```
import pandas as pd
import numpy as np
from tqdm import tqdm
from tqdm.notebook import tqdm_notebook
tqdm_notebook.pandas()
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
from tqdm import tqdm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.
```

```
! cp '/content/drive/My Drive/tweet-sentiment-extraction/preprocessed_train.csv' .
! cp '/content/drive/My Drive/tweet-sentiment-extraction/preprocessed_test.csv' .
#! cp '/content/drive/My Drive/tweet-sentiment-extraction/train.csv' .
#! cp '/content/drive/My Drive/tweet-sentiment-extraction/test.csv' .
```

```
train_df = pd.read_csv('preprocessed_train.csv')
test_df = pd.read_csv('preprocessed_test.csv')
```

```
train_df.shape,test_df.shape
```

```
((27469, 7), (3534, 3))
```

```
train_df.sample(5)
```

	textID	text	selected_text	sentiment	misspelled	start_indices
2588	c2112e0f01	i am so jealous	jealous	negative	No	3
11232	4d352ca9b1	lost a battle with the couch phone has been bl...	lost a battle with the couch phone has been bl...	neutral	No	0

```
test_df.sample(5)
```

	textID	text	sentiment

```
#train_df[train_df.end_indices<train_df.start_indices]
```

1709	fa89412e2a	living it up at empire hotel free bottle servi	positive

```
#train_df.loc[6393,'selected_text'] = 'amay the be with'
#train_df.loc[13668,'selected_text'] = 'utter curse these'
```

2222	131bc4a304	let me know how it goes babe good luck	positive

```
train_df.shape
```

```
(27469, 7)
```

```
train_df.head()
```

	textID	text	selected_text	sentiment	misspelled	start_indices	end_i
0	cb774db0d1	i would have responded if i were going	i would have responded if i were going	neutral	No	0	
1	549e992a42	sooo sad i will miss you here	sooo sad	negative	No	0	

```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: sentencepiece!=0.1.92 in /usr/local/lib/python3.6/dis
Requirement already satisfied: packaging in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: dataclasses; python_version < "3.7" in /usr/local/lib
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: protobuf in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: tokenizers==0.9.2 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: filelock in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: sacremoses in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from p
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: joblib in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: click in /usr/local/lib/python3.6/dist-packages (from
```

```
from transformers import RobertaTokenizer
tokenizer = RobertaTokenizer.from_pretrained('roberta-base',add_prefix_space=True)
tokenizer.encode(' hello world'),tokenizer.encode('hello world')
```

```
([0, 20760, 232, 2], [0, 20760, 232, 2])
```

```
tokenizer.encode('positive'),tokenizer.encode('negative'),tokenizer.encode('neutral')
```

```
([0, 1313, 2], [0, 2430, 2], [0, 7974, 2])
```

```
tokenizer.decode([0,1313,2])
```

```
'<s> positive</s>'
```

```
train_df = train_df[['text','selected_text','sentiment']]
train_df.head()
```

	text	selected_text	sentiment
0	i would have responded if i were going	i would have responded if i were going	neutral
1	sooo sad i will miss you here in san diego	sooo sad	negative
2	my boss is bullying me	bullying me	negative
3	what interview leave me alone	leave me alone	negative
	cons of curse why could not they put them on		

```
from sklearn.model_selection import train_test_split
x_train, x_val , y_train , y_val = train_test_split(train_df[['text','sentiment']],train_d
x_train.shape, x_val.shape , y_train.shape , y_val.shape
```

```
((21975, 2), (5494, 2), (21975,), (5494,))
```

input_ids - Indices of input sequence tokens in the vocabulary.

The input ids are often the only required parameters to be passed to the model as input. They are token indices, numerical representations of tokens building the sequences that will be used as input by the model.

attention_mask – Mask to avoid performing attention on padding token indices. Mask values selected in [0, 1]:

1 for tokens that are not masked,

0 for tokens that are maked.

