AMYA CODERs

704202002

Code To Sequence

Instruction Manual

Author:
Atul Sahay

April 21, 2020

Contents

1 Project's Directory Structure

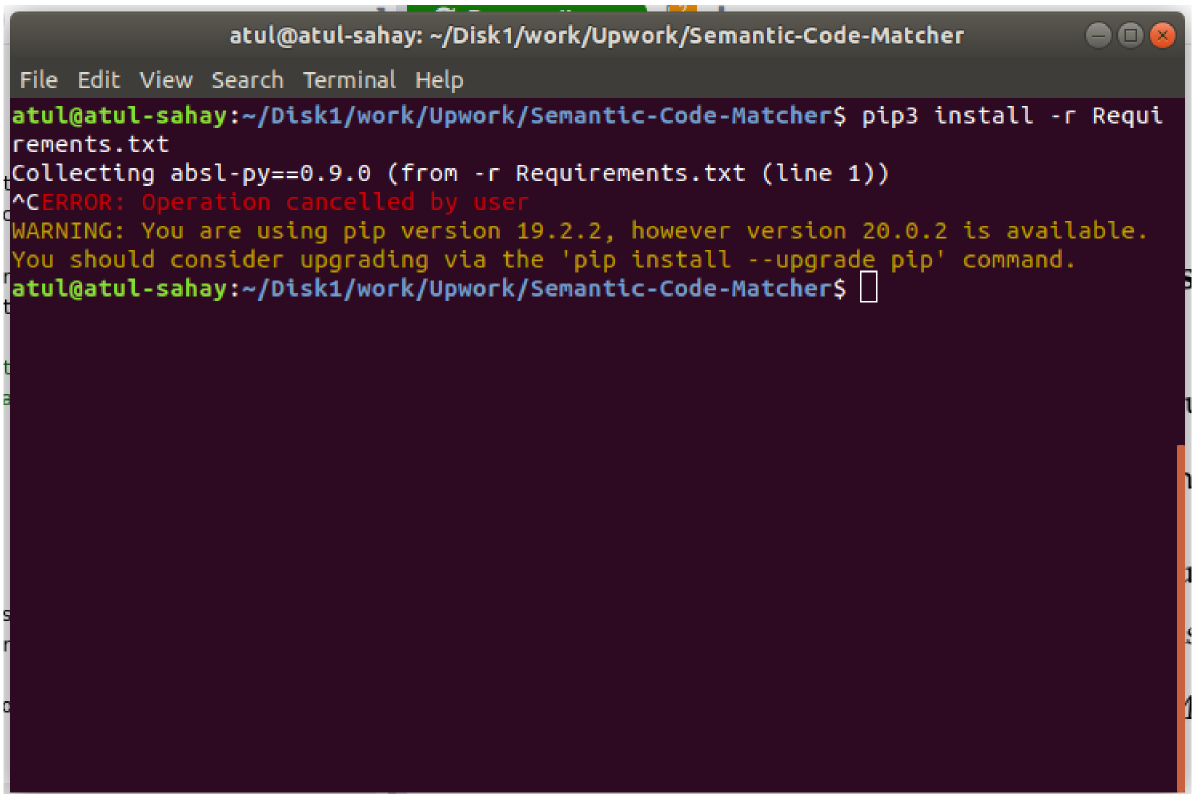
```
code2seq
├── Images
│   └── network.png
├── Python150kExtractor
│   ├── data
│   │   ├── dir
│   │   └── default
│   ├── extract.py
│   ├── data_download.sh
│   ├── extractSamples.sh
│   ├── preprocess.sh
│   └── README.md
├── models
│   └── python150k-default
├── code2seq.py
├── common.py
├── config.py
├── interactive_predict.py
├── model.py
├── Requirements.txt
├── train_python150k.sh
└── parser_python.py
```

2 Requirements

2.1 How to install required modules for the project?

- Search for the **Requirements.txt** file in the project repository.
- Open the terminal and write (be sure you are in the project directory): See Fig: 2.1

user: code2eq$: **pip3 install -r Requirements.txt**


atul@atul-sahay: ~/Disk1/work/Upwork/Semantic-Code-Matcher
File Edit View Search Terminal Help

```
atul@atul-sahay:~/Disk1/work/Upwork/Semantic-Code-Matcher$ pip3 install -r Requi
rements.txt
Collecting absl-py==0.9.0 (from -r Requirements.txt (line 1))
^CERROR: Operation cancelled by user
WARNING: You are using pip version 19.2.2, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
atul@atul-sahay:~/Disk1/work/Upwork/Semantic-Code-Matcher$ 
```

