An Extensive Study on Pre-trained Models for Program Understanding and Generation

Zhengran Zeng†
Southern University of Science and Technology
China
12032889@mail.sustech.edu.cn

Hanzhuo Tan
Southern University of Science and Technology and The Hong Kong Polytechnic University
China
hanzhuo.tan@connect.polyu.hk

Haotian Zhang
Kwai Inc.
China
zhanghaotian@kuaishou.com

Jing Li
The Hong Kong Polytechnic University
China
jing1li@comp.polyu.edu.hk

Yuqun Zhang*
Southern University of Science and Technology
China
zhangyq@sustech.edu.cn

Lingming Zhang
University of Illinois Urbana-Champaign
United States
lingming@illinois.edu

ABSTRACT

Automatic program understanding and generation techniques could significantly advance the productivity of programmers and have been widely studied by academia and industry. Recently, the advent of pre-trained paradigm enlightens researchers to develop general-purpose pre-trained models which can be applied for a broad range of program understanding and generation tasks. Such pre-trained models, derived by self-supervised objectives on large unlabelled corpora, can be fine-tuned in downstream tasks (such as code search and code generation) with minimal adaptations. Although these pre-trained models claim superiority over the prior techniques, they seldom follow equivalent evaluation protocols, e.g., they are hardly evaluated on the identical benchmarks, tasks, or settings. Consequently, there is a pressing need for a comprehensive study of the pre-trained models on their effectiveness, versatility as well as the limitations to provide implications and guidance for the future development in this area. To this end, we first perform an extensive study of eight open-access pre-trained models over a large benchmark on seven representative code tasks to assess their reproducibility. We further compare the pre-trained models and domain-specific state-of-the-art techniques for validating pre-trained effectiveness. At last, we investigate the robustness of the pre-trained models by inspecting their performance variations under adversarial attacks. Through the study, we find that while we can in general replicate the original performance of the pre-trained models on their evaluated tasks and adopted benchmarks, subtle performance fluctuations can refute the findings in their original papers. Moreover, none of the existing pre-trained models can dominate over all other models. We also find that the pre-trained models can significantly outperform non-pre-trained state-of-the-art techniques in program understanding tasks. Furthermore, we perform the first study for natural language-programming language pre-trained model robustness via adversarial attacks and find that a simple random attack approach can easily fool the state-of-the-art pre-trained models and thus incur security issues. At last, we also provide multiple practical guidelines for advancing future research on pre-trained models for program understanding and generation.

CCS CONCEPTS

• **Software and its engineering** → Reusability.

KEYWORDS

Code Representation, Deep Learning, Pre-Trained Language Models, Adversarial Attack

ACM Reference Format:
Zhengran Zeng†, Hanzhuo Tan, Haotian Zhang, Jing Li, Yuqun Zhang*, and Lingming Zhang. 2022. An Extensive Study on Pre-trained Models for Program Understanding and Generation. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '22), July 18–22, 2022, Virtual, South Korea.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3533767.3534390

1 INTRODUCTION

The research tasks of program understanding and generation, e.g., code summarization, code generation, code search, and defect prediction, have been increasingly studied for decades. Moreover, the advent of deep learning and machine learning techniques has strongly advanced the progress of such research domains [9, 25, 26, 50, 84]. Recently, a group of researchers have proposed natural language (NL)-programming language (PL) pre-trained models to provide general-purpose representations of the semantic connections between natural languages and program languages to support various program understanding and generation tasks once and

† Zhengran Zeng is also affiliated with the Research Institute of Trustworthy Autonomous Systems, Shenzhen, China.
* Yuqun Zhang is the corresponding author. He is also affiliated with the Research Institute of Trustworthy Autonomous Systems, Shenzhen, China and Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, China.

ISSTA '22, July 18–22, 2022, Virtual, South Korea

ACM ISBN 978-1-4503-9379-9/22/07…$15.00
https://doi.org/10.1145/3533767.3534390

for all [3, 16, 18, 21, 27]. In contrast to conventional deep learning techniques which require task-specific feature extraction or model selection, such NL-PL pre-trained models are usually derived from self-supervised language modeling tasks on large unlabelled program language corpora with simple modification and low-cost fine-tuning. Specifically, they tend to adopt the popular transformer [69] architectures, e.g., BERT [15] and GPT [55], for pre-training to facilitate the performance on program understanding and generation tasks. For instance, CodeBERT [18] is pre-trained on millions of program functions and natural language comments by two self-supervised tasks—Masked Language Modeling (i.e., randomly masking input tokens) and Replaced Token Detection [13] (i.e., randomly replacing input tokens) to predict the masked/replaced tokens for input tokens. After pre-training, the CodeBERT model is fine-tuned on different tasks. Moreover, the recently proposed CodeX[12], a GPT-based model pre-trained on massive code samples from Github, automatically fixed 28.8% of programming problems in the HumanEval[12] dataset on the automatic programming task.

