TSE: Second Progress Report

Team Deep Understanding (Shirish Singh, Aria Jiang, Yiqiao Liu)

April 2021

1 Changes Since First Progress Report

There is no change regarding novelty, value to user community, dataset(s) and/or comparison subjects.

2 Project Description and Research Questions

The project aims to build a code understanding tool that helps find code clones across multi-file projects. The tool will be based on machine learning techniques and models. A motivating scenario for such a tool is that if a developer wants to understand a piece of code that s/he has encountered for the first time, it would be helpful to find similar and familiar code that the developer might have developed before. The tool can find behavioral similarity between the new code and code the developer is familiar with. More broadly, the applications of behavioral similarity are manifold: a) Program understanding for first-time developers, b) code search, c) code refactoring, and so on. We will attempt to answer the following research questions:

- **Does obfuscation of variable names yield an improved code2seq [1] model?**
- **How can we utilize code2seq model's output for code similarity?**

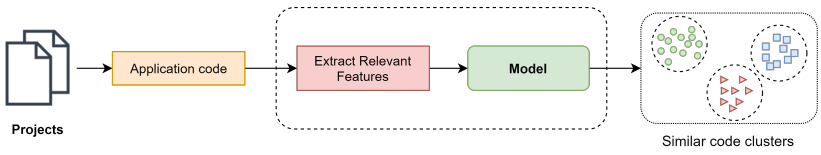

Figure 1: High level overview

3 Challenges

The obfuscation script provided by the researchers [2] failed on some Java programs. We do not consider the failure on a small number of data samples to be a significant disadvantage since this behavior was also observed in their research [2].

We faced technical challenges while pre-processing and training the code2seq model. The scripts were not able to fully pre-process all the files in the data set because of time and memory constraints. In addition, because of the large size of the model and limited GPU memory, we could only train the model on CPU. At last, we were able to successfully train the model in for 19 epochs.

Regarding performance evaluation, although all of the submission files serve similar roles, it is hard to tell which functions in these files are similar, i.e, there is no ground truth, so I need to manually select similar functions to test. At last, we are still not able to give a absolute value of accuracy, but it should be enough for comparing the relative accuracy between non-obfuscated-data-trained code2seq and obfuscated-data-trained model.

4 Threats to Validity

As mentioned in the challenges section, some files in the dataset were not obfuscated. Furthermore, some files were excluded from the training dataset because of the default time constraints of code2seq. Consequently, our obfuscated code model was trained on fewer samples than the original Java-small dataset. Our study results would likely change if we use a larger dataset or include the excluded files.

5 Demo

Our demo will take about five minutes.

First, we plan to give a one-minute introduction of what our project does, what our research questions are, why our project is interesting and how it contributes to the field of program understanding.

After the introduction, we plan to spend about two minutes explaining how we set up the correct environment, what datasets we are using, how we obfuscated the data, how we trained the models, and how we evaluated the models. If time permits, we will discuss some of the challenges we met during the process.

Lastly, we plan to spend two to three minutes explaining the results of our project. More specifically, how we compared the prediction results on the submission data set obtained from the two code2seq models and our findings. In this way, we are also answering the two research questions we put forth earlier.

6 Task Division

6.1 Task Division Summary for Phase 1

- **Aria Jiang**: Configured environment (installed necessary packages), adapted existing model (code2seq) so that it can evaluate the new dataset and outputted the prediction results.
- **Yiqiao Liu**: Apply word2vec to the output of code2seq, calculate the similarity/distance between each pair of embedding, and find the "closest" code.
- **Shirish Singh**: Environment setup, ASE Dataset extraction and cleaning.

6.2 Task Division Summary for Phase 2

- **Aria Jiang**: Re-evaluated submission data with code2seq's small model (instead of large), trained code2seq model with preprocessed java-small data for 19 epochs, trained code2seq model with obfuscated java-small data for 19 epochs with Shirish's help.
- **Yiqiao Liu**: Apply word2vec to the output of code2seq, calculate the similarity/distance between each pair of embedding, and find the "closest" code. Compare the new results with the former ones (in phase 1).
- **Shirish Singh**: Get obfuscation tool and setup environment. Pre-process java-small dataset for training. Setup training pr-processing scripts, train obfuscated code2seq and **troubleshooting environment issues**.

6.3 Task Division Plan for Demo

- **Aria Jiang**: Finish training code2seq model with obfuscated java-small data for 3000 epochs (if not feasible with given time, we will reduce the number of epochs and retrain the model for the original preprocessed java-small data with the same number of epochs). Evaluate submission data with code2seq model trained with obfuscated data. Prepare to present data processing, model training and evaluation during demo.
- **Yiqiao Liu**: Compare results between non-obfuscated results (as shown in Figure 3) and obfuscated ones after the obfuscated-data-evaluated submission data are ready. Prepare the slides for explaining the results, which includes the visualization of results.
- **Shirish Singh**: Wrap-up code2seq training, prepare materials for public release, prepare demo slides, assist in analysis, and address any technical challenges.

References

[1] Uri Alon, Shaked Brody, Omer Levy, and Eran Yahav. code2seq: Generating sequences from structured representations of code. In *International Conference on Learning Representations*, 2019.

[2] Rhys Compton, Eibe Frank, Panos Patros, and Abigail Koay. Embedding Java Classes with Code2vec: Improvements from Variable Obfuscation. In *Proceedings of the 17th International Conference on Mining Software Repositories*, MSR '20, page 243–253, New York, NY, USA, 2020. Association for Computing Machinery.