The attention mask is an optional argument used when batching sequences together. This argument indicates to the model which tokens should be attended to, and which should not.

```
MAX_LEN=92
count = x_train.shape[0]
input_ids = np.zeros((count,MAX_LEN),dtype='int32')
attention_mask = np.zeros((count,MAX_LEN),dtype='int32')
start_tokens = np.zeros((count,MAX_LEN),dtype='int32')
end_tokens = np.zeros((count,MAX_LEN),dtype='int32')
```

```
for i,each in tqdm(enumerate(x_train.values)):
  val = tokenizer.encode_plus(each[1],each[0],add_special_tokens=True,max_length=92,returr
  input_ids[i] = val['input_ids']
  attention_mask[i] = val['attention_mask']
  res = (tokenizer.encode(y_train.values[i]))
  res = res[1:-1] # to ignore <s> and </s>
  st = tf.where(val['input_ids']==res[0]).numpy()[0][1]
  start_tokens[i][st]=1
  ed = tf.where(val['input_ids']==res[-1]).numpy()[0][1]
  end_tokens[i][ed]=1
```

```
21975it [00:34, 640.21it/s]
```

```
input_ids.shape,attention_mask.shape,start_tokens.shape,end_tokens.shape
```

```
((21975, 92), (21975, 92), (21975, 92), (21975, 92))
```

```
#Visualize the results
import random

for _ in range(35,40):
  i = random.randint(0,x_train.shape[0])
  print(x_train.iloc[i]['text'],'>>>>',y_train.iloc[i])
  print('Input ids',input_ids[i])
  print('attention mask',attention_mask[i])
  print('Start tokens',start_tokens[i])
  print('end   Tokens',end_tokens[i])
  print('*'*50)
```

```
up earlier because of a stupid orthadontist appointment >>>> stupid
Input ids [    0  2430     2     2    62   656   142     9    10 12103 25691   625
  2533   661  4916     2     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1]
attention mask [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
**************************************************
happy birthday >>>> happy
Input ids [   0 1313    2    2 1372 4115    2    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1]
attention mask [1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
**************************************************
iiight thanks >>>> iiight thanks
Input ids [    0  1313     2     2 42661  5971  2446     2     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1]
attention mask [1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
**************************************************
yes she got all lovey with him he returned the love another also know y ur mad at he
Input ids [    0  1313     2     2  4420    79   300    70   657   219    19   123
    37  1835     5   657   277    67   216  1423 11540  7758    23    69
     2     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
```

```
           1     1     1     1     1     1     1     1     1     1     1     1
           1     1     1     1     1     1     1     1     1     1     1     1
           1     1     1     1     1     1     1     1     1     1     1     1
           1     1     1     1     1     1     1     1]
     attention mask [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     Start tokens [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

```
MAX_LEN=92
count = y_val.shape[0]
input_ids_val = np.zeros((count,MAX_LEN),dtype='int32')
attention_mask_val = np.zeros((count,MAX_LEN),dtype='int32')
start_tokens_val = np.zeros((count,MAX_LEN),dtype='int32')
end_tokens_val = np.zeros((count,MAX_LEN),dtype='int32')
```

```
for i,each in tqdm(enumerate(x_val.values)):
  val = tokenizer.encode_plus(each[1],each[0],add_special_tokens=True,max_length=92,returr
  input_ids_val[i] = val['input_ids']
  attention_mask_val[i] = val['attention_mask']
  res = (tokenizer.encode(y_val.values[i]))
  res = res[1:-1] # to ignore <s> and </s>
  st = tf.where(val['input_ids']==res[0]).numpy()[0][1]
  start_tokens_val[i][st]=1
  ed = tf.where(val['input_ids']==res[-1]).numpy()[0][1]
  end_tokens_val[i][ed]=1
```

```
     5494it [00:08, 656.59it/s]
```

```
input_ids_val.shape,attention_mask_val.shape,start_tokens_val.shape,end_tokens_val.shape
```

```
     ((5494, 92), (5494, 92), (5494, 92), (5494, 92))
```

```
import random

for _ in range(35,40):
  i = random.randint(0,x_val.shape[0])
  print(x_val.iloc[i]['text'],'>>>>',y_val.iloc[i])
  print('Input ids',input_ids_val[i])
  print('attention mask',attention_mask_val[i])
  print('Start tokens',start_tokens_val[i])
  print('end   Tokens',end_tokens_val[i])
  print('*'*50)
```

