XLNet Korean Pretraining with GPU

Presented by
Minho Ryu, Hanyang University
ryumin93@naver.com

Content

1. Data Collection: kowiki
2. Data Preprocessing: WikiExtractor
3. Sentencepiece Model: sentencepiece
4. Convert data to TFRecords file: data_utils.py
5. Pretrain XLNet with GPU: train_gpu.py

Data Collection: kowiki

1. Go to https://dumps.wikimedia.org/kowiki/latest/
2. Download kowiki-latest-pages-articles.xml.bz2

Index of /kowiki/latest/

../		
kowiki-latest-abstract.xml.gz	06-Jul-2019 01:44	67223120
kowiki-latest-abstract.xml.gz-rss.xml	06-Jul-2019 01:44	760
kowiki-latest-all-titles-in-ns0.gz	05-Jul-2019 17:05	5322995
kowiki-latest-all-titles-in-ns0.gz-rss.xml	05-Jul-2019 17:05	775
kowiki-latest-all-titles.gz	05-Jul-2019 17:05	11437968
kowiki-latest-all-titles.gz-rss.xml	05-Jul-2019 17:05	754
kowiki-latest-category.sql.gz	02-Jul-2019 12:42	3037663
kowiki-latest-category.sql.gz-rss.xml	05-Jul-2019 17:02	760
kowiki-latest-categorylinks.sql.gz	02-Jul-2019 12:52	79255658
kowiki-latest-categorylinks.sql.gz-rss.xml	05-Jul-2019 17:03	775
kowiki-latest-change_tag.sql.gz	02-Jul-2019 12:44	26090071
kowiki-latest-change_tag.sql.gz-rss.xml	05-Jul-2019 17:04	766
kowiki-latest-externallinks.sql.gz	02-Jul-2019 12:47	70604210
kowiki-latest-externallinks.sql.gz-rss.xml	05-Jul-2019 17:00	775
kowiki-latest-geo_tags.sql.gz	02-Jul-2019 12:49	1077654
kowiki-latest-geo_tags.sql.gz-rss.xml	05-Jul-2019 17:03	760
kowiki-latest-image.sql.gz	02-Jul-2019 12:45	1799655
kowiki-latest-image.sql.gz-rss.xml	05-Jul-2019 17:03	751
kowiki-latest-imagelinks.sql.gz	02-Jul-2019 12:43	18637960
kowiki-latest-imagelinks.sql.gz-rss.xml	05-Jul-2019 17:03	766
kowiki-latest-iwlinks.sql.gz	02-Jul-2019 12:45	6683888
kowiki-latest-iwlinks.sql.gz-rss.xml	05-Jul-2019 17:04	757
kowiki-latest-langlinks.sql.gz	02-Jul-2019 12:43	113853133
kowiki-latest-langlinks.sql.gz-rss.xml	05-Jul-2019 17:04	763
kowiki-latest-md5sums.txt	08-Jul-2019 06:18	2083
kowiki-latest-page.sql.gz	02-Jul-2019 12:45	70233095
kowiki-latest-page.sql.gz-rss.xml	05-Jul-2019 17:01	748
kowiki-latest-page_props.sql.gz	02-Jul-2019 12:46	14865298
kowiki-latest-page_props.sql.gz-rss.xml	05-Jul-2019 17:02	766
kowiki-latest-page_restrictions.sql.gz	02-Jul-2019 12:47	39087
kowiki-latest-page_restrictions.sql.gz-rss.xml	05-Jul-2019 16:59	787
kowiki-latest-pagelinks.sql.gz	02-Jul-2019 12:51	229958719
kowiki-latest-pagelinks.sql.gz-rss.xml	05-Jul-2019 17:00	763
kowiki-latest-pages-articles-multistream-index...>	03-Jul-2019 15:19	11646532
kowiki-latest-pages-articles-multistream-index...>	08-Jul-2019 10:59	835
kowiki-latest-pages-articles-multistream.xml.bz2	03-Jul-2019 15:19	675610537
kowiki-latest-pages-articles-multistream.xml.bz..>	08-Jul-2019 10:59	817
kowiki-latest-pages-articles.xml.bz2	03-Jul-2019 05:22	622843843
kowiki-latest-pages-articles.xml.bz2-rss.xml	06-Jul-2019 01:45	781

