PythonSaga: Redefining the Benchmark to Evaluate Code Generating LLMs

Ankit Yadav, Himanshu Beniwal, Mayank Singh
Discipline of Computer Science and Engineering, Indian Institute of Technology Gandhinagar
{ankityadav, himanshubeniwal, singh.mayank}@iitgn.ac.in

Abstract

Driven by the surge in code generation using large language models (LLMs), numerous benchmarks have emerged to evaluate these LLMs capabilities. We conducted a large-scale human evaluation of *HumanEval* and *MBPP*, two popular benchmarks for Python code generation, analyzing their diversity and difficulty. Our findings unveil a critical bias towards a limited set of programming concepts, neglecting most of the other concepts entirely. Furthermore, we uncover a worrying prevalence of easy tasks that can inflate model performance estimations. To address these limitations, we propose a novel benchmark, **PythonSaga**, featuring 185 hand-crafted prompts in a balanced representation of 38 programming concepts across diverse difficulty levels. The robustness of our bench-mark is demonstrated by the poor performance of existing Code-LLMs.

1. Introduction

Gap: Existing code generation benchmarks for LLMs like *HumanEval* and *MBPP* exhibit a critical bias, favoring a narrow set of programming concepts and overlooking a broader range.

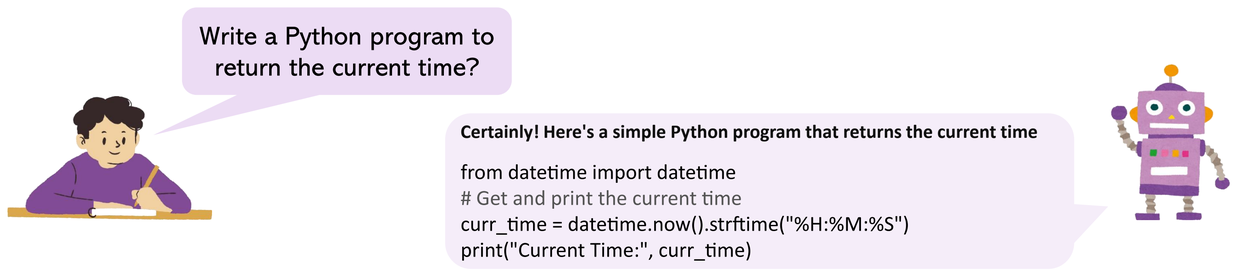

Figure 1: Illustration of a conversation wherein a human provides an input description, & a Code-LLM generates expected Python code

Motivation: We developed *PythonSaga*, a new benchmark with 185 hand-crafted prompts, ensuring a balanced representation of 38 diverse programming concepts across various difficulty levels. Our benchmark addresses the limitations of existing benchmarks and reveals the inadequate performance of current Code-LLMs when assessed on a more comprehensive range of tasks.

Research Questions:

- **Q1:** *Have Code-LLMs developed the generalization ability to solve any programming problem?*
- **Q2:** *What specific programming concepts pose challenges for Code-LLMs, limiting their problem-solving capabilities?*
- **Q3:** *How diverse are the programming concepts covered in current code generation benchmarks, and do they adequately represent various difficulty levels?*
- **Q4:** *How effective are Code-LLMs when evaluated on a more balanced and diverse set of programming concepts?*

Contributions:

1. Introduced a hierarchical classification of programming concepts into basic, intermediate, and advanced levels. *(Table 1)*
2. Evaluated HumanEval and MBPP, exposing their bias towards a limited set of concepts and overly simple problems. *(Section 2)*
3. Proposed PythonSaga, a new benchmark offering balanced coverage of 38 programming concepts across three difficulty levels. *(Section 3)*
4. Demonstrated that existing Code-LLMs perform poorly on PythonSaga, revealing disparities in their ability to handle different concepts and difficulties. *(Section 4)*

Basic	Intermediate	Advance
Function	OOPS	Trie
Mathematics	Stack	Tree
File Handling	Sorting	Heap
Basic Libraries	Hashing	Graph
Error Handling	Searching	Matrix
Input and Output	Recursion	Max Flow
In-Built Functions	Linked List	Disjoint Set
Pattern Replication	Bit Manipulation	Backtracking
Basic Data Structures	Queue & Dequeue	Greedy Search
Variable & Data Types	Regular Expression	Advanced OOPs
Control Flow & Conditions	Circular & Doubly Linked List	Context Managers
	Advanced String Manipulation	Divide and Conquer
		Dynamic Programming
		Closures and Decorators
		Concurrency and Parallelism

Table 1: A hierarchy of 38 programming concepts categorized into basic, intermediate, and advance categories.