```
feeling pretty good this morning lets hope it lasts through the day >>>> feeling pre
Input ids [    0  1313     2     2  2157  1256   205    42   662  8382  1034    24
 19646   149     5   183     2     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1]
attention mask [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
************************************************
my thumb hurts after breaking the nail >>>> hurts
Input ids [    0  2430     2     2   127 15459 15774    71  3433     5 13517     2
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1]
attention mask [1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
************************************************
why you never answer me >>>> why you never answer me
Input ids [   0 2430    2    2  596   47  393 1948  162    2    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1]
attention mask [1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Start tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
end   Tokens [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
************************************************
i know it curse sucks anyway you can get a fake id or something >>>> i know it curse
Input ids [    0  2430     2     2   939   216    24 26020 29384  6992    47    64
   120    10  4486 13561    50   402     2     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
```

```
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1]
     attention mask [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     Start tokens [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     end   Tokens [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     ************************************************
     which would be a lot easier to get if plane tickets to az didnt raise in price >>>>
     Input ids [    0  1313     2     2    61    74    28    10   319  3013     7   120
        114  3286  3308     7 15001 46405  1693    11   425     2     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1     1     1     1     1
          1     1     1     1     1     1     1     1]
     attention mask [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     Start tokens [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

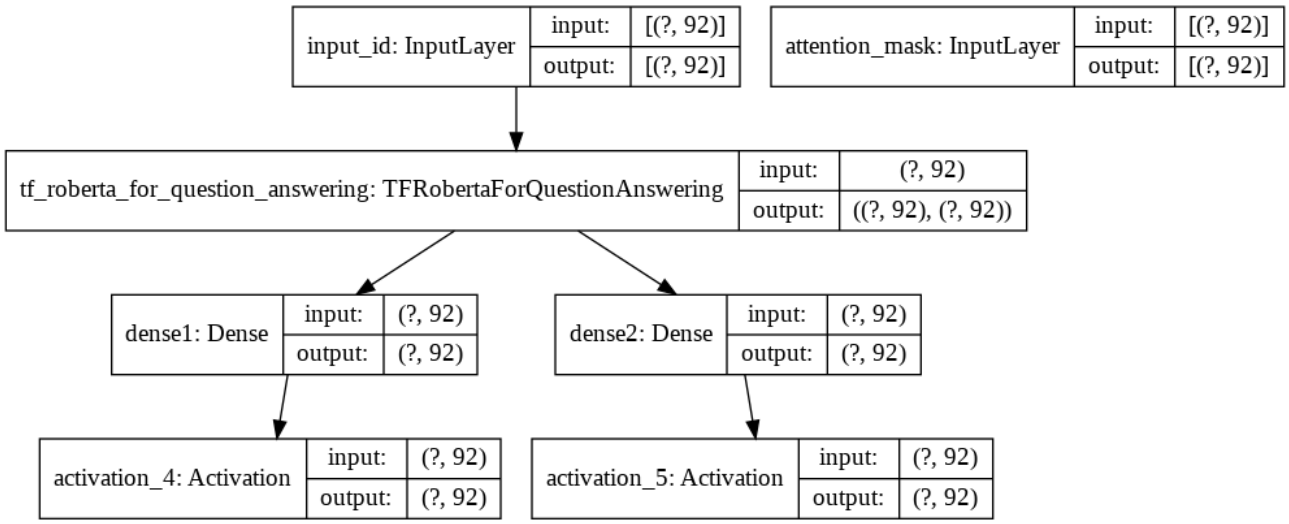
```
from transformers import TFRobertaForQuestionAnswering
roberta = TFRobertaForQuestionAnswering.from_pretrained('roberta-base')
```

```
     Some layers from the model checkpoint at roberta-base were not used when initializin
     - This IS expected if you are initializing TFRobertaForQuestionAnswering from the ch
     - This IS NOT expected if you are initializing TFRobertaForQuestionAnswering from th
     Some layers of TFRobertaForQuestionAnswering were not initialized from the model che
     You should probably TRAIN this model on a down-stream task to be able to use it for
```

