TSE: First Progress Report

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

March 2021

1 Project Description

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 [2] model?
- How can we utilize code2seq model's output for code similarity?

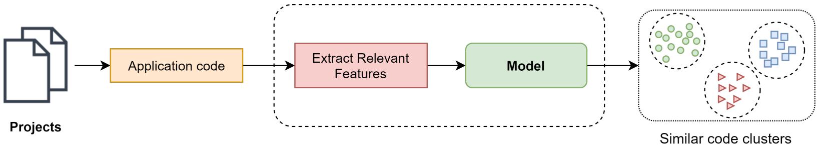


Figure 1: High level overview

2 Novelty

There is a project researching whether obfuscation of variable names may influence the performance of code2vec. That project achieved a positive answer. Therefore, in our project we aim to find out whether obfuscation can improve code2seq, a state-of-the-art model. And if so, how does it improve code2seq.

3 Value to User Community

Investigating performance on code2seq may help researchers/developers understand obfuscated programs better. In addition, program understanding can improve developer's performance by recommending method names.

4 Dataset

We plan to use two datasets for the project. The first one is the 105 student submissions [1] from the Advanced Software Engineering course that took place in Fall-2020. The second one is the java-small dataset provided by code2seq, originally used in another paper [1]. This dataset contains 11 relatively large Java projects (700K samples).

In our first phase, we adapted code2seq to evaluate the submission dataset (using large model). In our second phase, we plan to 1) obfuscate the java-small dataset used by code2seq to train the small code2seq model, 2) train a new code2seq based on the obfuscated java-small dataset using an existing tool [3], 3) re-evaluate the submission dataset with the small original code2seq model, and 4) compare results obtained from both models (original and the new one trained on obfuscated dataset)

Ground Truth:

We know that the application as a whole perform similarly, however, we do not have method level ground truth. We will find similar functions based on our technique and then manually verify the results for 105 submissions.

5 Comparison Subjects

We are comparing the similarity results we got by adapting original code2seq on submission data to the results by adapting obfuscation-trained code2seq on submission data. We found the original code2seq model given on the researchers' GitHub repository.

6 Logistics

By the end of the project, we will deliver the following items:

1. Processed training and testing dataset. If they are too big to be included on Github, we will post them on zenodo.

2. Open-source implementation of our scripts (training and testing) will be posted on Github.

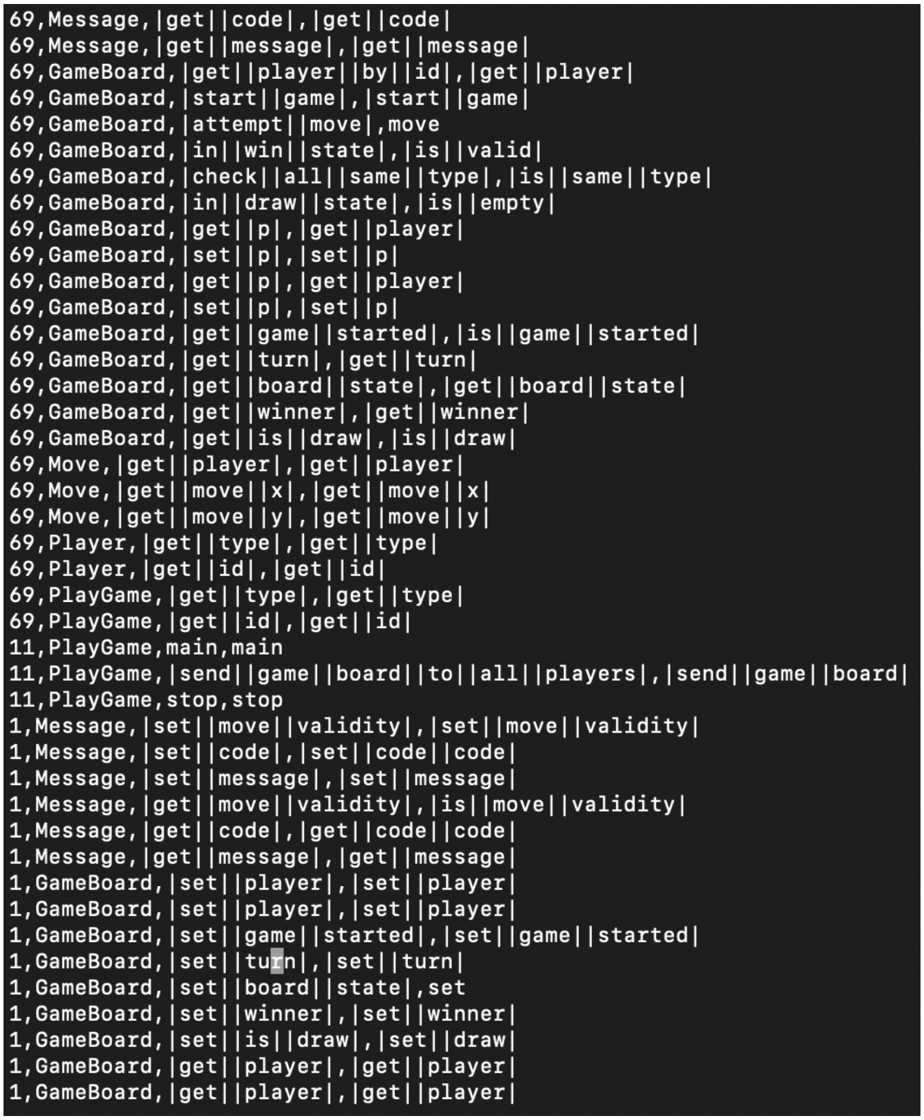
3. Final Report (delivered on Courseworks). We will use an obfuscation tool [3] [2] and the download link is included in the footnote.

