The artifact provides a Docker image to reproduce the experimental results in the paper "FIRA: Fine-Grained Graph-Based Code Change Representationfor Automated Commit Message Generation". The Docker image is in https://hub.docker.com/layers/djhao/icse_docker/newest/images/sha256-374381239d18ec1db1a5697f72864e2f8ddc9e28a7bc92df6439ed235da75caa and we also provide a zenodo link https://zenodo.org/record/5915220.

1 Requirements

- Docker
- NVIDIA GPU (option)
- NVIDIA Linux driver (option)
- NVIDIA Docker (option): Because we use GPUs to train and test the model, we use NVIDIA Docker to build and run the container. The installation guide of NVIDIA Docker is on https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html and the steps are as follows.

```
$ distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
   && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -
\
   && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-
docker.list | sudo tee
   /etc/apt/sources.list.d/nvidia-docker.list
$ sudo apt-get update
$ sudo apt-get install -y nvidia-docker2
$ sudo systemctl restart docker
```

- It is worth noting that, our artifact can also reproduce the results in the environment without **NVIDIA GPU**, **NVIDIA driver** and **NVIDIA Docker**, but it will cost more time.

2 Preparation

- Pull the image

```
$ docker pull djhao/icse_docker:newest
```

Because of the dataset and conda environment, the size of the image is around 9 GB.

- Run the image as a container using all GPUs

```
$ docker run -it --gpus all djhao/icse_docker:newest
```

or using the specified GPUs, such as 0, 1 in the following command

```
$ docker run -it --gpus '"device=0,1"' djhao/icse_docker:newest
```

- If you don't have GPUs, our artifact also support to be executed on only CPU, you can run

```
$ docker run -it djhao/icse_docker:newest
```

to enter the CPU environment.

- After you finish the reproduction, you can type the `exit` command to exit the container. Then you can execute the following command to remove the image.

```
$ docker image rm djhao/icse_docker:newest
```

3 Reproduction

After running the container, you will enter the folder **/icse_final**, and you can run the following commands in the container to reproduce the results.

3.1 Training of the model

A neural network model includes two process, training and inference. To train our model, you can run

```
$ python run_model.py train
```

The training process needs at most 30GB memory. When using GPUs, each epoch costs ~7minitues and the whole training process includes 50 epochs which needs ~6 hours. When using CPU, each epoch costs ~102 minutes and the whole training process needs ~3.5 days.

3.2 Inference of the model

Because the training time is long, we provide the model which is trained already in **best_model.pt**. With **best_model.pt**, you can directly evaluate the model and get the final output by executing

```
$ python run_model.py test
```

and the output will be saved as OUTPUT/output_fira. The inference process needs at most 8GB memory. When using GPU, the inference process costs ~6 minutes on one NVIDIA GeForce RTX 3090. When using CPU, the inference process costs ~1 hour.

3.3 Getting results of the paper

We use **table1.py**, **table2.py**, **table3.py**, **table4.py**, **table6.py** to get the results in **Table1-4** and **Table6**. The scripts compute various metrics of the commit messages in the folder `OUTPUT`. We put all the scripts, that is, **table1-4.py** and **table6.py** in one script **runtotal.sh**. You can run **runtotal.sh** and get the results in all tables of the paper, and the execution will cost around 3 minutes.

```
$ bash runtotal.sh
```

The results will be saved in **RESULTS/table1-4,6.txt**, which is in the same format as that in the paper.

It's worth noting that the above-mentioned time is obtained in our machine with 56 CPU cores.

4 Data

- Dataset: The folder `DataSet` contains all the input data of the model.
- OUTPUT: The folder `OUTPUT` contains the commit messages generated by FIRA and other compared approaches and ground truth.
- HumanEvaluation: The folder `HumanEvaluation` contains the scores of the six participants.
- RESULTS: The folder `RESULTS` contains the results of all the tables in the paper.

5 Metrics

The folder `Metrics` contains the scripts to compute the metrics to evaluate our approach, including BLEU, ROUGE-L, METEOR, and Penalty-BLEU. The scripts **talbe1-4,6.py** use the four scripts to compute the metrics.

`Bleu-B-Norm.py`, `Rouge.py`, and `Meteor.py` are from the scripts provided by Tao et al. [1], who conducted an experimental study on the evaluation of commit message generation models and found that B-Norm BLEU exhibits the most consistently with human judgements on the quality of commit messages.

6 Reference

[1] Tao W, Wang Y, Shi E, et al. On the Evaluation of Commit Message Generation Models: An Experimental Study[C]//2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2021: 126-136.