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input,Softmax,Dense,Activation,Dropout


input1 = Input(shape=(MAX_LEN,),name='input_id',dtype=tf.int32)
input2 = Input(shape=(MAX_LEN,),name='attention_mask',dtype=tf.int32)
start_scores,end_scores = roberta(input1,attention_mask = input2)
dense1 = Dense(units=MAX_LEN,activation='relu',name='dense1',kernel_regularizer = tf.keras
softmax1 = Activation('softmax')(dense1)
dense2 = Dense(units=MAX_LEN,activation='relu',name='dense2',kernel_regularizer = tf.keras
softmax2 = Activation('softmax')(dense2)


model = Model(inputs=[input1,input2],outputs=[softmax1,softmax2])


model.summary()
```

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_id (InputLayer)           [(None, 92)]         0
__________________________________________________________________________________________________
attention_mask (InputLayer)     [(None, 92)]         0
__________________________________________________________________________________________________
tf_roberta_for_question_answeri ((None, 92), (None,  124647170   input_id[0][0]
                                                                 attention_mask[0][0
__________________________________________________________________________________________________
dense1 (Dense)                  (None, 92)           8556        tf_roberta_for_ques
__________________________________________________________________________________________________
dense2 (Dense)                  (None, 92)           8556        tf_roberta_for_ques
__________________________________________________________________________________________________
activation_4 (Activation)       (None, 92)           0           dense1[0][0]
__________________________________________________________________________________________________
activation_5 (Activation)       (None, 92)           0           dense2[0][0]
==================================================================================================
Total params: 124,664,282
Trainable params: 124,664,282
Non-trainable params: 0
```

```
import tensorflow as tf
tf.keras.utils.plot_model(model, 'Model.png',show_shapes=True)
```

