EECS4080 Project Report:

1. **Introduction:**

   The project Professor Song asked me to do this project to find those Research related to code summarization or code generation and use the pre-trained model they gave to run and test to see if the results were consistent with what they gave in the Research.

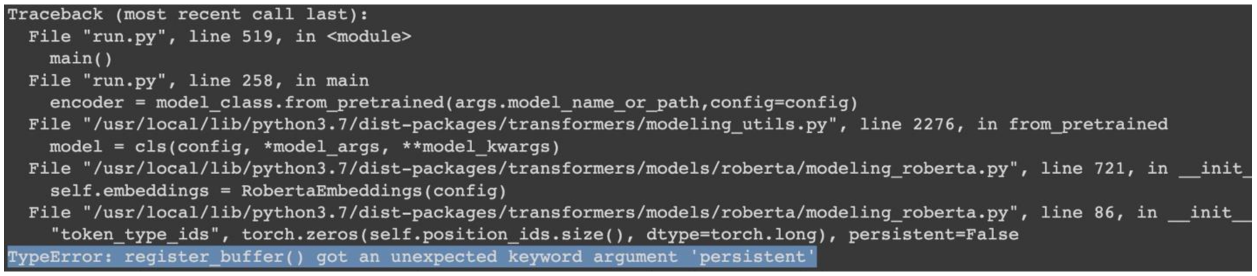
2. **Background:**

   My choice is CodeXGLUE, which Microsoft Research Asia launched, Development and Bing. It is a machine-learning dataset for training and evaluating natural language processing models. It consists of a series of language tasks, including text classification, semantic similarity calculation and text generation. CodeXGLUE has the advantage of providing a large amount of high-quality data, as well as clear and unambiguous evaluation criteria. This allows researchers to compare the performance of different models more objectively. In addition, CodeXGLUE provides a complete platform for researchers to submit and evaluate their models. In CodeXGLUE, I chose Code-Text, which is about Code summarization. Code-Text is a new machine-learning task that aims to let models learn how to generate annotations of program code. The significance of this task is that annotating program code can help developers better understand the logic of the code and can enhance the readability of the code. The Code-Text section of CodeXGLUE contains many pairs of program code, annotations, and short versions, and this data can be used to train the model. Code-Text also provides evaluation criteria for researchers to assess the model's performance. It generates natural language comments for a code and is evaluated by smoothed bleu-4 score. They use the dataset from CodeSearchNet, and the pre-trained model is CodeBERT.

3. **Problem:**

The first problem I encountered was that the run command they provide is segmented. This may cause the files generated after the first code run to be unusable in subsequent command runs on colab.

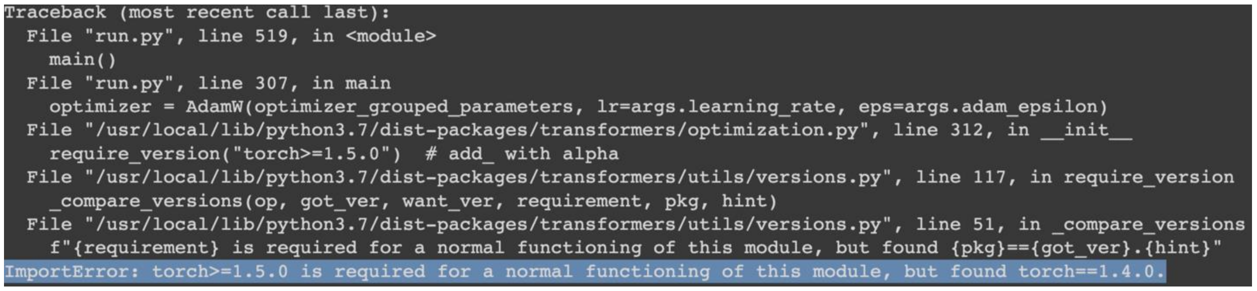
The second problem is that they provide a pipeline on the page that fine-tunes CodeBERT on this task. In the dependency, they mention that the torch version is 1.4.0. however, when I tested with the 1.4.0 torch, I got a TypeError showing an error argument called ' persistent' (Figure a).



```
Traceback (most recent call last):
  File "run.py", line 519, in <module>
    main()
  File "run.py", line 258, in main
    encoder = model_class.from_pretrained(args.model_name_or_path,config=config)
  File "/usr/local/lib/python3.7/dist-packages/transformers/modeling_utils.py", line 2276, in from_pretrained
    model = cls(config, *model_args, **model_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/transformers/models/roberta/modeling_roberta.py", line 721, in __init_
    self.embeddings = RobertaEmbeddings(config)
  File "/usr/local/lib/python3.7/dist-packages/transformers/models/roberta/modeling_roberta.py", line 86, in __init__
    "token_type_ids", torch.zeros(self.position_ids.size(), dtype=torch.long), persistent=False
TypeError: register_buffer() got an unexpected keyword argument 'persistent'
```