Data Preprocessing: WikiExtractor

1. Go to https://github.com/attardi/wikiextractor and Download code
2. Revise WikiExtractor.py
3. Run WikiExtractor.py
4. Merge files

Data Preprocessing: WikiExtractor

Revise WikiExtractor.py: split sentences with enter and add <eop> at the end of each paragrach

```
587             if out == sys.stdout:   # option -a or -o -
588                 header = header.encode('utf-8')
589             out.write(header)
590             for line in text:
591                 if out == sys.stdout:   # option -a or -o -
592                     line = line.encode('utf-8')
593                 out.write(line)
594                 out.write('\n')
595             out.write(footer)
```

Original WikiExtractor.py

```
70  from nltk.tokenize import sent_tokenize

588             if out == sys.stdout:   # option -a or -o -
589                 header = header.encode('utf-8')
590             first = True
591             for line in text:
592                 if first:
593                     first = False
594                     continue
595                 if out == sys.stdout:   # option -a or -o -
596                     line = line.encode('utf-8')
597                 sents = sent_tokenize(line)
598                 if len(sents) == 0:
599                     continue
600                 for sent in sents:
601                     out.write('\n')
602                     out.write(sent)
603                 out.write('<eop>')
604             out.write('\n')
```

Revised WikiExtractor.py

Data Preprocessing: WikiExtractor

Run WikiExtractor.py

```
$ python WikiExtractor.py --output ${output_dir} ${path_to_xml_file}
```

Data Preprocessing: WikiExtractor

Merge files

```
$ python merge_all.py --path ${output_dir} \
  --output ${output_file_name}
```

```python
import os
import glob
import argparse

def main():
    p = argparse.ArgumentParser()
    p.add_argument('--path', type=str)
    p.add_argument('--output', type=str)
    args = p.parse_args()
    files = glob.glob(os.path.join(args.path, '**/wiki_*'), recursive=True)
    out = open(args.output, "w")
    for file in files:
        with open(file, "r") as f:
            for line in f:
                out.write(line)

    out.close()


if __name__ == "__main__":
    main()
```

merge_all.py

Sentencepiece Model: sentencepiece

Train spiece model

```
$ python spiece.py
```

Then, [spiece.model, spiece.vocab] will be produced

```
import sentencepiece as spm

spm.SentencePieceTrainer.train('--input=${path_to_merged_file} '
                               '--model_prefix=sp10m.cased.v3 '
                               '--vocab_size=50000 '
                               '--character_coverage=0.99995 '
                               '--model_type=unigram '
                               '--control_symbols=<cls>,<sep>,<pad>,<mask>,<eod> '
                               '--user_defined_symbols=<eop>,.,(,),",-,_,£,€ '
                               '--shuffle_input_sentence '
                               '--input_sentence_size=10000000 ')
```

spiece.py

Convert data to TFRecords file: data_utils.py

Run data_utils.py

```
$ python data_utils.py \
  --bsz_per_host=${batch_size} \
  --num_core_per_host=${number_of_gpus} \
  --seq_len=512 \
  --reuse_len=256 \
  --input_glob=${path_to_merged_file} \
  --save_dir=${save_dir} \
  --num_passes=20 \
  --bi_data=True \
  --sp_path=spiece.model \
  --mask_alpha=6 \
  --mask_beta=1 \
  --num_predict=85
```

Then, [corpus_info.json, tfrecords] will be produced

Pretrain XLNet with GPU: train_gpu.py

Run train_gpu.py

```
$ python train.py \
  --record_info_dir=${save_dir}/tfrecords \
  --model_dir=${output_dir} \
  --train_batch_size=${batch_size} \
  --save_steps=${save_every} \
  --seq_len=512 \
  --reuse_len=256 \
  --mem_len=384 \
  --perm_size=256 \
  --n_layer=24 \
  --d_model=1024 \
  --d_embed=1024 \
  --n_head=16 \
  --d_head=64 \
  --d_inner=4096 \
  --untie_r=True \
  --mask_alpha=6 \
  --mask_beta=1 \
  --num_predict=85
```

Everything is done. Let's wait until it is finished!

Thank you !

Presented by
Minho Ryu, Hanyang University
ryumin93@naver.com