT as an automated evaluation tool for data quality, utilizing its powerful ability of instruction following. We evaluate which data (*i.e.*, before and after processing) is more suitable for training LLMs. Specifically, ChatGPT is prompted to consider various dimensions such as format, fluency, coherence, and informativeness, and finally make pairwise judgments for which text is more suitable for pre-training a foundation model. Data pairs that are identical before and after processing, or exceed the ChatGPT's context length limitation are excluded.

Table 1: Results of Automated Evaluation. ChatGPT is prompted to compare the data quality of 500 data pairs before and after processing, reporting the number of win/lose/tied cases as metrics. "Win" indicates better quality than raw data.

Datasets	#Win	#Lose	#Tied
Openwebtext2	338	162	0
Wikipedia(en)	333	161	6
HackerNews	382	112	6

Datasets. We focus on three distinct subsets (Openwebtext2, Wikipedia-en, and HackerNews) of The Pile, representing different data sources (web pages, encyclopedias, and news). The details of data processing recipes can be found in our Github repository.

Results. The experimental results are presented in Table 1. On all datasets, our processed data significantly outperforms the raw dataset. Through case studies, We observe three factors that may lead to negative evaluation results. One is webpage tail information, such as "Read more on other websites" in Openwebtext2 and "References" in Wikipedia. Another one is paragraphs that are unrelated to the central semantics, such as advertisement. The third is useless characters, which may related to the webpage format.

4.2 End-to-end Evaluation

To evaluate the data quality more intuitively, we trained a GPT-2 model using the raw and refined data respectively, denoted as GPT-2-raw and GPT-2-ref. We evaluated the language modeling capabilities of LLMs on other corpora in an end-to-end manner.

Implementation Details. Due to the limitation of computation resources, we select the 110M GPT-2 [23] as our foundation model.

Table 2: Capability of Language Modeling.

	LAMBADA (PPL)	WikiText (PPL)	1BW (PPL)	CBT-CN (ACC)	CBT-NE (ACC)
GPT-2-raw	134.04	97.32	220.50	61.05	44.48
GPT-2-ref	**122.43**	**81.98**	**175.59**	**72.60**	**50.98**

As the pretraining data WebText for GPT-2 is not fully open-source, we opt for a substitute pretraining dataset CommonCrawl, which is in the same category (*i.e.*, web pages) as WebText. To train the language model, we randomly initialize the parameters and used the raw and refined versions of CommonCrawl as the training data respectively. We train 12B tokens with a learning rate of 5e-4 and context length of 1024, utilizing one RTX 3090 GPU. To speed up the training process, we adopt FP16 mixed-precision training.

Evaluation Metrics and Datasets. To align with GPT-2, we employ datasets mentioned in the research paper to evaluate the language modeling capabilities [23]. These datasets include LAMBADA [22], CBT-CN, CBT-NE [11], WikiText103 [19], and 1BW [2]. These datasets have a relatively low overlap rate with CommonCrawl [23], which allows for evaluating the foundation model's ability to acquire language modeling skills and knowledge from pretraining data effectively. CBT-CN and CBT-NE are evaluated by Accuracy (ACC), while the others are evaluated by Perplexity (PPL).

Results. The evaluation results are shown in Table 2 and Figure 2. Notably, GPT-2-ref achieves the same loss as GPT-2-raw after only 0.25M steps, whereas GPT-2-raw reaches that level after 2M steps. Across all datasets, GPT-2-ref exhibits superior performance compared to GPT-2-raw. Additionally, the PPL of GPT-2-ref on the LAMBADA and WikiText103 datasets demonstrate a noticeable trend of faster decrease compared to GPT-2-raw in the initial stage. It indicates that utilizing our proposed data processing framework enhances both the efficiency and effectiveness of training foundation models. However, as the training progress, the rate of PPL reduction for GPT-2-ref slows down, ultimately resulting in only a marginal lead over GPT-2-raw. We speculate that this is due to repeating the refined data for more than four epochs when training GPT-2-raw. We manage to ensure consistency with the number of tokens used in training GPT-2-raw, given the limited size of the refined CommonCrawl dataset. This finding also poses a requirement for future works, whereby improving data quality needs to be balanced with the retention of an adequate quantity of data, to avoid loss oscillations caused by the repetition of training corpus.