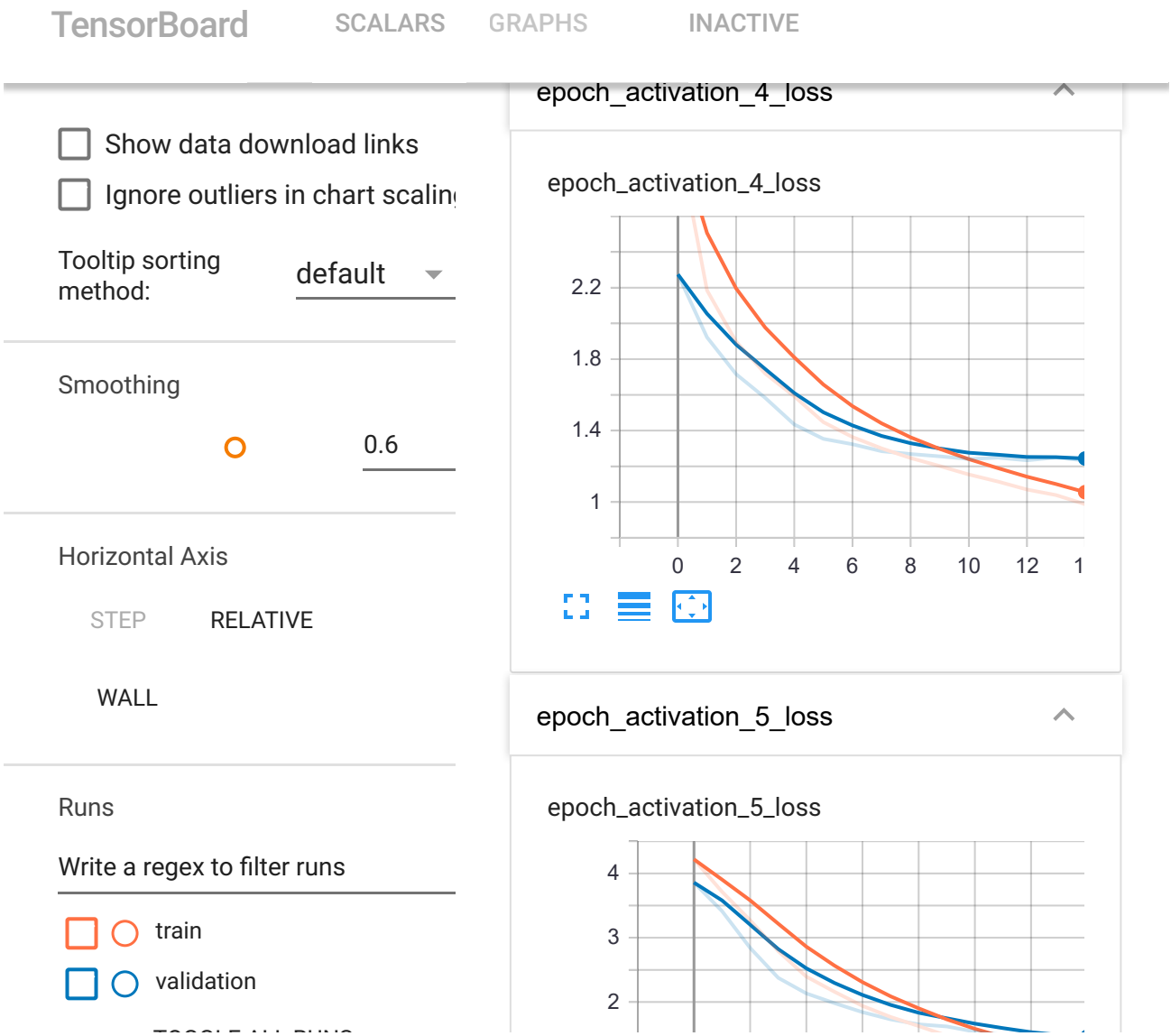



```
! rm -r '/content/checkpt'
! rm -r '/content/tensorboard_logs1'


input_data = (input_ids,attention_mask)
output_data = (start_tokens,end_tokens)

val = (input_ids_val,attention_mask_val)
output_val = (start_tokens_val,end_tokens_val)
```

```
output_val = (start_tokens_val,end_tokens_val)
val_data = (val,output_val)


%load_ext tensorboard
import datetime
import os
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.1,patience=2
log_dir= os.path.join("tensorboard_logs1" , datetime.datetime.now().strftime("%Y%m%d-%H%M%
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,histogram_freq=1, wr
! mkdir  'checkpt'
file_path = os.path.join('checkpt/model.hdf5')
checkpt_save = tf.keras.callbacks.ModelCheckpoint(filepath=file_path,save_weights_only=Tru
callbacks=[reduce_lr,tensorboard_callback,checkpt_save]


opt = tf.keras.optimizers.Adam(learning_rate=1e-5, epsilon=1e-08, clipnorm=1.0)
model.compile(optimizer=opt,loss='categorical_crossentropy')


train_dataset = tf.data.Dataset.from_tensor_slices((input_data, output_data)).shuffle(buff
val_dataset = tf.data.Dataset.from_tensor_slices(val_data).batch(32)


#model.fit(input_data,output_data,epochs=10,batch_size=128,validation_data=val_data,callba
model.fit(train_dataset,epochs=15,validation_data=val_dataset,callbacks=callbacks)
```

```
WARNING:tensorflow:Model failed to serialize as JSON. Ignoring...
Epoch 1/15
WARNING:tensorflow:Gradients do not exist for variables ['tf_roberta_for_question_an
WARNING:tensorflow:Gradients do not exist for variables ['tf_roberta_for_question_an
WARNING:tensorflow:Gradients do not exist for variables ['tf_roberta_for_question_an
WARNING:tensorflow:Gradients do not exist for variables ['tf_roberta_for_question_an
  1/344 [..............................] - ETA: 0s - loss: 9.0419 - activation_4_los
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
344/344 [==============================] - ETA: 0s - loss: 7.2790 - activation_4_los
Epoch 00001: val_loss improved from inf to 6.15025, saving model to checkpt/model.hd
344/344 [==============================] - 263s 764ms/step - loss: 7.2790 - activati
Epoch 2/15
344/344 [==============================] - ETA: 0s - loss: 5.9175 - activation_4_los
Epoch 00002: val_loss improved from 6.15025 to 5.35419, saving model to checkpt/mode
344/344 [==============================] - 261s 758ms/step - loss: 5.9175 - activati
Epoch 3/15
344/344 [==============================] - ETA: 0s - loss: 5.1881 - activation_4_los
Epoch 00003: val_loss improved from 5.35419 to 4.57568, saving model to checkpt/mode
344/344 [==============================] - 261s 760ms/step - loss: 5.1881 - activati
Epoch 4/15
344/344 [==============================] - ETA: 0s - loss: 4.5286 - activation_4_los
Epoch 00004: val_loss improved from 4.57568 to 3.97775, saving model to checkpt/mode
344/344 [==============================] - 261s 758ms/step - loss: 4.5286 - activati
Epoch 5/15
344/344 [==============================] - ETA: 0s - loss: 4.0054 - activation_4_los
Epoch 00005: val_loss improved from 3.97775 to 3.58621, saving model to checkpt/mode
344/344 [==============================] - 260s 757ms/step - loss: 4.0054 - activati
Epoch 6/15
344/344 [==============================] - ETA: 0s - loss: 3.6256 - activation_4_los
Epoch 00006: val_loss improved from 3.58621 to 3.35449, saving model to checkpt/mode
344/344 [==============================] - 261s 757ms/step - loss: 3.6256 - activati
Epoch 7/15
344/344 [==============================] - ETA: 0s - loss: 3.3217 - activation_4_los
Epoch 00007: val_loss improved from 3.35449 to 3.18402, saving model to checkpt/mode
344/344 [==============================] - 261s 760ms/step - loss: 3.3217 - activati
Epoch 8/15
344/344 [==============================] - ETA: 0s - loss: 3.0867 - activation_4_los
Epoch 00008: val_loss improved from 3.18402 to 3.03309, saving model to checkpt/mode
344/344 [==============================] - 261s 757ms/step - loss: 3.0867 - activati
Epoch 9/15


344/344 [==============================] - 261s 758ms/step - loss: 2.8904 - activati
```

```
tf.keras.backend.clear_session()
%tensorboard --logdir $log_dir --port 0
```




```
model.load_weights('checkpt/model.hdf5')
start_pred_train , end_pred_train = model.predict((input_data))
start_pred_train.shape,end_pred_train.shape
```

```
((21975, 92), (21975, 92))
```

```
strt =[]
end=[]
for i in tqdm(range(start_pred_train.shape[0])):
  s = tf.math.argmax(start_pred_train[i],axis=0).numpy()
  e = tf.math.argmax(end_pred_train[i],axis=0).numpy()
  strt.append(s)
  end.append(e)

len(strt),len(end)
```

```
100%|██████████| 21975/21975 [00:10<00:00, 2068.72it/s]
(21975, 21975)
```

```
pred_values = []
for i in tqdm(range(len(strt))):
  index1 = strt[i]
```

```
  index2 = end[i] +1
  pred = input_ids[i][index1:index2]
  mystring = tokenizer.decode(pred)
  pred_values.append(mystring)

actual_values = (y_train.values)
len(actual_values),len(pred_values)
```

```
100%|██████████| 21975/21975 [00:00<00:00, 67769.54it/s]
(21975, 21975)
```

```
x_train['selected_text'] = y_train
x_train['pred_text'] = pred_values


x_train.sample(15)
```