[1] https://github.com/jxm033f/4156-PublicAssignment/network/members

[2] https://github.com/basedrhys/obfuscated-code2vec

7 Results

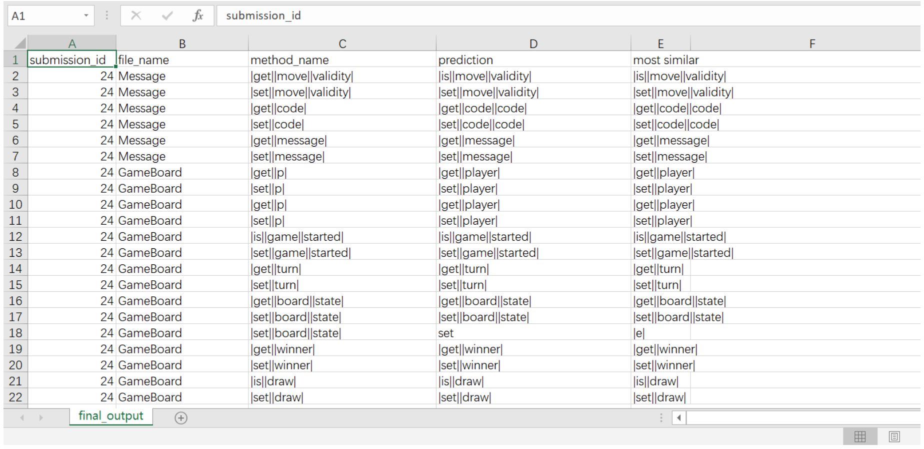
We outputted a csv file that outlines the prediction results from code2seq's large model on submission data. There are four columns: id of submission, name of the java file evaluated, original method name and predicted method name.



```
69,Message,|get||code|,|get||code|
69,Message,|get||message|,|get||message|
69,GameBoard,|get||player||by||id|,|get||player|
69,GameBoard,|start||game|,|start||game|
69,GameBoard,|attempt||move|,move
69,GameBoard,|in||win||state|,|is||valid|
69,GameBoard,|check||all||same||type|,|is||same||type|
69,GameBoard,|in||draw||state|,|is||empty|
69,GameBoard,|get||p|,|get||player|
69,GameBoard,|set||p|,|set||p|
69,GameBoard,|get||p|,|get||player|
69,GameBoard,|set||p|,|set||p|
69,GameBoard,|get||game||started|,|is||game||started|
69,GameBoard,|get||turn|,|get||turn|
69,GameBoard,|get||board||state|,|get||board||state|
69,GameBoard,|get||winner|,|get||winner|
69,GameBoard,|get||is||draw|,|is||draw|
69,Move,|get||player|,|get||player|
69,Move,|get||move||x|,|get||move||x|
69,Move,|get||move||y|,|get||move||y|
69,Player,|get||type|,|get||type|
69,Player,|get||id|,|get||id|
69,PlayGame,|get||type|,|get||type|
69,PlayGame,|get||id|,|get||id|
11,PlayGame,main,main
11,PlayGame,|send||game||board||to||all||players|,|send||game||board|
11,PlayGame,stop,stop
1,Message,|set||move||validity|,|set||move||validity|
1,Message,|set||code|,|set||code||code|
1,Message,|set||message|,|set||message|
1,Message,|get||move||validity|,|is||move||validity|
1,Message,|get||code|,|get||code||code|
1,Message,|get||message|,|get||message|
1,GameBoard,|set||player|,|set||player|
1,GameBoard,|set||player|,|set||player|
1,GameBoard,|set||game||started|,|set||game||started|
1,GameBoard,|set||turn|,|set||turn|
1,GameBoard,|set||board||state|,set
1,GameBoard,|set||winner|,|set||winner|
1,GameBoard,|set||is||draw|,|set||draw|
1,GameBoard,|get||player|,|get||player|
1,GameBoard,|get||player|,|get||player|
```

Figure 2: Code2Seq Prediction Output on Submission Data

Below is a screenshot of the final CSV file, adding a column showing the most similar results to the given predictions. KDTree is used to measure similarity.



