Programming Language Models in Multilingual Settings

Katzy, Jonathan

DOI
10.1145/3639478.3639787

Publication date
2024

Document Version
Final published version

Published in
Proceedings - 2024 ACM/IEEE 46th International Conference on Software Engineering

Programming Language Models in Multilingual Settings

Jonathan Katzy
J.B.Katzy@TUDelft.nl
Delft University of Technology
Delft, The Netherlands

ABSTRACT

Large language models have become increasingly utilized in programming contexts. However, due to the recent emergence of this trend, some aspects have been overlooked. We propose a research approach that investigates the inner mechanics of transformer networks, on a neuron, layer, and output representation level, to understand whether **there is a theoretical limitation that prevents large language models from performing optimally in a multilingual setting.** We propose to approach the investigation into the theoretical limitations, by addressing open problems in machine learning for the software engineering community. This will contribute to a greater understanding of large language models for programming-related tasks, making the findings more approachable to practitioners, and simply their implementation in future models.

KEYWORDS

Large Language Models, Explainable AI, Software Engineering, Code Completion, Multilingual, Programming Languages

ACM Reference Format:
Jonathan Katzy. 2024. Programming Language Models in Multilingual Settings. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3639478.3639787

1 INTRODUCTION

The rapid adaptation of Large Language Models (LLMs) to code-related tasks has shown that they are capable of solving a wide variety of programming tasks such as code completion, summarization, translation, and interpreting source code [1–7]. Many of the most modern models are also evaluated on multiple languages when testing their programming capabilities, often showing a difference in performance based on the language [8]. Although this performance gap is known, little research has been done on the source of this discrepancy.

In the context of multilingual models, research indicates that training with a mix of languages may be beneficial, especially for resource-scarce languages [4, 8, 9]. Yet, these assertions often lack specific references; they are based on theories about the models' capacity to generalize across languages, form abstract concepts internally, and apply these concepts to various languages.

We propose to approach this problem from two perspectives. We will first analyze the internal states of the model and how shared information may have an impact on performance. Then we will analyze the representations of outputs generated by models. Beginning with an overview of differences between settings and languages [10], and expanding the work to identify which characteristics of languages contribute the most to differences in representations.

More specifically we address the hypothesis **There is a theoretical limitation that prevents large language models from performing optimally in a multilingual setting**, by answering four Research Questions (RQ) that align with the areas of interest as follows:

- **RQ1** Do large language models trigger activation in distinct areas of the model when processing various programming languages?
- **RQ2** Does the inclusion of multiple languages in the training data lead to negative interference when a large language model generates predictions?
- **RQ3** Do the variations in token representations produced by models across different languages adversely affect their multilingual performance?
- **RQ4** What characteristics of languages are important to evaluate when selecting languages for pre-training and fine-tuning?

2 RELATED WORK

We categorize the related work into two primary sections. First, we will look at the work being done on analyzing the inner states of transformer models, focussing on the behavior of the neuron activation and the attention mechanism. Secondly, we will look at the research that has been done on the representation of languages and how it relates to a model's ability to learn between languages.

The analysis of the inner states of a model is focussed on two main aspects. In the first scenario, the activation of neurons has been analyzed to obtain information about dedicated neurons for certain tasks [11, 12]. This explains the contributions of each neuron in the final output. The second approach focusses more on the overall contributions of layers to the internal representations the model uses for reasoning called the residual stream [13]. Analysis of the residual stream can simplify explainability [14], the identification of world representations [15], and even the editing of knowledge that is currently present in networks [16].

Analyzing the representations of models has been an area that has had research since the start of the adoption of neural models in AI. The relationship to the loss surfaces has been analyzed to understand whether a model has adequately fit the data [17]. This has led to papers that analyze the smoothness of loss functions as a

ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0502-1/24/04.
https://doi.org/10.1145/3639478.3639787

metric for a model's ability to generalize [18] and optimize models to find flat regions on the loss landscape [19], while others attempt to measure the quality of local optima [20].

In addition to focussing on the relation between smoothness and a network's ability to generalize, the loss surface has also been analyzed with respect to the connectivity of multiple optima [21], where it was shown that local minima are connected along parametric curves on the loss surface [22].

W3 have seen initial success in analyzing the differences in representations found between common tokens in different programming languages. We describe an initial approach to comparing token representations using a cosine similarity in our NIER paper [10]. This work has shown that, while many languages are similar to each other, depending on the expected use case of the language, the representation of the same token can vary consistently across languages.

3 APPROACH

From related and previous work, we have seen that there is a gap in the knowledge about how LLMs work, how they generate outputs, and how the training setups can contribute to the difference in performance between languages. To fill these knowledge gaps, we will be working at the intersection of Explainable AI, AI for Software Engineering, and Deep Learning. Our research examines the performance of LLMs from two angles. First, we explore the models' behavior in terms of their internal knowledge representations and neuron activations. Secondly, we study the characteristics of the loss surface, focusing on how its local geometries may present challenges to the multilingual learning capabilities of neural models.