	text	sentiment	selected_text	pred_text
17346	i want cafe latteeeeeeeeeee	neutral	i want cafe latteeeeeeeeeee	i want cafe latteeeeeeeeeee
22824	thats ok ur a good person my idol soo good peo...	positive	good person	good person
8655	the blowout ended	neutral	the blowout ended	the blowout ended
26189	tired and gunna go to bed soon first time i ha...	negative	tired and	tired and
3162	i m totally confused and bored my life must ch...	negative	i m totally confused and bored	bored
7950	happy bank holiday	positive	happy bank holiday	happy bank holiday
9316	i have been to tara thai a few times for frien...	negative	food i had was pretty bad	had was pretty bad
4415	what an incredibly great day hahaha	positive	an incredibly great day hahaha	great
8334	i am so disgusted that my assumptions regardin...	negative	i am so disgusted	disgusted
15604	time warner talk about aol we did our presenta...	neutral	time warner talk about aol we did our presenta	time warner talk about aol we did our present

```
def jaccard(str1, str2):
    a = set(str1.lower().split())
    b = set(str2.lower().split())
    c = a.intersection(b)
    return float(len(c)) / (len(a) + len(b) - len(c))


scores=[]
for i in tqdm(range(len(actual_values))):
  scores.append(jaccard(actual_values[i],pred_values[i]))
```

```
100%|██████████| 21975/21975 [00:00<00:00, 211777.08it/s]
```

```
x_train['jaccard_score'] = scores
x_train.sample(15)
```