2. Human Annotation Experiments

Programming Concept Diversity: Five key concepts—Mathematics, Control Flow, Basic Data Structures, Variables, and In-Built Functions—dominate HumanEval *(72.1%)* and MBPP *(77.3%),* while 14 important concepts like OOP and Graphs are missing. Basic problems make up *78%*, with only *~3%* advanced. *(Fig. 2 & 3)*

Difficulty Level: Most problems in HumanEval (84.8%) and MBPP (89.6%) are Easy, with no Hard problems in MBPP, suggesting that both benchmarks favor simpler problems, potentially overstating Code-LLMs' generalization abilities.

3. PythonSaga: A New Benchmark for Code Generation Models

- PythonSaga provides 185 diverse prompts across 38 programming concepts, with balanced difficulty levels *(20% Easy, 40% Medium, 40% Hard),* addressing gaps in existing benchmarks.
- Problems are manually rephrased for clarity and realism, challenging Code-LLMs to go beyond pattern recognition and better understand real-world contexts.

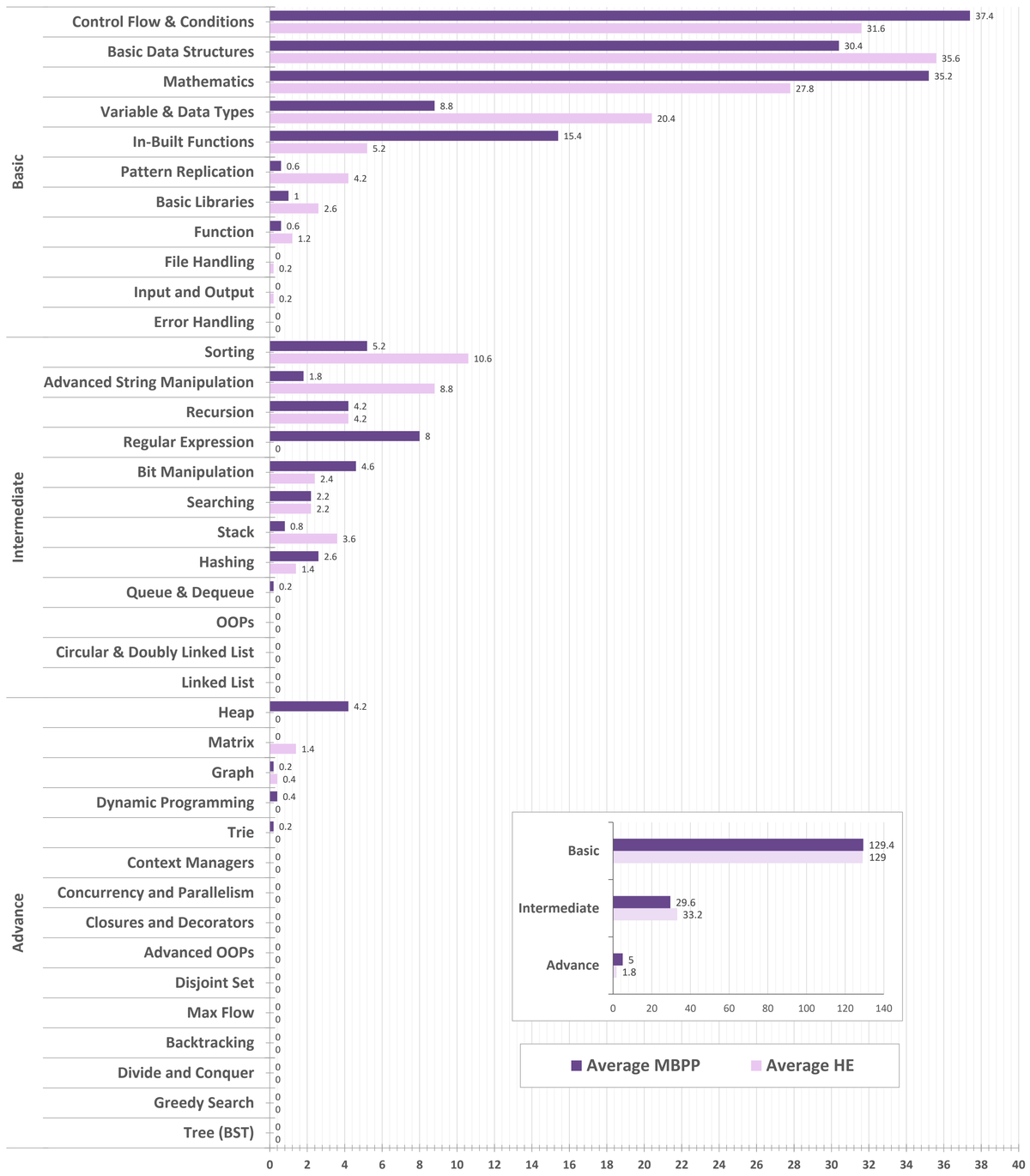

Figure 2: Average number of problems in each of the programming concepts for two benchmarks, HumanEval and MBPP. The average number of problems assigned to each programming concept was determined by averaging the concept labels provided by five independent annotators.