<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>1</td><td>submission_id</td><td>file_name</td><td>method_name</td><td>prediction</td><td>most similar</td><td></td></tr>
<tr><td>2</td><td>24</td><td>Message</td><td>|get||move||validity|</td><td>|is||move||validity|</td><td>|is||move||validity|</td><td></td></tr>
<tr><td>3</td><td>24</td><td>Message</td><td>|set||move||validity|</td><td>|set||move||validity|</td><td>|set||move||validity|</td><td></td></tr>
<tr><td>4</td><td>24</td><td>Message</td><td>|get||code|</td><td>|get||code||code|</td><td>|get||code||code|</td><td></td></tr>
<tr><td>5</td><td>24</td><td>Message</td><td>|set||code|</td><td>|set||code||code|</td><td>|set||code||code|</td><td></td></tr>
<tr><td>6</td><td>24</td><td>Message</td><td>|get||message|</td><td>|get||message|</td><td>|get||message|</td><td></td></tr>
<tr><td>7</td><td>24</td><td>Message</td><td>|set||message|</td><td>|set||message|</td><td>|set||message|</td><td></td></tr>
<tr><td>8</td><td>24</td><td>GameBoard</td><td>|get||p|</td><td>|get||player|</td><td>|get||player|</td><td></td></tr>
<tr><td>9</td><td>24</td><td>GameBoard</td><td>|set||p|</td><td>|set||player|</td><td>|set||player|</td><td></td></tr>
<tr><td>10</td><td>24</td><td>GameBoard</td><td>|get||p|</td><td>|get||player|</td><td>|get||player|</td><td></td></tr>
<tr><td>11</td><td>24</td><td>GameBoard</td><td>|set||p|</td><td>|set||player|</td><td>|set||player|</td><td></td></tr>
<tr><td>12</td><td>24</td><td>GameBoard</td><td>|is||game||started|</td><td>|is||game||started|</td><td>|is||game||started|</td><td></td></tr>
<tr><td>13</td><td>24</td><td>GameBoard</td><td>|set||game||started|</td><td>|set||game||started|</td><td>|set||game||started|</td><td></td></tr>
<tr><td>14</td><td>24</td><td>GameBoard</td><td>|get||turn|</td><td>|get||turn|</td><td>|get||turn|</td><td></td></tr>
<tr><td>15</td><td>24</td><td>GameBoard</td><td>|set||turn|</td><td>|set||turn|</td><td>|set||turn|</td><td></td></tr>
<tr><td>16</td><td>24</td><td>GameBoard</td><td>|get||board||state|</td><td>|get||board||state|</td><td>|get||board||state|</td><td></td></tr>
<tr><td>17</td><td>24</td><td>GameBoard</td><td>|set||board||state|</td><td>|set||board||state|</td><td>|set||board||state|</td><td></td></tr>
<tr><td>18</td><td>24</td><td>GameBoard</td><td>|set||board||state|</td><td>set</td><td>|e|</td><td></td></tr>
<tr><td>19</td><td>24</td><td>GameBoard</td><td>|get||winner|</td><td>|get||winner|</td><td>|get||winner|</td><td></td></tr>
<tr><td>20</td><td>24</td><td>GameBoard</td><td>|set||winner|</td><td>|set||winner|</td><td>|set||winner|</td><td></td></tr>
<tr><td>21</td><td>24</td><td>GameBoard</td><td>|is||draw|</td><td>|is||draw|</td><td>|is||draw|</td><td></td></tr>
<tr><td>22</td><td>24</td><td>GameBoard</td><td>|set||draw|</td><td>|set||draw|</td><td>|set||draw|</td><td></td></tr>
</table>

Figure 3: Results from similarity experiments with KDTree

8 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.

9 Task Division Plan for Phase 2

- **Aria Jiang**: Re-evaluate submission data with code2seq's small model (instead of large) and train code2seq with obfuscated java-small data.
- **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.

References

[1] Miltiadis Allamanis, Hao Peng, and Charles Sutton. A Convolutional Attention Network for Extreme Summarization of Source Code. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2091–2100. JMLR.org, 2016.

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

[3] 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.