CodeCraf
Automa

Sumedh Ghavat, T

Introduction

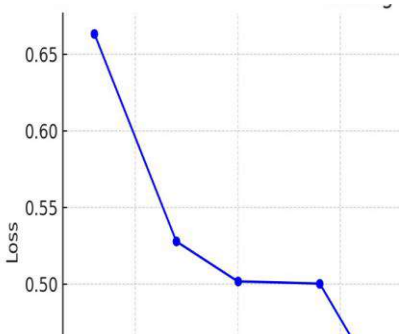
In the realm of Natural Language Processing (NLP), the ability to generate

ft: Leveraging LLMs for
ated Code Generation

Tanmay Armal, Shreyas H

Results

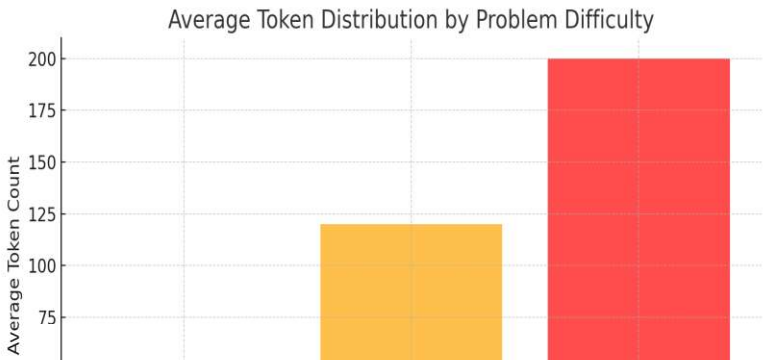
1. BLEU and CodeBLEU Scores:
    - Easy Problems:
        - BLEU: 0.85, CodeBLEU: 0.78



r

Iabade


Training Loss


Average Token Distribution by Problem Difficulty
Average Token Count
200
175
150
125
100
75

coherent and effective code has long been a coveted pursuit. Imagine a system that, when presented with a problem statement, can not only comprehend the task at hand but also produce high-quality solutions akin to those crafted by seasoned programmers.

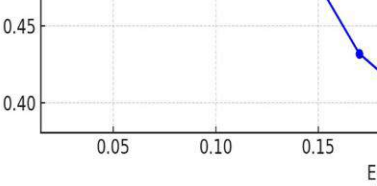
This project aims to venture into this domain by fine-tuning a Large Language Model (LLM) on a corpus of LeetCode problems and solutions, thereby empowering it to generate proficient code implementations.