3.1 Internal Analysis

For the internal analysis of the network, our objective is to understand which areas of the model are responsible for certain outputs. In our approach, we begin by identifying situations where we want to know if there is a difference in the behavior of the model. For these situations, we choose areas that are active areas of interest in the ML4SE community. Candidate situations are behaviors when working in common code idioms, differences between out-of-context library predictions versus in-context identifier predictions, and network activity when recalling data from the training set versus generating code. This will help to answer **RQ1** and **RQ2**.

To analyze the behavior of models, we can use a tuned lens [23] to analyze intermediate representations in a model. This gives us answers about which layers are most active when working with certain tokens and gives insight into how much the residual stream of the network is updated at each layer. Furthermore, we can save the attention activation of the models in different situations and analyze which model neurons are responsible for which outputs.

This will give us information about the states of the models in different situations and in different languages, which will allow us to compare model behavior across languages and gain insight into the behavior, which may explain the performance gap between languages in multilingual models.

3.2 Representation Analysis

When looking at the representations of models, we shift our attention from how a model creates the output, to what the model has learned. The final representation of tokens has been shown to contain a world model of the knowledge the model has gained, so it can be used to analyze the differences in knowledge representations between languages in models [24]. We intend to use the knowledge gained from these experiments to answer both **RQ3** and **RQ4**.

We can analyze the relationships between representations and use them as sample points on the loss surface used to train the model by taking the representations that the model produces when generating an output. We can then use these representations to identify problematic situations that cause higher error rates in models, which can explain whether a model with the given setup is able to be trained in a multilingual setting, or if the setup creates a local optimum the model cannot escape while training, preventing the model from learning other languages effectively. This is especially important for languages that share a large number of their tokens, as we postulate that this will lead to local optima from which the model may not escape. Being able to understand the scenarios in which these local optima occur will allow us to adapt the training of models to allow for multilingual performance by changing the order of training data, limiting imbalances between languages in the training data, or introducing language-specific tokens to prevent negative interference from other languages while training.

4 EXPECTED CONTRIBUTIONS

The primary goal of this research is to understand the reasons behind the differences in performance across different languages when using LLMs. Our goal for this research is not only to gain an understanding of the behavior of these models but also to create experiments that can give clear and actionable insight to developers working with LLMs. This aims to aid the development of a wider variety of LLMs without being bound to a single architecture.

In the wider adaptation of LLMs, the need to explain AI to researchers outside of the AI community is more prudent, as the models are gradually adopted into daily life. Having clear methods that can explain the reasons for predictions from a model can help increase trust and start to address issues that may arise with the originality of generated outputs and the implications of copying personal data.

The main contributions of the research will be approachable experiments that give actionable insights to developers who are adapting and creating LLMs for future research and the development of tools for real-world use cases.

5 CONCLUSION

We have shown that there is a research gap when it comes to identifying the limitations of models in the multilingual setting. We propose a research setup that analyzes the internal state of the models during inference times, as well as an analysis of the loss surface of the models. We use both of these approaches to identify and explain the limitations while aiming to create approachable experiments that can aid in the development of future large language models.

REFERENCES

[1] Github. Github copilot. https://github.com/features/copilot [Accessed: 2023].

[2] Maliheh Izadi, Roberta Gismondi, and Georgios Gousios. Codefill: Multi-token code completion by jointly learning from structure and naming sequences. In *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, page 401–412, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online, November 2020. Association for Computational Linguistics.

[4] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.

[5] Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, 2021.

[6] Ali Al-Kaswan, Toufique Ahmed, Maliheh Izadi, Anand Ashok Sawant, Prem Devanbu, and Arie van Deursen. Extending source code pre-trained language models to summarise decompiled binaries. In *Proceedings of the 30th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2023.

[7] Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. KDD '23, page 5673–5684, New York, NY, USA, 2023. Association for Computing Machinery.

[8] Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10, 2022.

[9] Alexey Svyatkovskiy, Shao Kun Deng, Shengyu Fu, and Neel Sundaresan. Intellicode compose: Code generation using transformer. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 1433–1443, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Jonathan Katzy, Maliheh Izadi, and Arie van Deursen. On the impact of language selection for trainingand evaluating programming language models. In *2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 2023.

[11] Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*, 2023.

[12] Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale. *arXiv preprint arXiv:2305.19911*, 2023.

[13] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[14] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023.

[15] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

[16] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[17] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[18] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[19] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[20] Serguei Barannikov, Daria Voronkova, Ilya Trofimov, Alexander Korotin, Grigorii Sotnikov, and Evgeny Burnaev. Topological obstructions in neural networks learning. *arXiv preprint arXiv:2012.15834*, 2020.

[21] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

[22] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

[23] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.