Guides on How to train and use model with BertSum
(I was using a local system of Ubuntu 20.04 with an independent graphical card)

Step 1: clone the BertSum Project from Github

```
git clone https://github.com/nlpyang/BertSum.git
```

Step2: download the CNN stories from here
https://cs.nyu.edu/~kcho/DMQA/

Step3: download the Stanford CoreNLP package here
https://stanfordnlp.github.io/CoreNLP/

Step4: add the path to the Stanford CoreNLP package to your bash_profile:

```
export CLASSPATH=/path/to/stanford-corenlp-4.0.0/stanford-corenlp-4.0.0.jar
```

Step5: Run the following three lines of code to format and the raw cnn-storyies data into PyTorch files

```
python preprocess.py -mode tokenize -raw_path RAW_PATH -save_path T
OKENIZED_PATH
```

-RAW_PATH is the directory containing story files (../raw_stories)
-JSON_PATH is the target directory to save the generated json files (../merged_stories_tokenized)

```
python preprocess.py -mode format_to_lines -raw_path RAW_PATH -save_path
JSON_PATH -map_path MAP_PATH -lower
```

-RAW_PATH is the directory containing tokenized files (../merged_stories_tokenized),
-JSON_PATH is the target directory to save the generated json files (../json_data/cnndm)
-MAP_PATH is the directory containing the urls files (../urls)

```
python preprocess.py -mode format_to_bert -raw_path JSON_PATH -save_path
BERT_DATA_PATH -oracle_mode greedy -n_cpus 4 -log_file ../logs/preprocess.log
```

-JSON_PATH is the directory containing json files (../json_data)
-BERT_DATA_PATH is the target directory to save the generated binary files (../bert_data)

Step6: Model Training (train with Bert + RNN model):

First run: For the first time, you should use single-GPU, so the code can download the BERT model. Change `-visible_gpus 0,1,2 -gpu_ranks 0,1,2 -world_size 3` to `-visible_gpus 0 -gpu_ranks 0 -world_size 1`, after downloading, you could kill the process and rerun the code with multi-GPUs.

```
python train.py -mode train -encoder rnn -dropout 0.1 -bert_data_path
../bert_data/cnndm -model_path ../models/bert_rnn -lr 2e-3 -visible_gpus 0,1,2  -gpu_ranks
0,1,2 -world_size 3 -report_every 50 -save_checkpoint_steps 1000 -batch_size 3000
-decay_method noam -train_steps 50000 -accum_count 2 -log_file ../logs/bert_rnn
-use_interval true -warmup_steps 10000 -rnn_size 768 -dropout 0.1
```

Step7: Using the trained model:

1. clone the repo

   git clone https://github.com/nlpyang/PreSumm

2. After step6 is finished, you should get a folder named bert_rnn under models inside your BertSum Project folder, copy the file inside it and paste it to the models inside your PreSumm folder

```
mv -r /path/to/BertSum/models/bert_rnn /path/to/PreSumm/models
```

3. run the following code to use your model:

```
python train.py -mode test_text -text_src
/path/to/PreSumm/raw_data/temp.raw_src -test_from
/path/to/PreSumm/models/model_step_148000.pt -task abs
```

-please make sure your remove all the new line characters when you paste you the article into the temp.raw_src file