5 RELATED WORKS

Data Quality Improvement. Current researches make efforts to improve pretraining data quality through deduplication, quality filtering, and increasing diversity [29]. Lee et al. [17] indicates removing duplicate data allows models to achieve comparable or better performance with fewer training steps. Regarding quality filtering, existing works rely on handcrafted heuristics rules or constructing neural classifiers to retain high-quality data [1, 5, 7]. However, the limited efficiency of neural methods hinders their application to massive datasets. Furthermore, Gunasekar et al. [10] and Li et al. [18] select textbook-quality data from web pages to train a 1.3B model, surpassing larger LLMs in commonsense and code tasks. While these works have shown individual improvements, there is still a lack of research considering data quality enhancement from a global perspective. This calls for attention from the community towards a unified data processing framework.

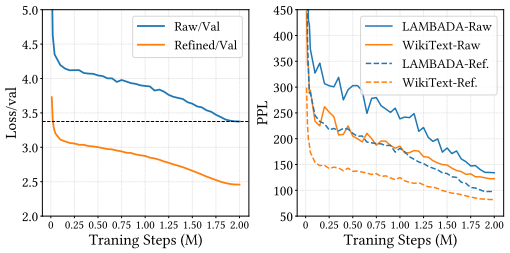

Figure 2: Performance as a function of training steps. The left illustrates the decreasing loss on the validation set, while the right indicates trends of PPL on LAMBADA and WikiText103.

Data Processing System. Numerous open-sourced datasets are accompanied by their own processing pipeline. For instance, Wenzek et al. [30] propose CCNet to filter high-quality web pages from CommonCrawl dataset. As large-scale pretraining corpus from diverse sources, BLOOM [25] and RedPajama [6] also provide details of their data construction process. While, the process are not universally applicable and can only be employed in reproducing specific dataset. The exploration of unified data processing frameworks remains scarce. As concurrent works, Zhou et al. [36] presents an interactive data management and evaluation system Oasis, whereas Chen et al. [3] similarly proposes a data processing framework data-juicer comprising multiple operators. In comparison to our paper, Oasis aims to provide a more comprehensive assessment, while its curation module is limited to filtering without cleaning capabilities. Additionally, data-juicer focuses on optimizing the fusion and rearrangement of multiple operators, significantly enhancing processing efficiency and ensuring data quality. However, our work includes additional practical and valuable functionalities, such as offering retrieval augmentation during the pretraining stage to supplement entity or topic-level knowledge for foundation models.

6 CONCLUSION

In this paper, we propose an integrated data processing framework. Users can customize the configuration of large-scale data to operate multi-level operators, enhancing data quality without coding manually. Additionally, users can employ the retrieval and evaluation modules to mitigate the issue of hallucinations during the pretraining stage. Our framework demonstrates improved data quality evaluated by ChatGPT on a subset of The Pile. Through an end-to-end evaluation approach, the foundation model trained on processed data exhibits superior generalization ability and language modeling capacity.

ACKNOWLEDGEMENTS

This research was supported by the Natural Science Foundation of China (61902209, 62377044, U2001212), and Beijing Outstanding Young Scientist Program (NO. BJJWZYJH012019100020098), Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China and Public Computing Cloud,Renmin University of China.

REFERENCES

[1] 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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[2] Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (Eds.). ISCA, 2635–2639. https://doi.org/10.21437/INTERSPEECH.2014-564

[3] Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. Data-Juicer: A One-Stop Data Processing System for Large Language Models. *CoRR* abs/2309.02033 (2023). https://doi.org/10.48550/ARXIV.2309.02033 arXiv:2309.02033

[4] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking Large Language Models in Retrieval-Augmented Generation. *CoRR* abs/2309.01431 (2023). https://doi.org/10.48550/ARXIV.2309.01431 arXiv:2309.01431

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* 24 (2023), 240:1–240:113. http://jmlr.org/papers/v24/22-1144.html

[6] Together Computer. 2023. *RedPajama: an Open Dataset for Training Large Language Models.* https://github.com/togethercomputer/RedPajama-Data

