

PYNBLINT & Connections to my PhD work

My PhD work is focused on techniques for building cloud-based compilers & static analysis tools. We are interested in tools that help “normal” text-based programming and visual programming languages. We haven’t really been thinking about notebooks at all, so my knowledge in this area is unfortunately quite low. I’ll still try answering the questions.

What did you learn, in your individual session, about static analysis for ML and the pynblint tool?

I learned about some of the problems with notebooks. For example about how data is shared between the cells, which is the reason for the “cells executed in a non-linear order” lint warning. As somebody who has spent most of his career writing Java and TypeScript, this was new (and quite concerning!). The need for notebook-specific linters now make a lot of sense, since the potential to make mistakes is so high.

We didn’t talk about machine learning, so nothing new there.

Will pynblint be useful to you in your WASP PhD project? Why or why not?

I don’t think there will be any reason for me to use the tool. However, I might be able to apply my findings to help with future development of pynblint.

Ideas for how the tool could be improved?

The main improvement would be to add IDE integration. I don’t want to leave my development environment to run analysis. I personally almost never use any analysis tool that needs to be run in a terminal setting. There are a few exceptions, like in CI/CD/test pipelines.

Another small suggestion is to expand the “-help” command. One common problem among many (all?) program analysis tools is that they can produce warnings that are technically correct, but that the programmer doesn’t want to see. I asked about this and Luigi mentioned that the tool can be configured to change error thresholds and/or hide certain warnings entirely. However, it isn’t clear to me how I would do that from looking at “pynblint -help”. I like it when tools contain their own documentation, rather than requiring the user to go to an external website.

What do you see as the limits for static analysis tools in ML? For code, models, and for data?

Luigi mentioned wanting to integrate pynblint with PyLint. Since notebook cells are allowed to be executed out of order, there will likely be tons of false positives from PyLint. Warnings about certain variables not being defined will probably appear very often. So then you need to apply heavy filtering on the results to avoid annoying the users with false positives. But then you’ll probably filter out false negatives too.. It’s a difficult trade-off. Perhaps there should be a “strict mode” flag where the entire notebook gets treated (analysed) as a sequential program, and then you can analyse it like any other python program.

For ML/models specifically, you have to limit yourself to analysing them as a black box I think. It would be nice if a static analyser could say that certain inputs to a model are unnecessary or incorrect, but I don’t know if that is possible. However this is far outside my comfort zone so I’m really just guessing.