Figure (a) TypeError: wrong argument 'persistent'

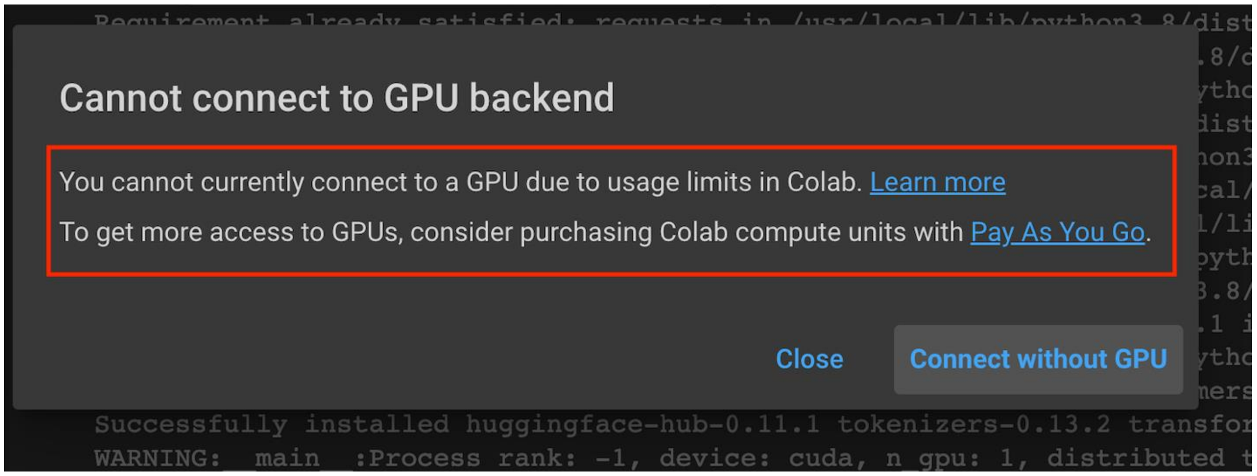
The third problem after I solved the second problem and tested further down, there was an error showing that the normal functioning of this module requires a torch version greater than or equal to 1.5.0 (Figure b).



```
Traceback (most recent call last):
  File "run.py", line 519, in <module>
    main()
  File "run.py", line 307, in main
    optimizer = AdamW(optimizer_grouped_parameters, lr=args.learning_rate, eps=args.adam_epsilon)
  File "/usr/local/lib/python3.7/dist-packages/transformers/optimization.py", line 312, in __init__
    require_version("torch>=1.5.0")  # add_ with alpha
  File "/usr/local/lib/python3.7/dist-packages/transformers/utils/versions.py", line 117, in require_version
    _compare_versions(op, got_ver, want_ver, requirement, pkg, hint)
  File "/usr/local/lib/python3.7/dist-packages/transformers/utils/versions.py", line 51, in _compare_versions
    f"{requirement} is required for a normal functioning of this module, but found {pkg}=={got_ver}.{hint}"
ImportError: torch>=1.5.0 is required for a normal functioning of this module, but found torch==1.4.0.
```

Figure(b) ImportError: torch version too old

The fourth problem after I solved the first three problems, I started to run the Ruby language tests in the mod. When I got to the fifth epoch of testing, I ran out of colab's GPU usage (Figure c). This means that I was not able to test all ten epochs.



```
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist
Successfully installed huggingface-hub-0.11.1 tokenizers-0.13.2 transfor
WARNING:__main__:Process rank: -1, device: cuda, n_gpu: 1, distributed t
```

Cannot connect to GPU backend

You cannot currently connect to a GPU due to usage limits in Colab. Learn more

To get more access to GPUs, consider purchasing Colab compute units with Pay As You Go.

