4.Roberta

January 7, 2022

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

```
Mounted at /content/drive
```

[ ]:
```
import os
import pathlib
from pathlib import Path
os.chdir("/content/drive/My Drive/Akarshan/BERT")
!ls -l
```

```
total 51350
-rw------- 1 root root  8388432 Dec 26 21:48  BERT5.hdf5
drwx------ 2 root root     4096 Dec  3 16:27  clr
-rw------- 1 root root   488058 Dec 25 21:03  Compare.ipynb
-rw------- 1 root root   258091 Dec 26 21:52 'Copy of Distllbert400000.ipynb'
-rw------- 1 root root    76810 Dec 26 21:53 'Copy of Roberta.ipynb'
drwx------ 2 root root     4096 Dec  3 16:27  Data
-rw------- 1 root root  8306584 Dec 24 07:57  DBert1hk.hdf5
-rw------- 1 root root 12719136 Dec 24 07:57  DBert4hk.hdf5
-rw------- 1 root root   251068 Dec 26 21:28  Distllbert400000.ipynb
-rw------- 1 root root   476335 Dec 26 21:08 'EDA on results.ipynb'
drwx------ 2 root root     4096 Dec 18 07:14 'misc model'
-rw------- 1 root root    78553 Dec 26 21:47  model.png
drwx------ 2 root root     4096 Dec  3 16:27  papers
-rw------- 1 root root  8306584 Dec 19 08:56  Rbert4.hdf5
-rw------- 1 root root   203164 Dec 26 21:29  Retraining.ipynb
-rw------- 1 root root    86347 Dec 19 06:43  Roberta.ipynb
-rw------- 1 root root 12719160 Dec 25 10:35  SBert.hdf5
-rw------- 1 root root   203507 Dec 25 10:50  SciBert400k.ipynb
```

[ ]:
```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
```

```
  print('Not using a high-RAM runtime')
else:
  print('You are using a high-RAM runtime!')
```

```
Your runtime has 27.3 gigabytes of available RAM

You are using a high-RAM runtime!
```

[ ]:
```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
Sun Dec 26 21:54:03 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   30C    P0    26W / 250W |      0MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

[ ]:
```
!pip install transformers
!pip install pympler
!pip install tensorflow_addons
```

```
Collecting transformers
  Downloading transformers-4.15.0-py3-none-any.whl (3.4 MB)
     || 3.4 MB 4.1 MB/s
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers) (4.8.2)
Collecting pyyaml>=5.1
```

```
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manyl
inux_2_12_x86_64.manylinux2010_x86_64.whl (596 kB)
     || 596 kB 92.1 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7
/dist-packages (from transformers) (4.62.3)
Collecting tokenizers<0.11,>=0.10.1
  Downloading tokenizers-0.10.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_6
4.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (3.3 MB)
     || 3.3 MB 65.0 MB/s
Collecting sacremoses
  Downloading sacremoses-0.0.46-py3-none-any.whl (895 kB)
     || 895 kB 90.4 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers) (2.23.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers) (1.19.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-
packages (from transformers) (21.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.2.1-py3-none-any.whl (61 kB)
     || 61 kB 673 kB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers) (3.4.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7
/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers)
(3.0.6)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers) (3.6.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.1.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub,
```

```
transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.2.1 pyyaml-6.0 sacremoses-0.0.46
tokenizers-0.10.3 transformers-4.15.0
Collecting pympler
  Downloading Pympler-1.0.1-py3-none-any.whl (164 kB)
     || 164 kB 4.1 MB/s
Installing collected packages: pympler
Successfully installed pympler-1.0.1
Collecting tensorflow_addons
  Downloading tensorflow_addons-0.15.0-cp37-cp37m-
manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
     || 1.1 MB 4.3 MB/s
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7
/dist-packages (from tensorflow_addons) (2.7.1)
Installing collected packages: tensorflow-addons
Successfully installed tensorflow-addons-0.15.0
```

[ ]:
```
import numpy as np
import pickle
import pandas as pd
import pickle
import time
import matplotlib.pyplot as plt
import seaborn as sns
from pympler import asizeof
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import transformers
from transformers import pipeline
from tensorflow.keras.layers import concatenate
from transformers import TFAutoModel, AutoTokenizer,␣
 ↪AutoConfig,TFAutoModelForSequenceClassification
from tensorflow.keras.callbacks import ModelCheckpoint
from clr import clr_callback
import tensorflow_addons as tfa
```

[ ]:
```
csvfile = 'Data//data.csv'
dropna = 'Data//datadropna.csv'
sent_data_file = 'Data//sent_data.csv'
label_file = 'Data//label.csv'
vocab_file = 'Data//vocab_tr_w.txt'
```

[ ]:
```
df = pd.read_csv(dropna,usecols = ['SBE','Label'])
# df.dropna(inplace=True)
print(df.head())
print(df.shape)
```

