On a Sustainable Training of Large Language Models for Source Code

Max Hort[1,*], Leon Moonen[1,2]

[1]Simula Research Laboratory, Oslo, Norway
[2]BI Norwegian Business School, Oslo, Norway

Abstract
Large language models (LLMs) have gained widespread attention and user adoption. These models, when trained on source code from platforms like GitHub, acquire a deep understanding of both the semantic and syntactic structures of code (i.e., code language models or CLMs). While CLMs offer tremendous assistance in software engineering tasks, their massive data requirements result in substantial energy consumption and CO_2 emissions. In this work, we aim to find solutions to help reduce the environmental impact of training CLMs. Rather than following the conventional wisdom that “more data is better”, we advocate for a refined approach to data in the training of CLMs. We propose that by intentionally decreasing training data volume while simultaneously enhancing data quality through data refinement techniques, we can reduce energy consumption while maintaining or even improving performance on software engineering tasks.

Keywords
sustainability, language model, data refinement, machine learning

1. Relevance and Novelty

Large Language Models (LLMs), like ChatGPT,[1] have garnered significant media attention and attracted a growing user base. These models can be trained on source code from platforms like GitHub,[2] enabling them to acquire both semantic and syntactic structure of code [1]. As a result, they can assist with various Software Engineering (SE) tasks, ultimately saving software developers valuable time on laborious tasks, such as bug fixing and code writing. Such Code Language Models (CLMs) rely on extensive pre-training corpora, but the sheer scale of these data and models leads to prolonged training times and high energy consumption. For instance, training a Transformer model can result in CO_2 emissions up to 17 times the average annual per-capita consumption in America [2]. More recent models, like BLOOM, trained on 46 natural and 13 programming languages, surpass this level, requiring more than a million GPU hours, 433,196 kWh of energy, and producing 81 tons of CO_2 [3, 4]. This prompts the question of whether it is necessary to use all available data for CLM training or if it is possible to reduce

ICT4S'24: The International Conference on Information and Communications Technology for Sustainability, June 24–28, 2022, Stockholm, Sweden
*Corresponding author.
maxh@simula.no (M. Hort); leon.moonen@computer.org (L. Moonen)
0000-0001-8684-5909 (M. Hort); 0000-0002-1761-6771 (L. Moonen)

[1]https://openai.com/chatgpt
[2]https://www.GitHub.com

the data volume to decrease energy consumption while maintaining high performance. Current practices are based on the assumption that more data is better [5]. However, we suggest a different approach: refining datasets to reduce data volume and energy consumption while improving data quality (e.g., removal of low-quality data). Consequently, we aim to create sustainable CLMs by intentionally decreasing training data volume, with data refinement techniques, while maintaining competitive performance on SE tasks.

2. Research Opportunities

While machine learning and LLM research is growing and sustainability is becoming a relevant factor, we believe that the research efforts on sustainability for CLMs are trailing behind. To address this, we outline three research opportunities:

Survey of refinement approaches: To gain a better understanding of the state-of-the-art for data refinement, and potential research gaps to fill, we aim to carry out a survey on data refinement techniques for CLMs. This can be seen as an addition to a recent survey performed by Albalak et al. [6], who described existing methods for data selection for language models. This study focused on LLMs trained on natural text, and techniques applied for source code play only a small role, which should be extended. Moreover, investigating further modalities, such as images, for data reduction strategies could provide a better understanding and inspiration for a more sustainable training of CLMs.

Understanding energy consumption: As a second research opportunity, we investigate data-centric factors contributing to the energy consumption in CLMs. Specifically, we will examine characteristics of data that lead to higher energy consumption during training and inference for downstream tasks. This investigation aims to provide insights into which data properties influence higher energy consumption, enabling informed decisions on which samples to remove for more sustainable training. The removal of samples that lead to high energy consumption holds promise for subsequent refinement of training data, such as focusing on Python samples if they require more energy for training CLMs compared with other programming languages.

Applying data refinement: While refinement techniques have shown success for NLP tasks and LLMs trained on text, they are relatively new and have not yet been applied to CLMs. Techniques such as distillation and coresets (a weighted subset of the datasets) [7] have proven successful in various ML tasks, yet they remain unexplored in the realm of language models, let alone CLMs [8]. These techniques can be applied for a sustainable training of CLMs.

Moreover, we aim to extend the range of data refinement strategies applied for training CLMs, for instance, by taking code quality into account (i.e., removing low-quality samples), which can improve effectiveness (e.g., achieve better performance with a lower number of samples).

3. Related Work

Training CLMs: Transformer-based CLMs are commonly trained on GitHub corpora, such as the CodeSearchNet dataset, which contains 8.5 million functions in 6 programming languages [9]. Simply randomly removing training data can negatively affect performance. However, employing systematic data refinement approaches can mitigate these effects or even improve performance by eliminating low-quality training data, including duplicated samples [10]. This approach reduces the volume of training data, thereby lowering CO_2 emissions during the training process while improving data quality.

To maintain quality control, several filtering stages have been applied in shared pre-trained models, including deduplication, filtering based the proportion of alphanumeric characters and the exclusion of code from GitHub repositories with a low number of stars [11, 12, 13, 14]. However, these filtering stages have not undergone systematic investigation (e.g., a single threshold is often chosen for filtering GitHub repositories based on stars without comparing performance before and after filtering).

