Giagnorio, A., Martin-Lopez, A., Bavota, G.: Enhancing Code Generation for Low-Resource Languages: No Silver Bullet. In: Burgueño, L. (ed.) Actas de las XXIX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2025). Sistedes (2025). https://hdl.handle.net/11705/JISBD/2025/3

Enhancing Code Generation for Low-Resource Languages: No Silver Bullet*

Alessandro Giagnorio[1], Alberto Martin-Lopez[1], and Gabriele Bavota[1]

Software Institute – USI Università della Svizzera italiana, Switzerland
{alessandro.giagnorio,alberto.martin,gabriele.bavota}@usi.ch

Keywords: Code Generation, Low-Resource Languages
Published in: ICPC 2025 - Proceedings of 33rd IEEE/ACM International Conference on Program Comprehensions, in press, 2025
Impact Factor: ICORE Rank A
Preprint: https://doi.org/10.48550/arXiv.2501.19085

Abstract. The advent of Large Language Models (LLMs) has significantly advanced the field of automated code generation. LLMs rely on large and diverse datasets to learn syntax, semantics, and usage patterns of programming languages. For low-resource languages (i.e., niche programming languages characterized by the scarcity of training data), the limited availability of such data hampers the models' ability to generalize effectively, resulting in poorer code generation performance as compared to high-resource languages. For this reason, there is a quest for techniques able to close this performance gap. We present an empirical study investigating the effectiveness of several approaches for boosting LLMs' performance on low-resource languages, namely: (i) a classic fine-tuning, which is however capped in size by the scarcity of training data; (ii) three variants of in-context learning, with prompts crafted to provide the LLM with additional information about the low-resource language (e.g., few-shot examples showcasing features of the targeted language); and (iii) a pre-training objective teaching the model how to translate between high- and low-resource languages. The context of our study are two low-resource languages (R and Racket) and six LLMs having different architectures and sizes. Our findings reveal that a fine-tuning is usually the best choice for smaller LLMs, possibly due to the fact that even a small dataset is sufficient to train their limited number of parameters. With the increase in size of the models, in-context learning becomes more and more effective, representing a safe and cheap bet (i.e., it always helps, but with different magnitudes). Differently, very large LLMs may deteriorate their performance on low-resource languages when fine-tuning is performed, possibly due to the lack of enough data needed to effectively update their weights.

* We acknowledge the financial support of the Swiss National Science Foundation for the PARSED project (SNF Project No. 219294).

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivatives 4.0 International" license.