Figure 1: Installation of the required modules

3 Data Load and Preprocessing

All these operations will be done in **Python150kExtractor** directory. change your directory as given in the Section 1

3.1 Data Download

For the data loading and data preprocessing, you need to follow the given mentioned steps:

1. First you need to create a directory structure: as provided in section 1 for "data". Here the data will be stored.

2. Now just run the file name "data_download.sh"

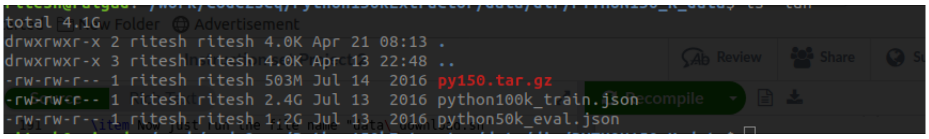
   $--\ >$ **$ chmod + data_download.sh**

   $--\ >$ **$ ./data_download.sh**

After these steps pleas run these command and check whether you are getting the similar directory structure or not.

$--\ >$ **$ ls -lah ./data/dir/**

Following two files should be present



```
total 4.1G
drwxrwxr-x 2 ritesh ritesh 4.0K Apr 21 08:13 .
drwxrwxr-x 3 ritesh ritesh 4.0K Apr 13 22:48 ..
-rw-rw-r-- 1 ritesh ritesh 503M Jul 14  2016 py150.tar.gz
-rw-rw-r-- 1 ritesh ritesh 2.4G Jul 13  2016 python100k_train.json
-rw-rw-r-- 1 ritesh ritesh 1.2G Jul 13  2016 python50k_eval.json
```

Figure 2: Directory structure of data/dir folder

3.2 Data Extraction

For the data extraction, you need to follow the given mentioned steps:

1. Just run the file name "extractSamples.sh"

   $--\ >$ **$ chmod + extractSamples.sh**

   $--\ >$ **$ ./extractSamples.sh**

2. Directory structure will resulted: as provided in section 1 for "data/default". Here the data will be extracted which will be preprocessed and used for training.

3.3 Data Peprocessing

For data preprocessing, you need to follow the given mentioned steps:

1. Make sure you have run the previous commands
2. Now just run the file name "data_download.sh"

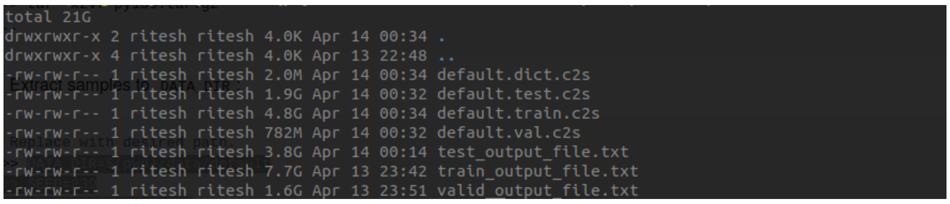
$--\ >$ **$ chmod + preprocess.sh**

$--\ >$ **$ ./preprocess.sh ./data/default**

After these steps pleas run these command and check whether you are getting the similar directory structure or not.

$--\ >$ **$ ls -lah ./data/default/**

Following files should be present:



```
total 21G
drwxrwxr-x 2 ritesh ritesh 4.0K Apr 14 00:34 .
drwxrwxr-x 4 ritesh ritesh 4.0K Apr 13 22:48 ..
-rw-rw-r-- 1 ritesh ritesh 2.0M Apr 14 00:34 default.dict.c2s
-rw-rw-r-- 1 ritesh ritesh 1.9G Apr 14 00:32 default.test.c2s
-rw-rw-r-- 1 ritesh ritesh 4.8G Apr 14 00:34 default.train.c2s
-rw-rw-r-- 1 ritesh ritesh 782M Apr 14 00:32 default.val.c2s
-rw-rw-r-- 1 ritesh ritesh 3.8G Apr 14 00:14 test_output_file.txt
-rw-rw-r-- 1 ritesh ritesh 7.7G Apr 13 23:42 train_output_file.txt
-rw-rw-r-- 1 ritesh ritesh 1.6G Apr 13 23:51 valid_output_file.txt
```

Figure 3: Directory structure of data/default folder

4 Training

All these operations will be done in the **code2seq** directory see the section 1

4.1 Pre-requisite

Make Sure you have the right files prepared from Step dataPreprocess

1. You should have these files in the root of the .Python150kExtractor/data/default:
   (a) {train,test,valid(_output_file.txt)} - these are python function ASTs tokenized (by space), 1 line per function.
   (b) {default.(train,test,valid).c2s} - these are binary files of the above mentioned file, for faster loading and processing.