```
   Label                                                SBE
0      1  To facilitate an easier notation throughout th...
1      0  Therefore _MATH_ defines a special order of ti...
2      0  This is important since only _MATH_ is the rea...
3      0  Note that in all contour time-integrals we ess...
4      0  Theorem _REF_ proves the equivalence of ensemb...
(1189321, 2)
```

0.1 Generating Embeddings

[ ]:
```
# Hyperparameters form paper

epoch = 30
patience = 10
lr = 1e-6
batch_size = 32
vocab = 30526 #will have to retrain Bert so not using
MAX_LEN = 128 #not enough ram for 256
```

[ ]:
```
model_name = 'roberta-base'
config = AutoConfig.from_pretrained(model_name,trianing =False, num_labels=2 )
config.output_hidden_states = False

BERT = TFAutoModel.from_pretrained(model_name,config = config)

tokenizer = AutoTokenizer.from_pretrained(model_name,
                                          do_lower_case=True,
                                          use_fast=True,
                                          max_length=MAX_LEN,
                                          truncation=True,
                                          pad_to_max_length=True)

pipe = pipeline('feature-extraction', model=BERT,
                tokenizer=tokenizer,device=1)
```

```
Downloading:   0%|          | 0.00/481 [00:00<?, ?B/s]


Downloading:   0%|          | 0.00/627M [00:00<?, ?B/s]


Some layers from the model checkpoint at roberta-base were not used when
initializing TFRobertaModel: ['lm_head']
```

```
- This IS expected if you are initializing TFRobertaModel from the checkpoint of
a model trained on another task or with another architecture (e.g. initializing
a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing TFRobertaModel from the
checkpoint of a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
All the layers of TFRobertaModel were initialized from the model checkpoint at
roberta-base.
If your task is similar to the task the model of the checkpoint was trained on,
you can already use TFRobertaModel for predictions without further training.
```

```
Downloading:   0%|          | 0.00/878k [00:00<?, ?B/s]
```

```
Downloading:   0%|          | 0.00/446k [00:00<?, ?B/s]
```

```
Downloading:   0%|          | 0.00/1.29M [00:00<?, ?B/s]
```

[ ]:
```
batch=500
df = df.iloc[300000:400000,:]
step = int(df.shape[0]/batch)
step
```

[ ]: 200

[ ]:
```
#### getting embedding vectors as bert output ###
# pipe returns embeddings for every token in a sent
# so features[x][0] is of shape (y,768) with y tokens in xth sentance
# taking the mean for y tokens give the embedding for the xth sent in total
# saving a batch of features as feature_matrix with 768 zeors as head
import pickle
import time
count = 500+500+500
for part in range(batch):
  i = part+count
  strt = time.time()
  indx = step*part
  indy = step*(part+1)
  # print(indx,indy)
  feature_matrix = array = np.empty(768, dtype=object)
  lst = []
  features = np.array(pipe(df['SBE'].iloc[indx:indy].to_list()))

  for idx in range(np.shape(features)[0]):
    sent_mean = np.mean(features[idx][0],axis =0)
    lst.append(sent_mean)
  # print(np.shape(lst))
  feature_matrix= np.array(lst)
```

```
    # print(np.shape(feature_matrix))
    # print(feature_matrix)

    with open('Data//embeddingRo//embeddings'+str(i),'wb') as f:
      pickle.dump(feature_matrix,f)

    print(f'Part {part+1} of {batch} done. in {(time.time()-strt)/60:.2f} min')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:17:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray
```