[7] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 5547–5569. https://proceedings.mlr.press/v162/du22c.html

[8] 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. 2021. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *CoRR* abs/2101.00027 (2021). arXiv:2101.00027 https://arxiv.org/abs/2101.00027

[9] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[10] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks Are All You Need. *CoRR* abs/2306.11644 (2023). https://doi.org/10.48550/ARXIV.2306.11644 arXiv:2306.11644

[11] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.02301

[12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training Compute-Optimal Large Language Models. *CoRR* abs/2203.15556 (2022). https://doi.org/10.48550/ARXIV.2203.15556 arXiv:2203.15556

[13] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 604–613.

[14] 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. *CoRR* abs/2001.08361 (2020). arXiv:2001.08361 https://arxiv.org/abs/2001.08361

[15] Denis Kocetkov, Raymond Li, Loubna Ben allal, Jia LI, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro Von Werra, and Harm de Vries. 2023. The Stack: 3 TB of permissively licensed source code. *Transactions on Machine Learning Research* (2023).

[16] 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 Sasko, Quentin Lhoest, Angelina McMillan-Major, Gérard 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 Ifeoluwa Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Alexandra Sasha Luccioni, and Yacine Jernite. 2022. The BigScience ROOTS Corpus: A 1.6TB Composite Multilingual Dataset. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[17] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 8424–8445. https://doi.org/10.18653/V1/2022.ACL-LONG.577

[18] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report. *CoRR* abs/2309.05463 (2023). https://doi.org/10.48550/ARXIV.2309.05463 arXiv:2309.05463

[19] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Byj72udxe

[20] Chenghao Mou, Chris Ha, Kenneth Enevoldsen, and Peiyuan Liu. 2023. *ChenghaoMou/text-dedup: Reference Snapshot.* https://doi.org/10.5281/zenodo.8364980

[21] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2021. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment* 2021, 12 (2021), 124003.

[22] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics. https://doi.org/10.18653/V1/P16-1144

[23] 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 (2019), 9.

[24] 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. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[25] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien

Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *CoRR* abs/2211.05100 (2022). https://doi.org/10.48550/ARXIV.2211.05100 arXiv:2211.05100

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/ARXIV.2302.13971 arXiv:2302.13971

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023). https://doi.org/10.48550/ARXIV.2307.09288 arXiv:2307.09288

[28] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2023. A Survey on Large Language Model based Autonomous Agents. *CoRR* abs/2308.11432 (2023). https://doi.org/10.48550/ARXIV.2308.11432 arXiv:2308.11432

[29] Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Data Management For Large Language Models: A Survey. *CoRR* abs/2312.01700 (2023). https://doi.org/10.48550/ARXIV.2312.01700 arXiv:2312.01700

[30] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association, 4003–4012. https://aclanthology.org/2020.lrec-1.494/

[31] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. *CoRR* abs/2305.19860 (2023). https://doi.org/10.48550/ARXIV.2305.19860 arXiv:2305.19860

[32] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. *CoRR* abs/2305.10429 (2023). https://doi.org/10.48550/ARXIV.2305.10429 arXiv:2305.10429

[33] Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. WuDaoCorpora: A super large-scale Chinese corpora for pre-training language models. *AI Open* 2 (2021), 65–68. https://doi.org/10.1016/J.AIOPEN.2021.06.001

[34] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: An Open Bilingual Pre-trained Model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

[35] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *CoRR* abs/2303.18223 (2023). https://doi.org/10.48550/ARXIV.2303.18223 arXiv:2303.18223

[36] Tong Zhou, Yubo Chen, Pengfei Cao, Kang Liu, Jun Zhao, and Shengping Liu. 2023. Oasis: Data Curation and Assessment System for Pretraining of Large Language Models. *CoRR* abs/2311.12537 (2023). https://doi.org/10.48550/ARXIV.2311.12537 arXiv:2311.12537

[37] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large Language Models for Information Retrieval: A Survey. *CoRR* abs/2308.07107 (2023). https://doi.org/10.48550/ARXIV.2308.07107 arXiv:2308.07107