4.2 Model Characteristics

- The whole model is trained on the GPU specification:
  - Name: Geforce RTX 2080 ti
  - Memory: 11 GB
  - Cuda version 10.1
- The whole process takes around 10-11 days for completion

4.3 To Train the code2seq Model

Run the command specified below

```
$ DATA_DIR=$./Python150kExtractor/data/default
$SEED=239
$DESC=default
$CUDA=0
$chmod +x train_python150k.sh
$./train_python150k.sh $DATA_DIR $DESC $CUDA $SEED
```

After the completion of training you will find a "model.final" in models/python150k-default directory. see Section 1 for the directory structure.

Note: You want to use a specific iteration file use **model_iterXX** file where specific iteration no.

5 Evaluation

ROUGE Metrix is used: for more info see this link: ”https://en.wikipedia.org/wiki/ROUGE_(metric)” subsectionPre-requisite **Make Sure you have the right files prepared from Step Training and should have the following files**

1. .models/python150k-default/model(_iterXX or .final) - You can load any file, either the final saved model or the specific iteration file
2. ./Python150kExtractor/data/default/default.test.c2s

5.1 Evaluation on Test set

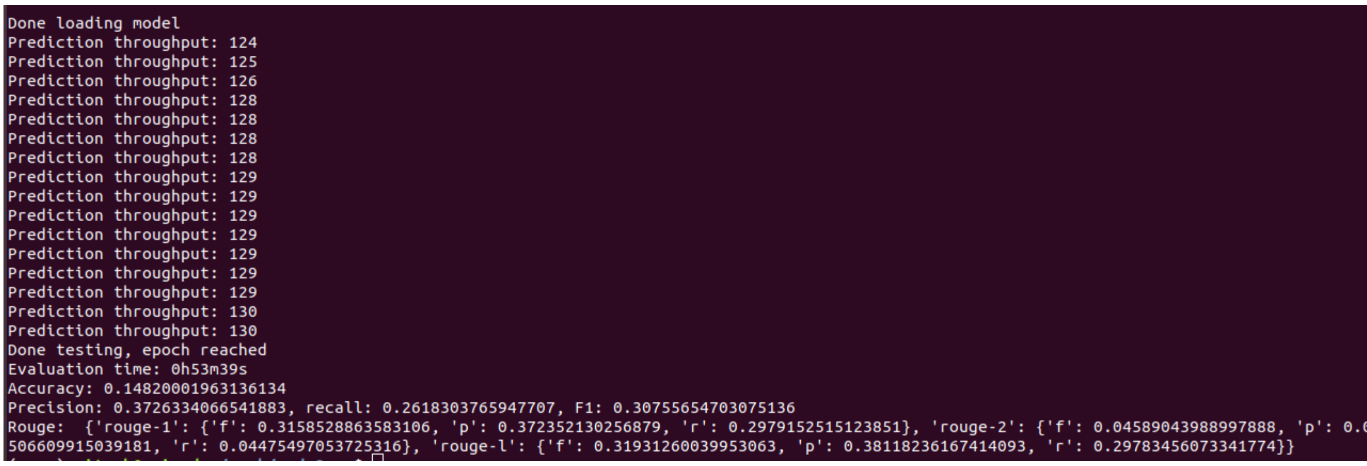
Run this command

$python3 code2seq.py

–load models/python150k-default/model.final

–test ./Python150kExtractor/data/default/default.test.c2s

You will get the following results



```
Done loading model
Prediction throughput: 124
Prediction throughput: 125
Prediction throughput: 126
Prediction throughput: 128
Prediction throughput: 128
Prediction throughput: 128
Prediction throughput: 128
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 129
Prediction throughput: 130
Prediction throughput: 130
Done testing, epoch reached
Evaluation time: 0h53m39s
Accuracy: 0.14820001963136134
Precision: 0.3726334066541883, recall: 0.2618303765947707, F1: 0.30755654703075136
Rouge:  {'rouge-1': {'f': 0.3158528863583106, 'p': 0.372352130256879, 'r': 0.2979152515123851}, 'rouge-2': {'f': 0.04589043988997888, 'p': 0.0
506609915039181, 'r': 0.04475497053725316}, 'rouge-l': {'f': 0.31931260039953063, 'p': 0.38118236167414093, 'r': 0.29783456073341774}}
```

