Exercise 8.01 (Teacher)

Resource(s)

- Jay Alamaar (Transformers) (https://www.youtube.com/watch?v=-QH8fRhqFHM&t=2s)

Open in Colab

(https://colab.research.google.com/github/jalammar/jalammar.github.io/blob/master/notebooks/Simple_Transforme

```
In [ ]: !pip install transformers
```

```
Collecting transformers
  Using cached transformers-4.12.5-py3-none-any.whl (3.1 MB)
Requirement already satisfied: packaging>=20.0 in /apps/tensorflow/2.4.1.cuda
11/lib/python3.8/site-packages (from transformers) (20.9)
Requirement already satisfied: regex!=2019.12.17 in /apps/tensorflow/2.4.1.cu
da11/lib/python3.8/site-packages (from transformers) (2021.3.17)
Requirement already satisfied: tqdm>=4.27 in /apps/tensorflow/2.4.1.cuda11/li
b/python3.8/site-packages (from transformers) (4.57.0)
Collecting sacremoses
  Using cached sacremoses-0.0.46-py3-none-any.whl (895 kB)
Requirement already satisfied: numpy>=1.17 in /apps/tensorflow/2.4.1.cuda11/l
ib/python3.8/site-packages (from transformers) (1.19.5)
Requirement already satisfied: requests in /apps/tensorflow/2.4.1.cuda11/lib/
python3.8/site-packages (from transformers) (2.25.1)
Collecting filelock
  Downloading filelock-3.4.0-py3-none-any.whl (9.8 kB)
Collecting huggingface-hub<1.0,>=0.1.0
  Using cached huggingface_hub-0.1.2-py3-none-any.whl (59 kB)
Collecting tokenizers<0.11,>=0.10.1
  Using cached tokenizers-0.10.3-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x8
6_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (3.3 MB)
Requirement already satisfied: pyyaml>=5.1 in /apps/tensorflow/2.4.1.cuda11/l
ib/python3.8/site-packages (from transformers) (5.3.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /apps/tensorflo
w/2.4.1.cuda11/lib/python3.8/site-packages (from huggingface-hub<1.0,>=0.1.0-
>transformers) (3.7.4.3)
Requirement already satisfied: pyparsing>=2.0.2 in /apps/tensorflow/2.4.1.cud
a11/lib/python3.8/site-packages (from packaging>=20.0->transformers) (2.4.7)
Requirement already satisfied: chardet<5,>=3.0.2 in /apps/tensorflow/2.4.1.cu
da11/lib/python3.8/site-packages (from requests->transformers) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in /apps/tensorflow/2.4.1.c
uda11/lib/python3.8/site-packages (from requests->transformers) (2021.5.30)
Requirement already satisfied: idna<3,>=2.5 in /apps/tensorflow/2.4.1.cuda11/
lib/python3.8/site-packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /apps/tensorflow/2.4.
1.cuda11/lib/python3.8/site-packages (from requests->transformers) (1.26.3)
Requirement already satisfied: click in /apps/tensorflow/2.4.1.cuda11/lib/pyt
hon3.8/site-packages (from sacremoses->transformers) (7.1.2)
Requirement already satisfied: joblib in /apps/tensorflow/2.4.1.cuda11/lib/py
thon3.8/site-packages (from sacremoses->transformers) (1.0.1)
Requirement already satisfied: six in /apps/tensorflow/2.4.1.cuda11/lib/pytho
n3.8/site-packages (from sacremoses->transformers) (1.15.0)
Installing collected packages: filelock, tokenizers, sacremoses, huggingface-
hub, transformers
Successfully installed filelock-3.4.0 huggingface-hub-0.1.2 sacremoses-0.0.46
tokenizers-0.10.3 transformers-4.12.5
```

Setup and Tokenization

Declare and assign values to the tokenizer and model variables. Distilgpt2 is a smaller version of the GPT2 model.

In [ ]:

```
from transformers import AutoTokenizer, AutoModelForCausalLM
tokenizer = AutoTokenizer.from_pretrained("distilgpt2")
model = AutoModelForCausalLM.from_pretrained("distilgpt2", output_hidden_state
s = True)
```

```
2021-11-29 11:06:49.786124: I tensorflow/stream_executor/platform/default/dso
_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
```

Assign a value to the text string to be tokenized, and then present it to the model's generate function. The model correctly returns 'Redemption' as the next word in the sequence.

In [ ]:

```
text = "The Shawshank"

# Tokenize the input string
input = tokenizer.encode(text, return_tensors="pt")

# Run the model
output = model.generate(input, max_length = 5, do_sample = False)

# Print the output
print('\n',tokenizer.decode(output[0]))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

 The Shawshank Redemption
```

In [ ]:

```
# Print the token ides (of the input and output)
output
```

Out[ ]:

```
tensor([[  464, 18193,  1477,   962, 34433]])
```

From words to vectors and back

In [ ]:

```
# Print the input token ids
text = "The Shawshank"
input = tokenizer(text, return_tensors="pt")['input_ids']
input
```

Out[ ]:

```
tensor([[  464, 18193,  1477,   962]])
```

In [ ]:

```
tokenizer.convert_ids_to_tokens(input[0])
```

Out[ ]:

```
['The', 'ĠShaw', 'sh', 'ank']
```

Breathe meaning into numbers (Embedding)

This model has a vocabulary of 50,257 tokens, each with an embedding of 768 numbers.

In [ ]:

```
# This is the embedding matrix of the model
model.transformer.wte # Dimensions are: (Number of tokens in vocabulary, dimen
sion of model)
```

Out[ ]:

```
Embedding(50257, 768)
```

In [ ]:

```
import tensorflow as tf
```

In [ ]:

```
# View all of the embeddings.
model.transformer.wte.weight

# View the embedding vector for token #464 ('The')
model.transformer.wte.weight[464]

# View the size of the embedding vector for token #464
len(model.transformer.wte.weight[464])
```

Out[ ]:

```
768
```

In [ ]:

```
text = "The chicken didn't cross the road because it was"

# Tokenize the input string
input = tokenizer.encode(text, return_tensors="pt")

# Run the model
output = model.generate(input, max_length = 20, do_sample = True)

# Print the output
print('\n',tokenizer.decode(output[0]))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

 The chicken didn't cross the road because it was like, "Oh wow. That's the b
est
```