# 03_interact

April 2, 2024

[ ]:
```
from transformers import PreTrainedTokenizerFast
from transformer_mt.modeling_transformer import TransfomerEncoderDecoderModel
```

```
c:\Users\Nech\anaconda3\envs\comp5500-hw6\Lib\site-packages\tqdm\auto.py:21:
TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

[ ]:
```
# You probably need to modify the paths to your tokenizers
# For tokenizers, you need to provide the path to a directory that contains the␣
 ↪tokenizer.json file
# For model, you need to provide the path to a directory that contains the␣
 ↪model.pt file

source_tokenizer = PreTrainedTokenizerFast.from_pretrained("../en_de_output_dir/
 ↪en_tokenizer")
target_tokenizer = PreTrainedTokenizerFast.from_pretrained("../en_de_output_dir/
 ↪de_tokenizer")

model = TransfomerEncoderDecoderModel.from_pretrained("../en_de_output_dir")
model.eval()
```

[ ]:
```
TransfomerEncoderDecoderModel(
  (positional_emb): Embedding(128, 512)
  (encoder_embeddings): Embedding(32000, 512)
  (decoder_embeddings): Embedding(32000, 512)
  (out_proj): Linear(in_features=512, out_features=32000, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (encoder): ModuleList(
    (0-5): 6 x TransformerEncoderLayer(
      (self_attention): MultiHeadAttention(
        (k): Linear(in_features=512, out_features=512, bias=True)
        (q): Linear(in_features=512, out_features=512, bias=True)
        (v): Linear(in_features=512, out_features=512, bias=True)
        (mix): Linear(in_features=512, out_features=512, bias=True)
      )
      (att_layer_norm): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
      (fcn): Sequential(
```

```
          (0): Linear(in_features=512, out_features=2048, bias=True)
          (1): ReLU()
          (2): Linear(in_features=2048, out_features=512, bias=True)
        )
        (fcn_layer_norm): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
        (dropout): Dropout(p=0.1, inplace=False)
      )
    )
    (decoder): ModuleList(
      (0-5): 6 x TransformerDecoderLayer(
        (self_attention): MultiHeadAttention(
          (k): Linear(in_features=512, out_features=512, bias=True)
          (q): Linear(in_features=512, out_features=512, bias=True)
          (v): Linear(in_features=512, out_features=512, bias=True)
          (mix): Linear(in_features=512, out_features=512, bias=True)
        )
        (corss_attention): MultiHeadAttention(
          (k): Linear(in_features=512, out_features=512, bias=True)
          (q): Linear(in_features=512, out_features=512, bias=True)
          (v): Linear(in_features=512, out_features=512, bias=True)
          (mix): Linear(in_features=512, out_features=512, bias=True)
        )
        (att_layer_norm): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
        (cross_att_layer_norm): LayerNorm((512,), eps=1e-05,
elementwise_affine=True)
        (fcn_layer_norm): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
        (fcn): Sequential(
          (0): Linear(in_features=512, out_features=2048, bias=True)
          (1): ReLU()
          (2): Linear(in_features=2048, out_features=512, bias=True)
        )
        (dropout): Dropout(p=0.1, inplace=False)
      )
    )
)
```

### 0.1 Task

Try out your model. Feel free to use your own sentences and to compare the model outputs to Google Translate. Find at least three sentences that are translated well, and one that is translated badly.

Feel free to change beam size and max_length parameters.

[ ]:
```
input_ids = source_tokenizer.encode("large language models.",␣
 ↪return_tensors="pt")
output_ids = model.generate(
    input_ids,
```

```
    max_length=10,
    bos_token_id=target_tokenizer.bos_token_id,
    eos_token_id=target_tokenizer.eos_token_id,
    pad_token_id=target_tokenizer.pad_token_id,
)
target_tokenizer.decode(output_ids[0])
```

[ ]: 'Große Sprach modelle.'