The artifact provides a Docker image to reproduce the experimental results in the paper "Revisiting Learning-based Commit Message Generation". The Docker image is in https://hub.docker.com/layers/djhao/icse-msg-study/latest/images/sha256-c879d3dbb703e4a0d59f4aad54e95c65984203feefd08980ed560a4a9adae921 and we also provide a zenodo link https://doi.org/10.5281/zenodo.7042270.

1 Requirements

- Docker

2 Preparation

- Pull the image

```
$ docker pull djhao/icse-msg-study:latest
```

The size of the image is around 2.5 GB.

- Run the image as a container

```
$ docker run -it djhao/icse-msg-study:latest
```

- 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-msg-study:latest
```

3 Data

- `CommitMessages` contains the generated commit messages used and analyzed in each RQ.
- `Scripts` contains the scripts to reproduce the results of this paper, which analyze the commit messages in the folder `CommitMessages`.

4 Reproduction

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

We name the scripts to get the Table ***X***/Figure ***X*** presented in RQ ***Y*** of the paper as `get_rqY_tableX.py`/`get_rqY_figureX.py`, which is placed in the folder `Scripts`. For example, the Figure 5 is presented in RQ2, so the script to get it is `get_rq2_table5.py`. You can run each script to get the corresponding table/figure, which will be saved in `./TablesAndFigures/rqY_tableX.tex` or `./TablesAndFigures/rqY_figureX.png`. The table is in the format of *latex*, and can be compiled to *pdf*, and the package *multirow* and *booktabs* are needed for compilation.

```
$ python get_rqY_tableX.py
```

or

```
$ python get_rqY_figureX.py
```

In addition, we put the commands to execute all the scripts to one single script `run_total.sh`, and you can get all the tables/figures of the paper by running, and the execution will cost around 14 minutes.

```
$ bash run_total.sh
```