

Final Project Proposal

Code Prediction

For the final project I want to stick, generally, with the topic area I chose for the midterm paper. I find the algorithm classification problem interesting, so the code completion / algorithm prediction problem is really interesting too. For this final project, I want to further explore the state-of-the-art and practical implementations of code completion/algorithm prediction.

Primarily, I think it would be interesting to do a study of the practicality of the different code completion AI's on various types of coding. The first step is to figure out the coverage of the various methods across languages and what type of input is necessary for it to work. My goal would be to find 3-5 academic and productionalized models to work with that overlap on some reasonable type of coding. I will definitely use the CodeXGLUE state-of-the-art implementation and ideally another highly performant academic model. In addition to that, there are a few popular code completion tools to use in practice and I will choose of those.

The major piece of work for this project would be the creation of a "test set" of reasonable, real-world code and a system of metrics for evaluation. This is where the engineering will be important, but may be scattered due to the various implementations. However, similar to important contributions of other papers where they introduce a standard for evaluation, I aim to contribute a method and metric to help estimate the functional improvement due to these tools.

This world of AI for software engineering has an important distinction from the world of general machine learning and AI - it is focused on improving the life cycle of software engineering. It is my personal opinion that code completion / algorithm prediction is the holy grail of AI for SE, but I know the code completion suggestions are far from that. However, how far away is it really?

That is my core motivating question. I believe that distance may be estimable through performance on various tasks across the best implementations we currently have available. I envision there are a few categories of performance I will want to measure: accuracy of suggestion, contribution of logic, and multi-step prediction. The accuracy of suggestion is mostly a code completion idea, whereas the contribution of logic and multi-step prediction are vital components of algorithm prediction. I'm not sure about the current state of the practical offerings and if they have any concept of the actual algorithm being constructed, or if they are only functional for completion. I think this is something I look forward to finding out across the board.