Program Synthesis with Large Language Models

MohammadAmin Raeisi

Summary

This paper introduces two datasets to test Python code synthesis. The first is a new dataset called Mostly Basic Programming Problems (MBPP). It contains 974 short Python functions designed to be solved by beginner programmers, text descriptions of those programs, and test cases to check for functional correctness. The second is the MathQA-Python dataset, a Python synthesis dataset containing 23914 problems, produced by rewriting the solutions to a subset of the MathQA dataset into Python. These two datasets exercise different points in the space of synthesis tasks: MBPP contains more usage of imperative control flow, such as loops and conditionals. In contrast, MathQA-Python contains more complex natural language descriptions.

Both datasets show that a large language model performs surprisingly well at a few-shot synthesis of Python programs from a prompt. Fine-tuning further on each dataset yields a further increase in synthesis performance. It is especially notable for MBPP because the fine-tuning set is extremely small. They evaluate the model performance at scales ranging from 244M to 137B parameters, finding that performance continues to improve with increased model size. The largest models they consider can synthesize solutions to 59.6% of the problems from MBPP using few-shot learning. For most model sizes, fine-tuning increases performance by about ten percentage points. The synthesis task is indeed easier on the smaller, hand-verified MBPP dataset.

They improve the model's ability to engage in dialog about code and its performance in response to natural-language feedback from humans. They show that the model can incorporate short natural language hints to repair its outputs and clarify under-specified prompts, increasing few-shot performance from 30% without human feedback to 65% with four dialog turns, yielding a 50% error reduction.

Strengths

- It has compared the strengths and weaknesses of the existing methods by presenting two new benchmarks.
- They have provided a lot of diagrams and tables to evaluate their benchmarks.
- I don't like this article at all and I can't think of anything else positive!

Weaknesses

- This article has not proposed any new theoretical ideas and has only compared the existing methods.
- The text of the article is very long despite the fact that it does not present any specific idea.
- The text of the article is a little complicated and incomprehensible for those who are not familiar with language models.