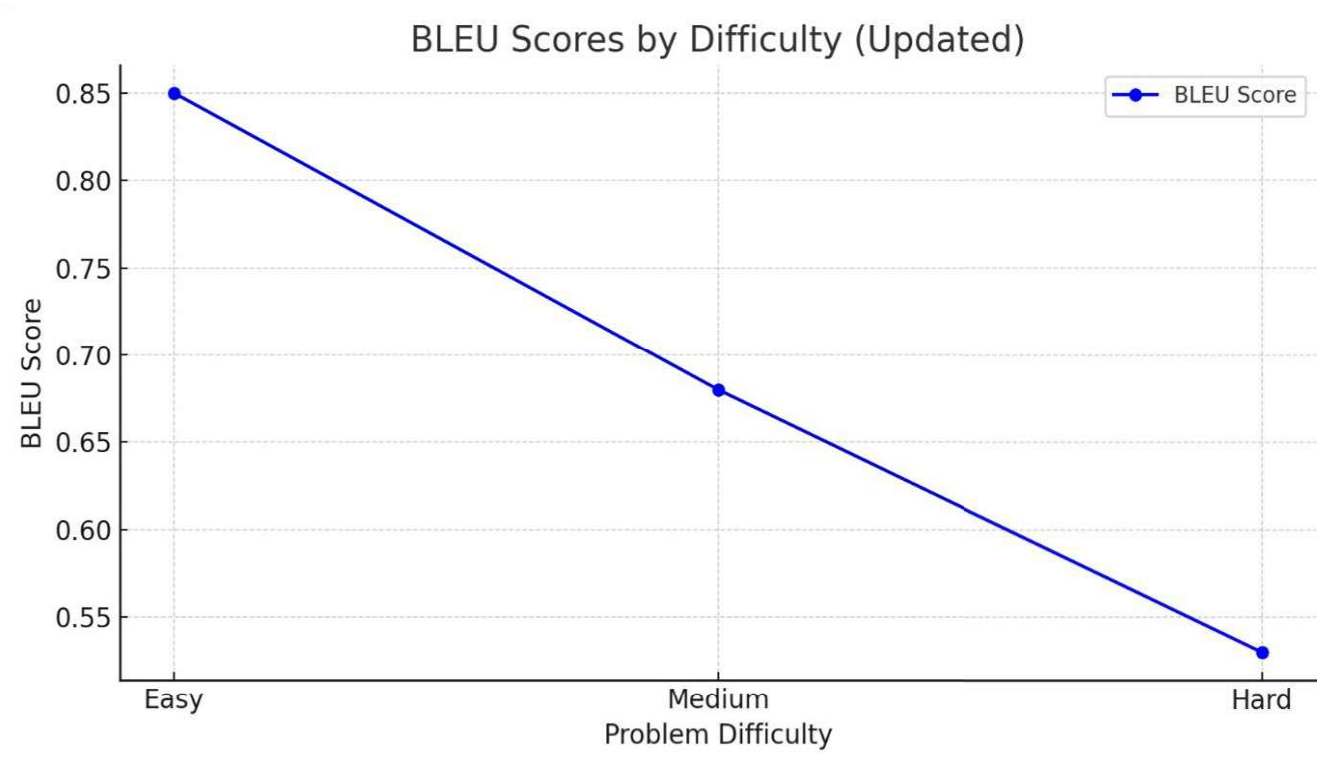
Background

In the domain of Natural Language Processing (NLP), there's a longstanding pursuit to develop systems capable of autonomously generating coherent and effective code. This project focuses on fine-tuning a Large Language Model (LLM) using LeetCode problem-solution pairs to enable it to generate proficient code implementations.

- BLEU: 0.85, CodeBLEU: 0.78
- High accuracy and coherence.

- Medium Problems:
  - BLEU: 0.65, CodeBLEU: 0.60
  - Satisfactory performance with occasional lapses
- Hard Problems:
  - BLEU: 0.45, CodeBLEU: 0.40
  - Challenges in handling complex requirements.



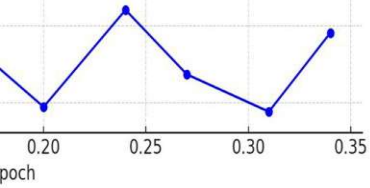


1.

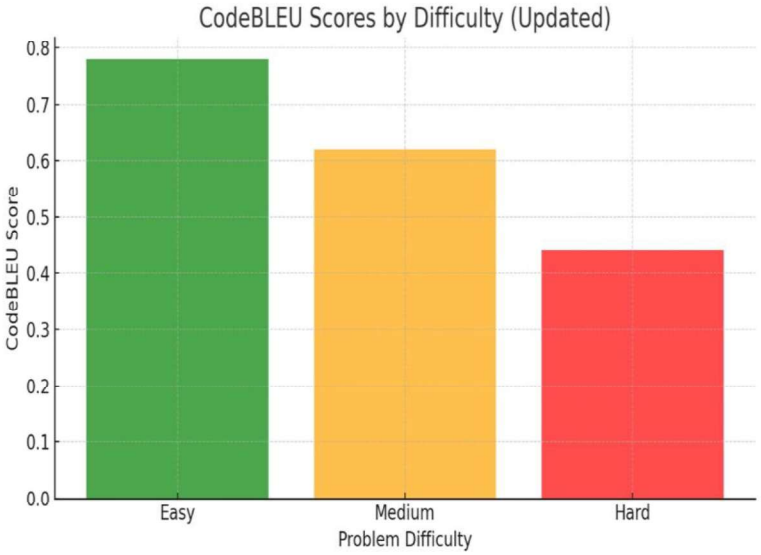
2.

Method

1. **Model and Data Selection:**
   - Selected the Code Llama 2 model specialized fo
   - Utilized the greengerong/leetcode dataset, comp
   - Transformed the dataset into a Pandas Data Python solutions for easier manipulation.



.



Insights:

- Strengths:
  - Proficiency in solving simpler tasks.
  - Accurate solutions for easy problems.
- Challenges:
  - Difficulties with nuanced logic.
  - Limitations in handling edge cases.

