The Evolution of Type Annotations in Python: An Empirical Study

This paper presents the first large-scale empirical study of the evolution of type annotations and type errors in Python. The study is based on an analysis of 1,414,936 type annotation changes, which we extract from 1,123,393 commits among 9,655 projects. Our results show that (i) type annotations are getting more popular, and once added, often remain unchanged in the projects for a long time, (ii) projects follow three evolution patterns for type annotation usage -- regular annotation, type sprints, and occasional uses -- and that the used pattern correlates with the number of contributors, (iii) more type annotations help find more type errors (0.704 correlation), but nevertheless, many commits (78.3%) are committed despite having such errors. Our findings show that better developer training and automated techniques for adding type annotations are needed, as most code still remains unannotated, and they call for a better integration of gradual type checking into the development process.

Source code and results

The source code and the data are public available on GitHub: https://github.com/sola-st/PythonTypeAnnotationStudy or in PythonTypeAnnotationStudy-master.zip

As timestamp, the last commit before submitting the artifacts is d0bef9c488dc599f8400949111a567fd66707948

About the project folder structure:

- The folder *./Resources/Output/* contains results for RQ1, RQ2 and RQ3.
- The folder *./Resources/Output_typeErrors* contains results for RQ2 and RQ4.
- [README, INSTALL] 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.md* contains all the requirements to run the source code
- [INSTALL] The file *INSTALL.md* installs the requirements and runs the source code

Reproduce the results

- Requirements: Python 3.5+
- Suggested: Ubuntu OS

- Run the following commands:
    - sudo apt install python3-pip
    - sudo apt install python3-virtualenv
    - virtualenv -p /usr/bin/python3 test-env
    - source test-env/bin/activate
    - pip3 install -r requirements.txt
    - python3 script_typeAnnotation_analysis.py
    - python3 ./PlotResultsAndComputeStats.py

    - Wait around three minutes and the paper figures are in:
        - Figure 2: ./Resources/Output/annotationsPerYear2.pdf
        - Figure 3: ./Resources/Output/elements_annotated.pdf
        - Figure 4: ./Resources/Output_typeErrors/per_project/facebookresearch-pytext.pdf (and deepinsight-insightface.pdf and hhatto-autopep8.pdf)
        - Figure 5: ./Resources/Output/perc_annotations_lines_per_commit.pdf
        - Figure 7: ./Resources/Output/num_changes.pdf
        - Figure 9: ./Resources/Output/TopChanged_arg.pdf (and TopChanged_ret.pdf and TopChanged_var.pdf)
        - Figure 11: ./Resources/Output_typeErrors/errors_vs_annotations.pdf

- If you want to run all the experiments from scratch (~50 hours on a 48-core server):
    - sudo apt install python3-pip
    - sudo apt install python3-virtualenv
    - virtualenv -p /usr/bin/python3 test-env
    - source test-env/bin/activate
    - pip3 install -r requirements.txt
    - python3 ./results_replicability.py --slow
    - python3 script_typeAnnotation_analysis.py
    - python3 ./PlotResultsAndComputeStats.py

A usage example with a random repository:
- In ./GitHub run 'git clone https://github.com/httpie/httpie.git'
- Remove all the files from ./Resources/log
- Run the following command:

```
sudo apt install python3-pip
sudo apt install python3-virtualenv
virtualenv -p /usr/bin/python3 test-env
source test-env/bin/activate
pip3 install -r requirements.txt
python3 ./results_replicability.py --new
```

- Wait a few minutes and you can find the results in ./Resources/log/