```
Part 1 of 500 done. in 1.18 min
Part 2 of 500 done. in 1.20 min
Part 3 of 500 done. in 1.22 min
Part 4 of 500 done. in 1.16 min
Part 5 of 500 done. in 1.22 min
Part 6 of 500 done. in 1.18 min
Part 7 of 500 done. in 1.32 min
Part 8 of 500 done. in 1.22 min
Part 9 of 500 done. in 1.24 min
Part 10 of 500 done. in 1.24 min
Part 11 of 500 done. in 1.23 min
Part 12 of 500 done. in 1.23 min
Part 13 of 500 done. in 1.21 min
Part 14 of 500 done. in 1.30 min
Part 15 of 500 done. in 1.31 min
Part 16 of 500 done. in 1.27 min
Part 17 of 500 done. in 1.23 min
Part 18 of 500 done. in 1.23 min
Part 19 of 500 done. in 1.16 min
Part 20 of 500 done. in 1.21 min
Part 21 of 500 done. in 1.22 min
Part 22 of 500 done. in 1.21 min
Part 23 of 500 done. in 1.25 min
Part 24 of 500 done. in 1.23 min
Part 25 of 500 done. in 1.23 min
Part 26 of 500 done. in 1.28 min
Part 27 of 500 done. in 1.26 min
Part 28 of 500 done. in 1.30 min
Part 29 of 500 done. in 1.25 min
Part 30 of 500 done. in 1.27 min
Part 31 of 500 done. in 1.26 min
Part 32 of 500 done. in 1.27 min
Part 33 of 500 done. in 1.25 min
Part 34 of 500 done. in 1.25 min
```

```
Part 35 of 500 done. in 1.23 min
Part 36 of 500 done. in 1.26 min
Part 37 of 500 done. in 1.23 min
Part 38 of 500 done. in 1.26 min
Part 39 of 500 done. in 1.24 min
Part 40 of 500 done. in 1.27 min
Part 41 of 500 done. in 1.24 min
Part 42 of 500 done. in 1.29 min
Part 43 of 500 done. in 1.31 min
Part 44 of 500 done. in 1.35 min
Part 45 of 500 done. in 1.26 min
Part 46 of 500 done. in 1.28 min
Part 47 of 500 done. in 1.25 min
Part 48 of 500 done. in 1.29 min
Part 49 of 500 done. in 1.28 min
Part 50 of 500 done. in 1.27 min
Part 51 of 500 done. in 1.27 min
Part 52 of 500 done. in 1.30 min
Part 53 of 500 done. in 1.26 min
Part 54 of 500 done. in 1.27 min
Part 55 of 500 done. in 1.21 min
Part 56 of 500 done. in 1.20 min
Part 57 of 500 done. in 1.21 min
Part 58 of 500 done. in 1.26 min
Part 59 of 500 done. in 1.19 min
Part 60 of 500 done. in 1.22 min
Part 61 of 500 done. in 1.26 min
Part 62 of 500 done. in 1.22 min
Part 63 of 500 done. in 1.20 min
Part 64 of 500 done. in 1.22 min
Part 65 of 500 done. in 1.20 min
Part 66 of 500 done. in 1.21 min
Part 67 of 500 done. in 1.22 min
Part 68 of 500 done. in 1.24 min
Part 69 of 500 done. in 1.24 min
Part 70 of 500 done. in 1.22 min
Part 71 of 500 done. in 1.20 min
Part 72 of 500 done. in 1.16 min
Part 73 of 500 done. in 1.23 min
Part 74 of 500 done. in 1.20 min
Part 75 of 500 done. in 1.21 min
Part 76 of 500 done. in 1.17 min
Part 77 of 500 done. in 1.15 min
Part 78 of 500 done. in 1.18 min
Part 79 of 500 done. in 1.21 min
Part 80 of 500 done. in 1.20 min
Part 81 of 500 done. in 1.24 min
Part 82 of 500 done. in 1.20 min
```