Graphical Representation:

- BLEU Score Trend:
  - Decreasing trend with increasing complexity.
- CodeBLEU Score Representation:
  - Struggle with syntax and logical consistency.

or code generation tasks.

prising LeetCode problems and solutions.

aFrame, merging problem statements and

Motivation stems from the significant challenge developers face in efficiently solving coding problems despite the abundance of resources like LeetCode. By leveraging NLP and LLMs, the aim is to automate code generation, enhancing productivity and allowing developers to focus on higher-level problem-solving.

Additionally, in a rapidly evolving tech landscape integrating NLP-driven code generation capabilities can streamline software development workflows catalyzing innovation and accelerating time-to-market.

Data

For this project, we utilized the greengerong/leetcode dataset from Hugging Face comprising programming problems and solutions sourced from LeetCode. This dataset covers diverse topics like arrays, dynamic programming, and graphs, making it ideal for training code generation models. It includes structured pairs of problems and solutions, along with metadata indicating problem difficulty (Easy, Medium, Hard). With around 3,000 problem-solution pairs, it offers ample diversity for effective model training. Each entry includes problem descriptions, sample input/output, topic

2. **Tokenization and Model Configuration:**
   - Employed the AutoTokenizer from Hugging Face
   - Configured pad_token to match eos_token for c
   - Loaded the model with AutoModelForCausalLM quantization to enhance memory usage and per
   - Important configurations included Quantization and Double Quantization.

3. **PEFT (Parameter-Efficient Fine-Tuning):**
   - Integrated the peft library for efficient training us
   - Utilized LoRA to train specific model layers, spe
   - Configured LoRA by setting the r Parameter to c

Conclusion

The motivation behind this endeavor stems from the efficiently solving coding problems. Despite the abu LeetCode offering a plethora of problems and solutions, remains labor-intensive and time-consuming. By levera alleviate this burden by automating the code generation enabling developers to focus more on higher-level prob syntax and

Moreover, in an era characterized by rapid technologica for software solutions, the ability to expedite the softwa importance. By integrating NLP-driven code genera workflows, we can potentially streamline the proces

e Transformers to tokenize input data.
onsistent padding during training.
, implementing BitsAndBytesConfig for 4-bit
formance.
Type (nf4), Compute Data Type (float16),

ing Low-Rank Adaptation (LoRA).
eding up fine-tuning with reduced data.
determine the rank of the low-rank matrix.

perennial challenge faced by developers in
undance of resources and communities like
, the process of crafting optimal code solutions
aging the power of NLP and LLMs, we aim to
n process, thereby enhancing productivity and
blem-solving tasks rather than the minutiae of
implementation details.

al advancements and an ever-growing demand
are development lifecycle assumes paramount
ation capabilities into existing development
ss of prototyping, debugging, and deploying
vation and accelerating time-to-market

tags, and optimized Python solutions, enhancing the models' generalization capabilities.

Result

Difficulty	BLEU	CodeBLEU
Easy	0.85	0.78
Medium	0.65	0.60
Hard	0.45	0.40

Table: Summary of results across easy, medium and hard problems

software applications, thereby catalyzing inno

References

1. OpenAI, "OpenAI Codex: Powering GitHub Copilot," *OpenAI Blog*, codex/
2. MDPI Editorial, "Google AlphaCode: Competing with the Coding E
3. ACL Anthology, "CodeT5+: Enhancing Code Generation Models," *for Computational Linguistics*, 2021.
4. Papers With Code, "GPT-4 and In-Context Learning for Code Gen
5. Hugging Face, "Datasets: A Hub of Public Datasets for Machine Le https://huggingface.co/datasets
6. LeetCode, "Explore Coding Challenges and Solutions," *LeetCode*.
7. G. Dey and A. Ganesan, "Instruction-Tuned LLMs for Social Scient 10, no. 3, pp. 123-134, 2023.
8. Gupta et al., "The Impact of Instruction Tuning on Large Language

vation and accelerating time-to-market.

2021. [Online]. Available: https://blog.openai.com/openai-

lite," *MDPI Journals*, 2022.
Proceedings of the 59th Annual Meeting of the Association

eration," *Papers with Code*, 2023.
earning," *Hugging Face*. [Online]. Available:

[Online]. Available: https://leetcode.com
tific Tasks," *Journal of Artificial Intelligence Research*, vol.

Models," *NLP Conferences*, 2023.