Ministry of Education of the Republic of Belarus
Educational institution
Belarusian State University of
Informatics and Radioelectronics

UDC: 004.056.55

ZHONG
Wu

METHODS AND ALGORITHMS OF AUTOMATIC CODE GENERATION

Abstract
for a Master's Degree
in the Specialty 7-06-0612-03 Information Management Systems

_____

Rybak, V. A
Candidate of Technical Sciences,
Associate Professor
Rybak Victor Alexandrovich

_____

Minsk 2025

INTRODUCTION

In recent years, the field of software development has witnessed a transformative shift toward automation, driven by advancements in artificial intelligence (AI) and machine learning (ML). One of the most promising areas of research within this domain is code generation, which involves using algorithms to automatically produce code based on human-defined specifications or examples. This capability not only accelerates the development process but also helps mitigate common programming errors, thereby enhancing code quality and maintainability.

The rapid evolution of programming languages and paradigms has introduced complexities that challenge traditional coding practices. As a result, developers increasingly seek tools that can assist in automating mundane and repetitive coding tasks. This demand has led to the emergence of various code generation models, particularly those based on deep learning techniques. Among these, the codegen model stands out due to its ability to understand and generate code across multiple programming languages, making it a versatile tool for developers.

Despite the progress made, many existing models face limitations in accurately generating high-quality code that meets specific user requirements. Fine-tuning pre-trained models has emerged as a promising solution to enhance their performance in specialized tasks. In this context, Low-Rank Adaptation (LoRA) serves as an innovative approach to fine-tune large language models efficiently, enabling them to adapt to new tasks with minimal computational resources.

This research aims to explore the use of LoRA for fine-tuning the codegen model, specifically leveraging the xlcost-text-to-code dataset. By focusing on this dataset, the study seeks to improve the model's ability to generate code that aligns more closely with developer expectations. Through rigorous evaluation and analysis, this work endeavors to contribute to the growing body of knowledge in the field of automated code generation, ultimately enhancing the tools available to software developers.

Recent advancements in AI-driven code generation have significantly accelerated software development workflows, yet practical adoption remains constrained by the computational demands of large language models. This study demonstrates how Low-Rank Adaptation (LoRA) fine-tuning of the CodeGen-350M model achieves a 62.5% improvement in functional correctness, elevating the Pass@1 rate from 4.88% to 7.93% on the HumanEval benchmark while maintaining manageable resource requirements (13.6GB VRAM on RTX 2060 SUPER). The success of this parameter-efficient approach suggests promising applications in scenarios requiring: (1) rapid prototyping where lightweight model adaptation is preferred, and (2) educational contexts where moderate accuracy gains with constrained hardware could assist coding instruction. Our results establish that even

modest-scale models can achieve meaningful performance improvements through targeted architectural modifications, providing a practical pathway for organizations with limited computational resources to leverage AI-assisted code generation.

The following sections will detail the methodology employed in this research, present the results of the experiments conducted, and discuss the implications of the findings for future developments in code generation technologies.

GENERAL DESCRIPTION OF WORK

Relevance of the subject

The work corresponds to paragraph 6 «Ensuring the safety of man, society, state» of the State Program of innovative development of the Republic of Belarus for 2021–2025. The work was carried out in the educational institution Belarusian State University of Informatics and Radioelectronics.

The aim and tasks of the work

The primary aim of this research is to enhance the performance of the codegen model in code generation tasks through LoRA fine-tuning. The specific tasks include:

1.Fine-tuning the model using the xlcost-text-to-code dataset.

2.Evaluating the performance of the fine-tuned model across various code generation tasks.

3.Analyzing the quality and accuracy of the generated code and providing suggestions for improvement.

Personal contribution of the author

As the author of this research, I have contributed to several key aspects:

The content of this research reflects the personal contributions of the author, which include the scientific formulation of the LoRA-based fine-tuning approach, the selection and preprocessing of the xlcost-text-to-code dataset, and the implementation and monitoring of the training process. Additionally, the author designed and conducted experiments to evaluate the effectiveness of the generated code, processed and analyzed the results, and formulated key conclusions. The task definition and discussion of the findings were carried out in collaboration with the academic supervisor, Rybak Victor Alexandrovich Associate Professor at the Department, Candidate of Technical Sciences.

