SimSCOOD: Systematic Analysis of Out-of-Distribution Generalization in Fine-tuned Source Code Models

Hossein Hajipour\(^1\), Ning Yu\(^2\), Cristian-Alexandru Staicu\(^1\), Mario Fritz\(^1\)

\(^1\)CISPA Helmholtz Center for Information Security, \(^2\)Netflix Eyeline Studios

**Introduction**

- Generative models suffer from finite size of data.
- Programs have complex compositional nature.
  - Given a complex enough grammar we can have infinite number of potential programs.
- Despite having access to the large code datasets to pre-train these models, it remains challenging in practice to fully cover the code distribution, specifically in fine-tuning datasets.
- It is crucial to analyze the behavior of these models in different scenarios.
  - Beyond the traditional train/test splits.

**Contributions**

- We propose a systematic approach to simulate various OOD scenarios by masking out sub-regions of source code distribution along the length, syntax, and semantics dimensions.
- We find that the performance of the fine-tuned models can significantly deteriorate in various OOD scenarios despite the model encountering similar examples during the pre-training phase.
- Our systematic analysis shows that, while full fine-tuning and LoRA fine-tuning perform comparably on in-distribution code data, LoRA fine-tuning demonstrates significantly better performance on OOD data.

**Overview of SimSCOOD**

- I. Original code distribution along a dimension.
- II. OOD simulation by masking out a sub-region.
- III. Model fine-tuning.
- IV. Evaluation on OOD data.

**Experiments**

**Setup**

- **Tasks:** Text-to-Code, Code refinement
- **Models:** GraphCodeBERT [1], CodeT5 [2], CodeT5+ [3], Code Llama [4]
- **Fine-tuning:** Full fine-tuning, LoRA fine-tuning [5]

**How Do Fine-tuned Models Generalize?**

- Overall results of the model performance for different scenarios in text-to-code task.
  - The results provide the relative exact match to the 100% baseline for different OOD and few-data regime scenarios.
  - FT denotes full fine-tuning, and LoRA refers to the LoRA fine-tuning method.

**How fine-tuned models perform on the full dataset?**

- Exact match results of the fine-tuned models using the full fine-tuning dataset.

**Can fine-tuned LLMs generate unseen language elements?**

- Performance of fine-tuned models, can significantly deteriorate in OOD scenarios, even when the models have seen similar code samples during pre-training.
- While full fine-tuning and LoRA fine-tuning methods show comparable results over in-distribution data, LoRA fine-tuning significantly outperforms full fine-tuning in OOD scenarios.
- By incorporating a small amount of relevant data into the fine-tuning set, models can achieve substantial performance enhancements.

**References**