PyTy: Repairing Static Type Errors in Python

PyTy is an automated program repair approach specifically designed for Python type errors. PyTy utilizes a learning-based model trained on a dataset of Python type error fixes called PyTyDefects.

Purpose

Submission for ICSE 2024 Artifact:
- Available Badge: We provide the artifact with a permanent DOI from Zenodo and also maintain a public GitHub repository for the project.
- Reusable Badge: We describe how to reproduce the paper's results using Docker and use the tool to fix new bugs in other repositories.

Provenance

- The source code and data are publicly available on Zenodo and GitHub: DOI (TODO) and https://github.com/sola-st/PyTy.

- As a timestamp, the last GitHub commit before submitting the artifact is: (TODO).

Data

We also include the dataset we collected, named PyTyDefects. The full dataset in JSON format is available in the folder: ./src/Input. Each JSON file represents a commit containing one or more type error fixed after applying our delta debugging technique to isolate the fixes. This dataset can be reused for other studies and approaches.

Setup

- Hardware: To run the script in "FAST MODE", a normal computer suffices. For "SLOW MODE", we used (and recommend) a server with 250 GB RAM, 48 Intel Xeon CPU cores with 2.2Ghz and an NVIDIA Telta V100 GPU.
- Software:
  - Ubuntu OS
  - Docker ([installation instructions](https://docs.docker.com/engine/install/ubuntu/))

Important files

You can find the detailed instructions:
- [README] The file *README.md* contains instructions to reproduce results and to reuse the code with a different dataset
- [LICENSE] The file *LICENSE.md* contains the open source licence for the source code
- [REQUIREMENT] The file *requirements.txt* contains all the requirements to run the source code
- [PAPER] The file paper.pdf contains the paper accepted at ICSE 2024