Testing and implementation of results

In this study, I will test and implement the results through the following methods: Designing a series of benchmark tests to evaluate the model's performance before and after fine-tuning. Comparing the effectiveness and efficiency of the

generated code using standardized metrics.

Author's publications

According to the results of the research presented in the dissertation, _2_ author's works was published, including: _2_ articles and abstracts in conference proceedings.

Plagiarism

An examination of the dissertation «Methods and algorithms of automated code generation» by Zhong Wu carried out for the correctness of the use of borrowed materials using the network resource «Antiplagiat» (access address: https://antiplagiat.ru) in the online mode __. . __. As a result of the verification, the correctness of the use of borrowed materials was established (the originality of the thesis is __%)

SUMMARY OF WORK

The introduction addresses the problems of automating code generation in software development, highlighting the transformative shift driven by advancements in artificial intelligence (AI) and machine learning (ML). It emphasizes the challenges faced by existing models in generating high-quality code that meets specific user requirements and introduces Low-Rank Adaptation (LoRA) as an innovative solution for fine-tuning pre-trained models like CodeGen. The research aims to enhance the model's performance using the XLCoST-text-to-code dataset, contributing to the broader field of automated code generation.

The general description of work shows the connection between the work and the priority areas of scientific research, the aim and tasks of the research, the personal contribution of the applicant for a scientific degree, the approbation of the dissertation results.

The first chapter reviews the relevant literature in the field of code generation, covering the definition, significance, and applications of code generation, as well as the role of machine learning, including different model architectures such as RNNs and Transformers. It discusses the evolution of code generation techniques and highlights the challenges and future research directions within this domain.

The second chapter details the research design and methodology, including the selection and preprocessing of datasets, the choice of models, and the fine-tuning strategy employed. It specifically describes how Low-Rank Adaptation (LoRA) is utilized to enhance the performance of the code generation model across multiple programming languages.

The third chapter presents the experimental results of the model fine-tuned with LoRA, showcasing its performance across various tasks. It analyzes the training convergence, computational efficiency, and optimization behavior, comparing the model's performance before and after fine-tuning, and evaluating code generation quality through specific metrics such as Pass@k.

The fourth chapter discusses the implications of the experimental results, emphasizing the effectiveness of LoRA, the hardware constraints, and their impact on model performance. It offers suggestions for future research, including resource-aware optimizations and data augmentation, aiming to further improve the performance and applicability of code generation models.

CONCLUSION

This research advances the critical area of code generation through parameter-efficient optimization of the CodeGen model using Low-Rank Adaptation (LoRA). By fine-tuning the CodeGen-350M architecture on the xlcost-text-to-code dataset, we achieved a 62.5% improvement in functional correctness, elevating the Pass@1 rate from 4.88% to 7.93% on the HumanEval benchmark while maintaining practical deployment feasibility (13.6GB VRAM on RTX 2060 SUPER GPUs). The results demonstrate that LoRA effectively adapts moderate-scale models to specialized coding tasks without full parameter updates, though the absolute performance remains below state-of-the-art levels. These findings establish an important baseline for resource-constrained scenarios where lightweight adaptation is preferred over computationally intensive fine-tuning. Future work should investigate: (1) higher-rank LoRA configurations for improved accuracy, (2) dynamic rank allocation across network layers, and (3) integration with retrieval-augmented generation to enhance output quality. This direction promises to bridge the gap between computational efficiency and generation capability, ultimately delivering more accessible AI-assisted programming tools.

LIST OF AUTHOR'S PUBLICATIONS

1-A. Wu Z., & Rybak, V. A. (2024). Evaluation methods for code generation models. Information technologies and management : proceedings of the 60th scientific conference of graduate students, undergraduates and students, Minsk, April 22-26, 2024

2-A. Wu, Z. (2025). the role of machine learning in automating code generation. International Center for Scientific Partnership "New Science", May 08, 2025.