Figure 4: Test Set Result

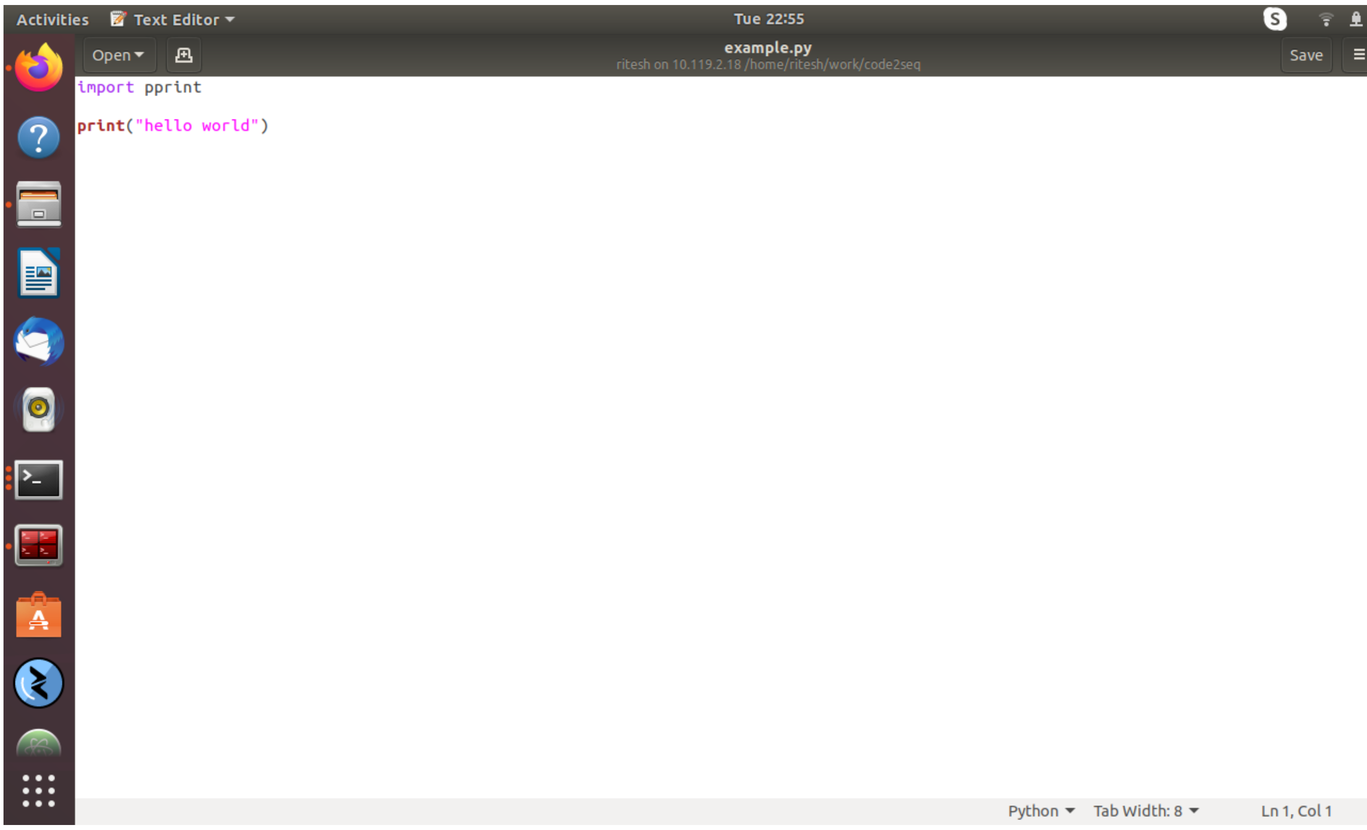
6 Helping Files

You can use the parser_python.py for getting the ASTs(for the input to the train model) for the RAW python code.

Run code

python3 parser_python.py example.py

Where example.py contains the raw python code See the figure.

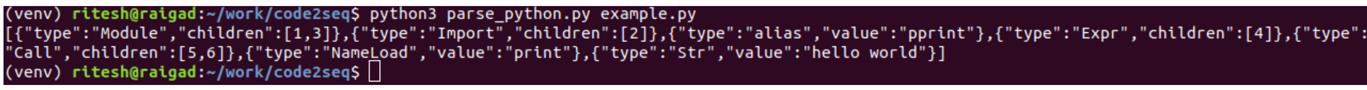


```python
import pprint

print("hello world")
```

Figure 5: Raw python code

The result is shown in the below figure



```
(venv) ritesh@raigad:~/work/code2seq$ python3 parse_python.py example.py
[{"type":"Module","children":[1,3]},{"type":"Import","children":[2]},{"type":"alias","value":"pprint"},{"type":"Expr","children":[4]},{"type":
"Call","children":[5,6]},{"type":"NameLoad","value":"print"},{"type":"Str","value":"hello world"}]
(venv) ritesh@raigad:~/work/code2seq$
```

Figure 6: Parse results