BPE Tokenization

I used subword-nmt packages files to set up and install libraries, also used fairseq.git, mosedecoder. Then utilized dataset generated by my teammate, original sentences and grammar-corrected sentences. Ran subword-nmt commands to iterate and generate files, first learn bpe and then apply bpe, use different source files including original, corrected files with training, validation and test categories. Finally, four output files generated include lang8-trainBPE.en, lang8-train.gec, lang8-validBPE.en, lang8-validBPE.gec.

These commands are required because a lot of set up and installation involved, and the tokenization process is not done by only one line of code. One file for each vocabulary is generated to match the number of source files. BPE learning is done with training set, applied to validation and test set.

APPENDIX

Python code used in Google Colab to implement BPE tokenization for original and corrected dataset.

```
!pip install subword-nmt
!pip install https://github.com/rsennrich/subword-nmt/archive/master.zip
!git clone https://github.com/pytorch/fairseq.git
!ls
%cp fairseq/examples/translation/prepare-iwslt14.sh prepare-iwslt14.sh
!rm -r fairseq
!chmod 777 prepare-iwslt14.sh
!./prepare-iwslt14.sh
from google.colab import drive
drive.mount('/content/drive')
!ls
%cd drive
%cd 'My Drive'
!ls
%cp lang8-test.en /content/
%cp lang8-test.gec /content/
%cp lang8-train.en /content/
%cp lang8-train.gec /content/
%cp lang8-valid.en /content/
%cp lang8-valid.gec /content/
%cd ..
%cd ..
!subword-nmt learn-bpe -s 10000 < lang8-train.en > cs410output.en
!subword-nmt learn-bpe -s 10000 < lang8-train.gec > cs410output.gec
!subword-nmt apply-bpe -c cs410output.en < lang8-test.en > new-lang8-test1.en
!subword-nmt apply-bpe -c cs410output.en < lang8-valid.en > new-lang8-test.en
!subword-nmt apply-bpe -c cs410output.gec < lang8-valid.gec > new-lang8-test.gec
```

```
!cat cs410output.en > head
!cat lang8-train.en lang8-train.gec | subword-nmt learn-bpe -s 10000 -o cs410output.en
!cat lang8-train.en lang8-train.gec | subword-nmt learn-bpe -s 10000 -o cs410output.gec
!subword-nmt apply-bpe -c cs410output.en < lang8-train.en | subword-nmt get-vocab > vocab_file.en
!subword-nmt apply-bpe -c cs410output.gec < lang8-train.gec | subword-nmt get-vocab > vocab_file.gec
!subword-nmt apply-bpe -c cs410output.en < lang8-valid.en | subword-nmt get-vocab > vocab_filevalid.en
!subword-nmt apply-bpe -c cs410output.gec < lang8-valid.gec | subword-nmt get-vocab > vocab_filevalid.gec
!subword-nmt apply-bpe -c cs410output.en --vocabulary vocab_file.en --vocabulary-threshold 50 < lang8-train.en > lang8-trainBPE.en
!subword-nmt apply-bpe -c cs410output.gec --vocabulary vocab_file.gec --vocabulary-threshold 50 < lang8-train.gec > lang8-trainBPE.gec
!subword-nmt apply-bpe -c cs410output.en --vocabulary vocab_file.en --vocabulary-threshold 50 < lang8-valid.en > lang8-validBPE.en
!subword-nmt apply-bpe -c cs410output.gec --vocabulary vocab_file.gec --vocabulary-threshold 50 < lang8-valid.gec > lang8-validBPE.gec
from google.colab import files
files.download('lang8-trainBPE.en')
files.download('lang8-trainBPE.gec')
files.download('lang8-validBPE.en')
files.download('lang8-validBPE.gec')
```