

TRANSFORMER

1. Use a **OpenNMT** library <https://github.com/OpenNMT/OpenNMT-py> and follow their instructions to setup and configure Transformer FP32 model.
2. Get **English-German - Transformer** model from here <https://s3.amazonaws.com/opennmt-models/transformer-ende-wmt-pyOnmt.tar.gz> and get **WMT** datasets with shared **SentencePiece** model from here https://s3.amazonaws.com/opennmt-trainingdata/wmt_ende_sp.tar.gz.
3. Details of inference run are described here: <https://opennmt.net/OpenNMT-py/options/translate.html>

A. For example, the following script can be used for inference run:

```
## translate
python3 translate.py -model <path-to-model> \
    -src <path-to-source> \
    -output <name-of-output> \
    -replace_unk \
    -verbose \
    -report_bleu \
    -report_rouge \
    -report_time
```

i.

4. To avoid dependency of the evaluation results on the selected tokenization scheme, please use **spm decode** to detokenize the output of translation from here <https://github.com/google/sentencepiece>, and then to calculate **BLEU** score, please apply **multi-bleu.perl** script from here <https://github.com/moses-smt/mosesdecoder>

A. If obtained BLEU score is **26.98** for WMT14 dataset and **28.09** for WMT17 dataset, then reference model is setup properly.

5. Substitute **<multi_headed_attn.py>** in **../onmt/modules/** folder with the file provided in

this package.

- A. Comment/uncomment the appropriate part of the code to reproduce the specific task and method (REXP / 2D LUT) for selected precision (from source code line 256~). For example:

- i. For **2D LUT** method precision **int16**

- `attn = softmax_LUT(scores, exp_LUT_1x101_int16, Softmax_LUT_11x60_int16, 32768)`

- ii. For **REXP** method

- `attn = softmax_LUT_rexp(scores, -1)`

6. Put files **<icml2020_LUT.py>** and **<iv_rexp_LUT.py>** in the main folder of Transformer

- A. Comment/uncomment the appropriate part of the code to reproduce the **REXP** method for selected precision (from source code **<iv_rexp_LUT.py>** line 310~). For example:

- i. **# int16 case (1x13 1x16)**

1. `rexp_LUT = torch.torch.ShortTensor(rexp_LUT_1x13_int16)`
2. `scale = 32768`
3. `expln_LUT = torch.torch.ShortTensor(expln_LUT_1x16_int16)`
4. `scale_eln = 32768`

- ii. **# uint8 case (1x8 1x16)**

1. `rexp_LUT = torch.torch.ShortTensor(rexp_LUT_1x8_uint8)`
2. `scale = 256`
3. `expln_LUT = torch.torch.ShortTensor(expln_LUT_1x16_uint8)`
4. `scale_eln = 256`

7. Use **<test_wmt_2014.sh>** or **<test_wmt_2017.sh>** to reproduce the results showed in our paper.