Figure(c) Colab's GPU usage limit

4. **Approach:**

For the first problem, I wrote all the commands into a script file and then bash, according to a Stack Overflow post.

For the second problem, I saw a similar problem in the issue part of GitHub. I then found the Python 3.7 file and deleted the line of code about persistent.

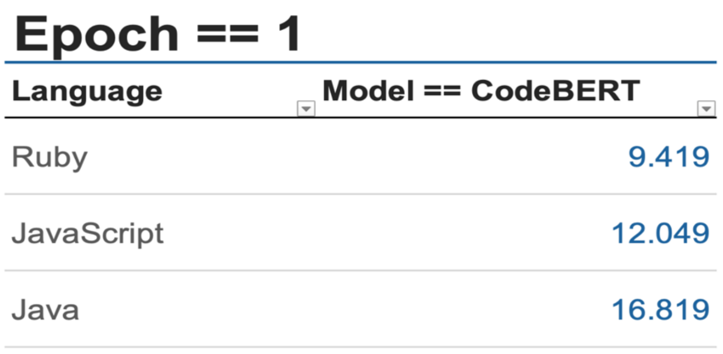
For the third problem, I found the wrong version of torch. Then I restored the code I deleted to solve the second problem and changed the torch version to the latest one. This solved both the second and third problems.

For the fourth problem, I adjusted the number of epochs in the fine-tune pipeline from 10 to 1. However, my results were somewhat different from the data in Research.

5. Result:

(1) Epochs ==1:

In the pipeline, I adjusted the number of epochs to 1 and tested the three languages Ruby, JavaScript, and Java, and obtained the results in Figure d. Figure e, the data they gave in the research, was obtained after they ran ten epochs.



Epoch == 1

Language	Model == CodeBERT
Ruby	9.419
JavaScript	12.049
Java	16.819

Figure(d) My result for Three language

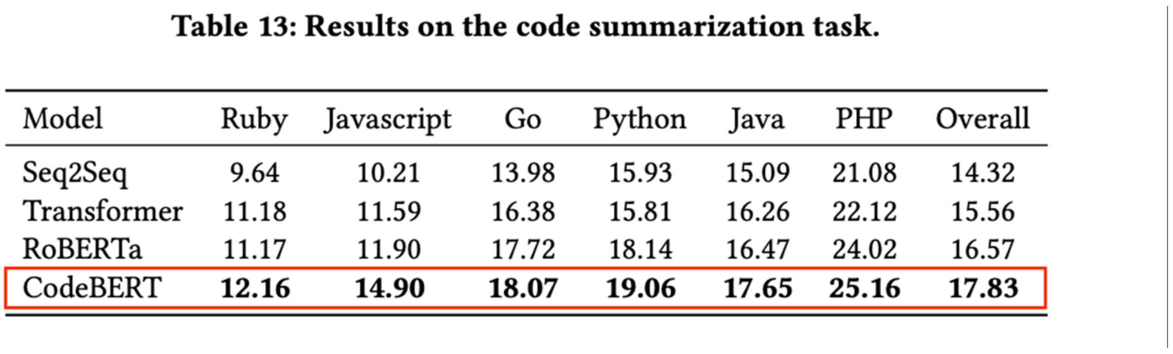


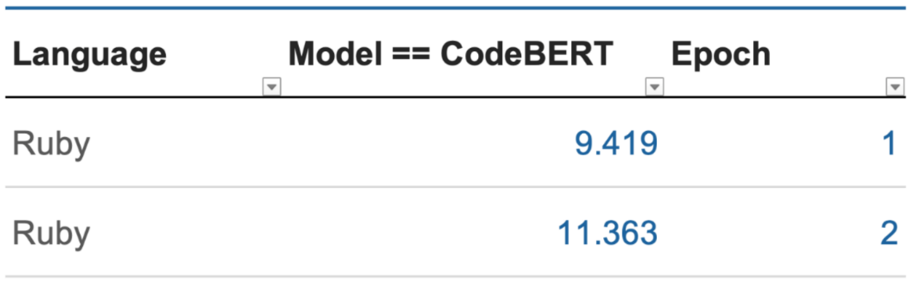
Table 13: Results on the code summarization task.

