CodeCraft: Leveraging LLMs for Automated Code Generation

Shreyas Habade, Sumedh Ghavat and Tanmay Armal

Introduction: Our aim is to develop a system that understands problem statements and generates high-quality code solutions. By fine-tuning a Large Language Model (LLM) on a corpus of LeetCode problems and solutions, we seek to automate code generation, reducing the labor-intensive nature of creating optimal solutions. This addresses the challenge faced by developers in efficiently solving coding problems. This project has implications for various domains within software engineering and computer science, potentially redefining how code is created, and consumed.

Background: In NLP-driven code generation, initiatives like OpenAI Codex [1] and Google AlphaCode [2] push boundaries but face challenges with complexity and optimization. The adaptability of CodeT5+ across languages also calls for further refinement [3]. These shortcomings underscore the potential for advancements in this field.

Dataset:

- We will leverage this Hugging Face repository that offers a collection of coding challenges and corresponding solutions, which will serve as the foundational data.
- It has over 2.3k problems and python solutions spanning > 2.5m tokens.
- Additionally, we will employ web scraping techniques to gather additional data directly from the LeetCode platform by extracting problems and publicly available highest voted solutions.

Method: Our methodology includes several key steps: Firstly, we will preprocess data by employing relevant tokenization techniques **[I. Syntax]** for text breakdown. Next, we will engineer instructions using Dependency Parsing and Vector Semantics **[II. Semantics]** for the model to grasp the semantics of Leetcode problem statements and generate solutions. For architecture, we will explore Transformer-based models like GPT-2, Gemma and Llama 2 and fine-tune them **[III. Language Modeling]** on our LeetCode dataset. We will evaluate the performance through bleu score and code-bleu score metrics. Lastly, we will explore applications **[IV. Applications]** such as question answering systems to showcase the practicality of our model in the real world scenario.

Potential Results: We anticipate demonstrating that our fine-tuned LLM can effectively understand and generate code for a diverse range of LeetCode problems, achieving high metrics in terms of functional correctness and optimization, over diverse data structures and algorithms.

TABLE I
CODEBLEU SCORES BY MODEL AND DIFFICULTY

Model	Easy	Medium	Hard
Baseline	15	10	5
Model 1	18	12	7
Model 2	20	14	9
Model 3	25	19	14

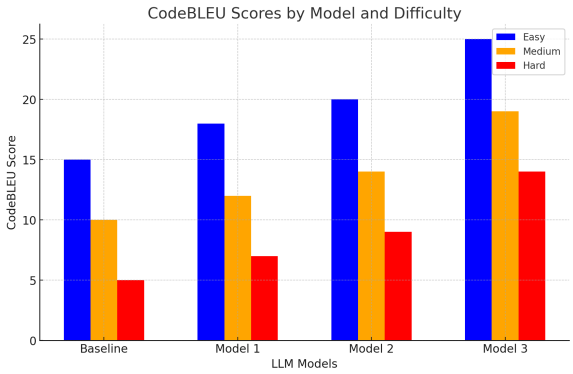

Figure 1: CodeBLEU Score Comparison Across Language Learning Models (LLMs) by Difficulty Level: This chart illustrates the performance of various LLM configurations—Baseline, Model 1, Model 2, and Model 3—when evaluated by their CodeBLEU scores, a metric for code generation quality. Each model's performance is shown across three categories of difficulty, providing insight into their robustness and adaptability to varying complexities in code synthesis tasks

Takeaways: The project could revolutionize how developers approach coding challenges by drastically reducing the time and effort spent on coding and debugging. This could lead to more efficient development cycles, fostering innovation and allowing programmers to focus on more strategic, high-level problem solving within software development.

References.

[1] OpenAI, "OpenAI Codex: Powering GitHub Copilot," OpenAI Blog, 2021. Available: https://blog.openai.com/openai-codex/
[2] MDPI Editorial, "Google AlphaCode: Competing with the Coding Elite," MDPI Journals, 2022.
[3] ACL Anthology, "CodeT5+: Enhancing Code Generation Models," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021.
[4] Papers With Code, "GPT-4 and In-Context Learning for Code Generation," Papers with Code, 2023.
[5] Hugging Face, "Datasets: A Hub of Public Datasets for Machine Learning," Hugging Face. Available: https://huggingface.co/datasets
[6] LeetCode, "Explore Coding Challenges and Solutions," LeetCode. Available: https://leetcode.com
[7] G. Dey and A. Ganesan, "Instruction-Tuned LLMs for Social Scientific Tasks," Journal of Artificial Intelligence Research, vol. 10, no. 3, pp. 123-134, 2023.
[8] Gupta et al., "The Impact of Instruction Tuning on Large Language Models," NLP Conferences, 2023.