Sustainable Machine Learning: The computational costs associated with training state-of-the-art Machine Learning (ML) and Deep Learning (DL) models increased by a factor of 300,000 between 2012 and 2018 [15, 16], a concerning trend highlighted in various studies [17, 18, 19]. This surge in computational costs poses challenges for researchers with limited computational resources [20], and also raises environmental concerns due to the associated CO_2 emissions[2]. Therefore, it is imperative to assess ML model quality and performance not only on metrics such as accuracy, but also on energy consumption. Techniques to improve training efficiency include quantization, model pruning, algorithm optimisation and dataset reduction [21]. For instance, Verdecchia et al. achieved a 92% improvement in energy efficiency when training DL models on structured data by reducing dataset size and number of features, [17].

Efficient and sustainable training of language models: The sustainable training of LLMs has been investigated from various perspectives, most notably: hardware design, parallelisation, batch sizing, layer dropping and data selection [22, 23, 24]. Another promising approach for promoting efficient training, with less data, is the BabyLM Challenge [25], which restricted the amount of training data for a text corpus to 10 million and 100 million words.

Acknowledgments

This work is supported by the Research Council of Norway through the secureIT project (IKTPLUSS #288787). Max Hort is supported through the Marie Skłodowska-Curie Actions Postdoctoral Fellowship programme (condenSE 101151798).

References

[1] M. Chen et al., Evaluating Large Language Models Trained on Code, 2021. doi:10.48550/arXiv.2107.03374. arXiv:2107.03374.

[2] E. Strubell et al., Energy and Policy Considerations for Deep Learning in NLP, in: Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3645–3650. doi:10.18653/v1/p19-1355.

[3] T. Le Scao et al., BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, 2022. doi:10.48550/arXiv.2211.05100. arXiv:2211.05100.

[4] A. S. Luccioni et al., Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model, 2022. doi:10.48550/arXiv.2211.02001. arXiv:2211.02001.

[5] J. Hoffmann et al., Training Compute-Optimal Large Language Models, 2022. doi:10.48550/arXiv.2203.15556. arXiv:2203.15556.
[6] A. A. et al., A survey on data selection for language models, 2024. arXiv:2402.16827.
[7] B. Mirzasoleiman et al., Coresets for Data-efficient Training of Machine Learning Models, in: Proceedings of the 37th ICML, PMLR, 2020, pp. 6950–6960.
[8] N. Sachdeva, J. McAuley, Data Distillation: A Survey, 2023. doi:10.48550/arXiv.2301.04272. arXiv:2301.04272.
[9] H. Husain et al., CodeSearchNet Challenge: Evaluating the State of Semantic Code Search, 2020. doi:10.48550/arXiv.1909.09436. arXiv:1909.09436.
[10] M. Allamanis, The Adverse Effects of Code Duplication in Machine Learning Models of Code, 2019. doi:10.48550/arXiv.1812.06469. arXiv:1812.06469.
[11] H. Touvron et al., LLaMA: Open and Efficient Foundation Language Models, 2023. doi:10.48550/arXiv.2302.13971. arXiv:2302.13971.
[12] D. Fried et al., InCoder: A Generative Model for Code Infilling and Synthesis, 2022. doi:10.48550/arXiv.2204.05999. arXiv:2204.05999.
[13] E. Nijkamp et al., CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis, 2023. doi:10.48550/arXiv.2203.13474. arXiv:2203.13474.
[14] L. B. Allal et al., SantaCoder: Don't reach for the stars!, 2023. doi:10.48550/arXiv.2301.03988. arXiv:2301.03988.
[15] G. et al., Green AI: Do deep learning frameworks have different costs?, in: Proceedings of the 44th ICSE, ACM, Pittsburgh Pennsylvania, 2022, pp. 1082–1094. doi:10.1145/3510003.3510221.
[16] R. Schwartz et al., Green AI, Communications of the ACM 63 (2020) 54–63. doi:10.1145/3381831.
[17] R. Verdecchia et al., Data-Centric Green AI: An Exploratory Empirical Study, in: International Conference on ICT for Sustainability (ICT4S), Plovdiv, Bulgaria, 2022, pp. 1–11.
[18] R. Verdecchia et al., A Systematic Review of Green AI, 2023. doi:10.48550/arXiv.2301.11047. arXiv:2301.11047.
[19] M. Hort et al., An Exploratory Literature Study on Sharing and Energy Use of Language Models for Source Code, in: 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE Computer Society, 2023, pp. 1–12. doi:10.1109/ESEM56168.2023.10304803.
[20] S. Tipirneni et al., StructCoder: Structure-Aware Transformer for Code Generation, 2022. doi:10.48550/arXiv.2206.05239. arXiv:2206.05239.
[21] S. Lei, D. Tao, A Comprehensive Survey of Dataset Distillation, 2023. doi:10.48550/arXiv.2301.05603. arXiv:2301.05603.
[22] B. Zhuang et al., A Survey on Efficient Training of Transformers, 2023. doi:10.48550/arXiv.2302.01107. arXiv:2302.01107.
[23] J. Kaddour et al., No Train No Gain: Revisiting Efficient Training Algorithms For Transformer-based Language Models, 2023. doi:10.48550/arXiv.2307.06440. arXiv:2307.06440.
[24] A. Albalak et al., A Survey on Data Selection for Language Models, 2024. arXiv:2402.16827.

[25] A. Warstadt et al., Call for Papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus, 2023. arXiv:2301.11796.