Model	Ruby	Javascript	Go	Python	Java	PHP	Overall
Seq2Seq	9.64	10.21	13.98	15.93	15.09	21.08	14.32
Transformer	11.18	11.59	16.38	15.81	16.26	22.12	15.56
RoBERTa	11.17	11.90	17.72	18.14	16.47	24.02	16.57
CodeBERT	**12.16**	**14.90**	**18.07**	**19.06**	**17.65**	**25.16**	**17.83**

Figure(e) Research's result for epochs==10

（2） Epochs == 1，2

What Figure f shows is that to test the effect of epoch changes on the test results for the same language, I ran two tests on the Ruby language, which are epochs == 1 and epochs == 2.



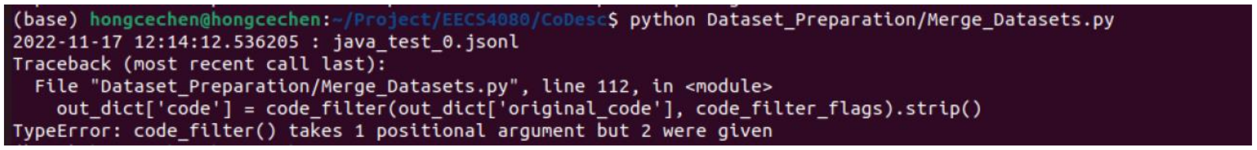
Language	Model == CodeBERT	Epoch
Ruby	9.419	1
Ruby	11.363	2

Figure(f) Result for Ruby epochs == 1 and 2

6. Discussion

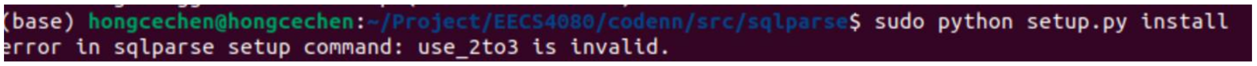
The Research results were more significant than the results I got. However, in the second test, I tested Ruby with epochs equal to 1 and 2. As the value of epochs increases, the results obtained from the test also increase accordingly. Therefore, it can be guessed that the data in this Research is somewhat reliable.

Before I found CodeXGLUE, I ran into a lot of research code that basically failed to test. For most Research, the code sent out by the researcher rarely runs straight away. Some researchers need to describe the type and version of the library they are using. This makes me spend a long time researching the version used, and it often fails because the code is so complex. Some codes have a lot of TypeErrors (even case errors in file names). As in figure g, the number of positional arguments in code_filter() is written wrong. For this kind of error, there is little I can do to solve it because the code involved is too complicated. In figure h, after checking StackOverflow, I found that only setuptools with version less than 58 can support use_2to3. Finding a code that can run properly in this Research is still challenging. Finding a code that runs appropriately in this Research is difficult.



```
(base) hongcechen@hongcechen:~/Project/EECS4080/CoDesc$ python Dataset_Preparation/Merge_Datasets.py
2022-11-17 12:14:12.536205 : java_test_0.jsonl
Traceback (most recent call last):
  File "Dataset_Preparation/Merge_Datasets.py", line 112, in <module>
    out_dict['code'] = code_filter(out_dict['original_code'], code_filter_flags).strip()
TypeError: code_filter() takes 1 positional argument but 2 were given
```

Figure(g) TypeError: wrong number of argument



```
(base) hongcechen@hongcechen:~/Project/EECS4080/codenn/src/sqlparse$ sudo python setup.py install
error in sqlparse setup command: use_2to3 is invalid.
```

Figure(h) Setuptools version problem

7. ***Summary:***

   In this course, I learned a lot of exciting things from Professor Song. This was the first I had been exposed to this aspect of machine learning, but I was very interested in it. The knowledge of machine learning and deep learning is a tough challenge for me now, and Professor Song first let me test these research codes so that I could understand this knowledge relatively simply. This project stimulates my passion and interest in machine learning and deep learning. I encountered a lot of problems and challenges while working on the project. I learned how to use tools like git and set up a virtual machine for Linux and a cloud server such as colab. For this project, I also learned about torch libraries and transformers that will be used in deep learning.