4. Results

Model	Size	Pass@1	Pass@10
StarCoderBase	7B	0.0029	0.0149
StarCoder2	7B	0.0024	0.0217
Code Llama	7B	0.0067	0.0472
CodeQwen1.5-Chat	7B	0.0059	0.0497
Nxcode-CQ-orpo	7B	0.0048	0.0348
Mistral-Instruct-v0.1	7B	0.014	0.0552
Code Llama Instruct	7B	0.0178	0.0744
Deepseek Coder Instruct	6.7B	0.0345	0.1091
Code Llama Python	7B	0.024	0.0979
Llama 3	8B	0.0383	0.1494
Phi-2	2.7B	0.0302	0.1187
OpenCodeInterpreter-DS	6.7B	0.0151	0.0526
Deepseek Coder	6.7B	0.0345	0.1145
Code Llama Python	13B	0.0453	0.1514
GPT-3.5	NA	0.0724	0.2384
GPT-4	NA	0.1243	0.3311

Table 2: Comparison between open and closed-source models on PythonSaga. We use the number of samples (n) as 20 all models.

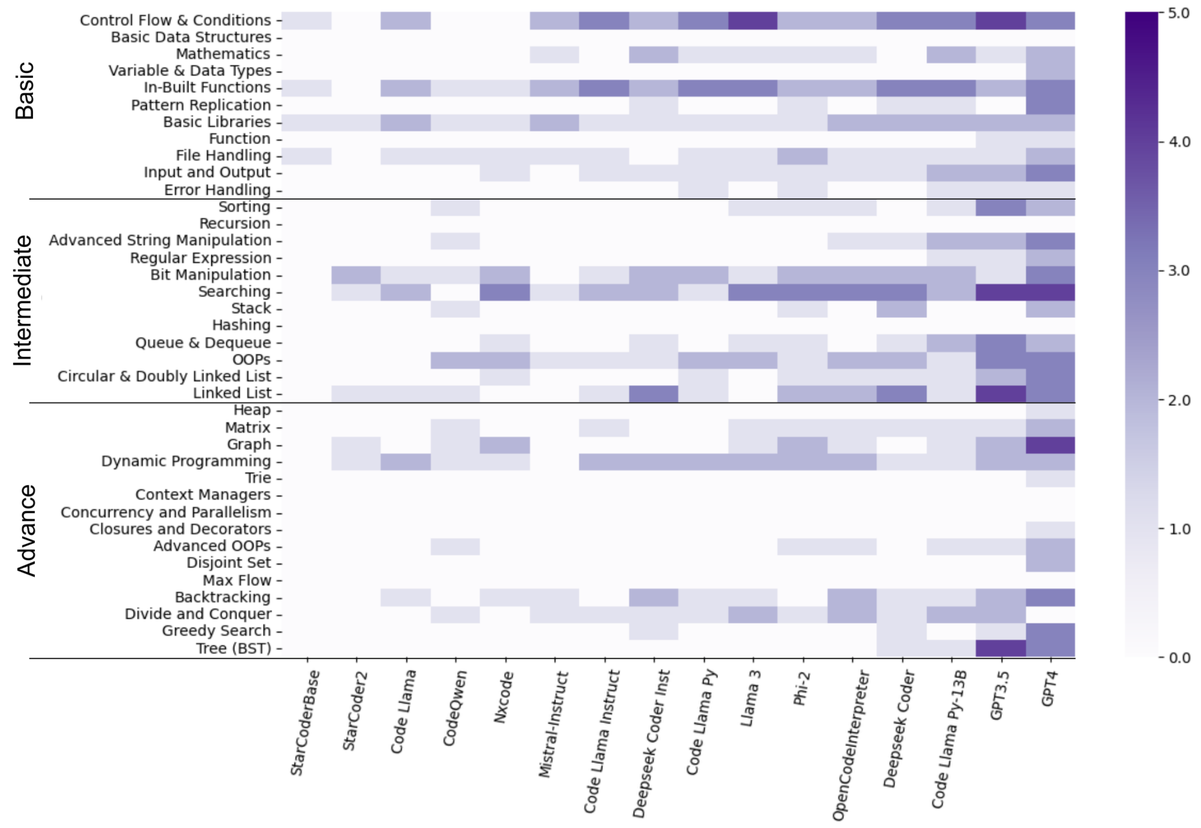

Figure 3: A heatmap showing the number of problems in PythonSaga solved by each LLM for a given programming concept. A model succeeds if at least one of the n(=20) generated samples passes all test cases.

Conclusion

A balanced evaluation framework and a new benchmark to address limited concept diversity and difficulty in existing code generation benchmarks.