Despite the effectiveness and versatility of the NL-PL pre-trained models shown in their original papers, their performance evaluations can nevertheless be susceptible to bias. Specifically, while there are many potential tasks which the NL-PL pre-trained models can be applied for [24, 54, 65, 66, 83], they hardly are applied for fully identical tasks for comprehensive performance comparison and analytics. Moreover, even for certain identical tasks they are evaluated upon, they sometimes adopt diverse benchmarks such that their performance can hardly be precisely compared either. Furthermore, many studies have addressed that general-purpose language models for NL are essentially defective. For instance, their efficacy can be hard to reproduce [6]. In addition, they are quite sensitive to noises and attacks under a set of behavioral tests [59]. Such facts altogether can indicate a strong need of an extensive study for the existing NL-PL pre-trained models.

In this paper, to our best knowledge, we conduct the first comprehensive study on the existing NL-PL pre-trained models to enhance the understanding of their strengths and limitations. Specifically, our objectives of the study include (1) validating the performance of different NL-PL pre-trained models, (2) comparing such models with the previous domain-specific state-of-the-art (SOTA) models, and (3) investigating the robustness of pre-trained models. Note that although the existing work, e.g., CodeXGLUE [43], also presents experimental studies on NL-PL pre-trained models, they hardly evaluate all studied models simultaneously on universal benchmarks and fail to provide insightful analyses on them.

To this end, we first determine to extensively study the effectiveness of the pre-trained models with expanded tasks and datasets which have not been explored by any previous work. In particular, we adopt eight mainstream pre-trained models, i.e., CodeBERT [18], CodeGPT [43], CodeT5 [71], CodeTrans [16], ContraCode [27], CoTexT [53] GraphCodeBERT [21], and PLBART [3] as our studied subjects mainly because they are publicly available SOTA models designed for a broad spectrum of the program understanding and generation tasks. We then evaluate all the studied pre-trained models on top of benchmark `CodeXGLUE` [43] because its inclusive datasets have been widely adopted by many of the studied models and it also presents deterministic training/testing data split for strengthening the fairness of their performance comparison. Meanwhile, we evaluate the effectiveness of the studied models against the non-pre-trained SOTA techniques of individual program understanding and generation tasks on top of an extended benchmark. Next, we further inspect the robustness of the studied models via adversarial attacks with multiple existing approaches and a simple random attack approach proposed in our paper.

Our study exposes multiple important and interesting insights for designing and developing future NL-PL pre-trained models. Specifically, we find that no pre-trained models can dominate the other models on all studied downstream tasks and their validity can be compromised. More specifically, while the performance of our studied pre-trained models can be generally replicated on their originally adopted benchmarks, subtle performance fluctuations can refute the performance comparison results of the studied models in their original papers. For instance, although PLBART reports performance superiority over CodeBERT, our study indicates that it cannot outperform CodeBERT for defect detection and code search. We also find that there hardly exists a dominating pre-trained model and pre-training with multiple objectives (e.g., the encoder-decoder-based models) potentially compromises the pre-training power for individual objectives. Moreover, by comparing with the domain-specific SOTA techniques, we find that the pre-trained models could enable prominent performance on multiple studied tasks, e.g., CodeGPT improves 10.18% and 63.94% on clone detection and code search over the datasets adopted in the original papers of the corresponding SOTA techniques. At last, we propose a new adversarial attack framework for evaluating the robustness of the pre-trained models and demonstrate for the first time that despite the presented excellent performance, the studied models can be easily attacked by a simple random attack approach. Interestingly, even though certain models, e.g., GraphCodeBERT, enhances the performance over existing models, e.g., CodeBERT, by injecting auxiliary information (i.e., DFG) for modeling, it is even more vulnerable to adversarial attacks. Accordingly, our study also reveals various practical guidelines for advancing NL-PL pre-trained models in the near future.

To summarize, our paper makes the following contributions.

- **Dataset.** For the commonly studied tasks, we collect an extensive dataset including a widely-used dataset `CodeXGLUE` and the datasets from domain-specific techniques such that each task is assigned with common and sufficient datasets for evaluating multiple NL-PL pre-trained models.
- **Study.** An extensive study with general-purpose NL-PL pre-trained models on the proposed datasets which contributes on (1) validating and calibrating performance of the original papers, (2) comprehensive and standardized experimental studies compared to the original papers, (3) adopting adversarial attacks for NL-PL pre-trained models to study their robustness for the first time.
- **Implications.** This work reveals multiple implications including (1) reliable and replicable experiments are essential before releasing pre-trained models since subtle performance fluctuations across multiple runs can easily override their performance comparisons; (2) proposing dominating general-purpose models can be challenging—surprisingly, state-of-the-art encoder-decoder-based models can compromise their encoder components and