	text	sentiment	selected_text	pred_text	jaccard_score
26530	good morning and happy mothers day everyone	positive	happy	happy	1.000000
18407	i wish i could get sushi delivered to work	positive	wish	wish	1.000000
16574	i was talking with my best friend ï¿½ureo abou...	neutral	i was talking with my best friend ï¿½ureo abou...	i was talking with my best friend ï¿½ureo abo...	0.625000
8328	friday night and still working oh wait it is s...	neutral	friday night and still working oh wait it is s...	friday night and still working oh wait it is ...	0.846154
27187	screen on the green started yesterday ahhh i m...	negative	missed		0.000000
16195	thats not a golf buggy lol it is a australia z...	neutral	thats not a golf buggy lol it is a australia z...	thats not a golf buggy lol it is a australia ...	1.000000
132	those splinters look very painful but	negative	painful	painful	1.000000

For Training data

```
print('Mean jaccard score for neutral data:',x_train[x_train.sentiment =='neutral']['jacca
print('Mean jaccard score for positive data:',x_train[x_train.sentiment =='positive']['jac
print('Mean jaccard score for negative data:',x_train[x_train.sentiment =='negative']['jac
```

```
Mean jaccard score for neutral data: 0.9584513736408609
Mean jaccard score for positive data: 0.6704743503107081
Mean jaccard score for negative data: 0.6575072456600813
```

```
start_pred_val , end_pred_val = model.predict((val))
print(start_pred_val.shape,end_pred_val.shape)
strt_val =[]
end_val=[]
for i in tqdm(range(start_pred_val.shape[0])):
  s = tf.math.argmax(start_pred_val[i],axis=0).numpy()
  e = tf.math.argmax(end_pred_val[i],axis=0).numpy()
  strt_val.append(s)
  end_val.append(e)
print(len(strt_val),len(end_val))
```

```
pred_values_val = []
for i in tqdm(range(len(strt_val))):
```

```
  index1 = strt_val[i]
  index2 = end_val[i] +1
  pred = input_ids_val[i][index1:index2]
  mystring = tokenizer.decode(pred)
  pred_values_val.append(mystring)


actual_values_val = (y_val.values)
print(len(actual_values_val),len(pred_values_val))

scores_val=[]
for i in tqdm(range(len(actual_values_val))):
  scores_val.append(jaccard(actual_values_val[i],pred_values_val[i]))
```

```
  4%|▌         | 201/5494 [00:00<00:02, 2003.82it/s](5494, 92) (5494, 92)
100%|██████████| 5494/5494 [00:02<00:00, 2116.51it/s]
100%|██████████| 5494/5494 [00:00<00:00, 67088.94it/s]
100%|██████████| 5494/5494 [00:00<00:00, 226251.67it/s]5494 5494
5494 5494
```

For Validation Data

```
x_val['selected_text'] = y_val
x_val['predicted_text'] = pred_values_val
x_val['jaccard_score'] = scores_val
x_val.sample(10)
```

	text	sentiment	selected_text	predicted_text	jaccard_score
9350	what i meant to say at yardhouse waikiki is bd...	neutral	what i meant to say at yardhouse waikiki is bd...	what i meant to say at yardhouse waikiki is b...	1.000000
1751	do not really feel like i got a tan i gave up ...	neutral	do not really feel like i got a tan i gave up ...	do not really feel like i got a tan i gave up...	1.000000
16483	ready to go home more hrs of wrk	neutral	ready to go home more hrs of wrk	ready to go home more hrs of wrk	1.000000
24048	wx it looks like it did in ohio after a tornad...	neutral	it looks like it did in ohio after a tornado hit	wx it looks like it did in ohio after a tornado	0.818182
4688	i chilled in my room with my baby book missed	neutral	i chilled in my room with my baby book missed	i chilled in my room with my baby book missed	1.000000

```
print('Mean jaccard score for neutral data:',x_val[x_val.sentiment =='neutral']['jaccard_s
print('Mean jaccard score for positive data:',x_val[x_val.sentiment =='positive']['jaccarc
print('Mean jaccard score for negative data:',x_val[x_val.sentiment =='negative']['jaccarc
```

```
Mean jaccard score for neutral data: 0.9509615270832638
Mean jaccard score for positive data: 0.5123192610962835
Mean jaccard score for negative data: 0.4816971096798263
```