```
Part 83 of 500 done. in 1.19 min
Part 84 of 500 done. in 1.19 min
Part 85 of 500 done. in 1.16 min
Part 86 of 500 done. in 1.20 min
Part 87 of 500 done. in 1.18 min
Part 88 of 500 done. in 1.22 min
Part 89 of 500 done. in 1.23 min
Part 90 of 500 done. in 1.26 min
Part 91 of 500 done. in 1.26 min
Part 92 of 500 done. in 1.26 min
Part 93 of 500 done. in 1.24 min
Part 94 of 500 done. in 1.26 min
Part 95 of 500 done. in 1.22 min
Part 96 of 500 done. in 1.25 min
Part 97 of 500 done. in 1.26 min
Part 98 of 500 done. in 1.26 min
Part 99 of 500 done. in 1.23 min
Part 100 of 500 done. in 1.27 min
Part 101 of 500 done. in 1.26 min
Part 102 of 500 done. in 1.29 min
Part 103 of 500 done. in 1.26 min
Part 104 of 500 done. in 1.22 min
Part 105 of 500 done. in 1.25 min
Part 106 of 500 done. in 1.23 min
Part 107 of 500 done. in 1.28 min
Part 108 of 500 done. in 1.23 min
Part 109 of 500 done. in 1.27 min
Part 110 of 500 done. in 1.27 min
Part 111 of 500 done. in 1.24 min
Part 112 of 500 done. in 1.28 min
Part 113 of 500 done. in 1.30 min
Part 114 of 500 done. in 1.29 min
Part 115 of 500 done. in 1.26 min
Part 116 of 500 done. in 1.22 min
Part 117 of 500 done. in 1.25 min
Part 118 of 500 done. in 1.25 min
Part 119 of 500 done. in 1.25 min
Part 120 of 500 done. in 1.26 min
Part 121 of 500 done. in 1.26 min
Part 122 of 500 done. in 1.25 min
Part 123 of 500 done. in 1.26 min
Part 124 of 500 done. in 1.26 min
Part 125 of 500 done. in 1.26 min
Part 126 of 500 done. in 1.23 min
Part 127 of 500 done. in 1.28 min
Part 128 of 500 done. in 1.25 min
Part 129 of 500 done. in 1.25 min
Part 130 of 500 done. in 1.25 min
```

```
Part 131 of 500 done. in 1.27 min
Part 132 of 500 done. in 1.25 min
Part 133 of 500 done. in 1.29 min
Part 134 of 500 done. in 1.20 min
Part 135 of 500 done. in 1.24 min
Part 136 of 500 done. in 1.30 min
Part 137 of 500 done. in 1.29 min
Part 138 of 500 done. in 1.22 min
Part 139 of 500 done. in 1.29 min
Part 140 of 500 done. in 1.23 min
Part 141 of 500 done. in 1.24 min
Part 142 of 500 done. in 1.23 min
Part 143 of 500 done. in 1.24 min
Part 144 of 500 done. in 1.24 min
Part 145 of 500 done. in 1.29 min
Part 146 of 500 done. in 1.24 min
Part 147 of 500 done. in 1.28 min
Part 148 of 500 done. in 1.26 min
Part 149 of 500 done. in 1.24 min
Part 150 of 500 done. in 1.28 min
Part 151 of 500 done. in 1.26 min
Part 152 of 500 done. in 1.22 min
Part 153 of 500 done. in 1.29 min
Part 154 of 500 done. in 1.29 min
Part 155 of 500 done. in 1.30 min
Part 156 of 500 done. in 1.27 min
Part 157 of 500 done. in 1.25 min
Part 158 of 500 done. in 1.28 min
Part 159 of 500 done. in 1.26 min
Part 160 of 500 done. in 1.31 min
Part 161 of 500 done. in 1.23 min
Part 162 of 500 done. in 1.26 min
Part 163 of 500 done. in 1.21 min
Part 164 of 500 done. in 1.30 min
Part 165 of 500 done. in 1.26 min
Part 166 of 500 done. in 1.22 min
Part 167 of 500 done. in 1.25 min
Part 168 of 500 done. in 1.36 min
Part 169 of 500 done. in 1.33 min
Part 170 of 500 done. in 1.38 min
Part 171 of 500 done. in 1.34 min
Part 172 of 500 done. in 1.30 min
Part 173 of 500 done. in 1.27 min
Part 174 of 500 done. in 1.26 min
Part 175 of 500 done. in 1.29 min
Part 176 of 500 done. in 1.27 min
Part 177 of 500 done. in 1.27 min
Part 178 of 500 done. in 1.24 min
```

```
Part 179 of 500 done. in 1.32 min
Part 180 of 500 done. in 1.25 min
Part 181 of 500 done. in 1.29 min
Part 182 of 500 done. in 1.29 min
Part 183 of 500 done. in 1.24 min
Part 184 of 500 done. in 1.26 min
Part 185 of 500 done. in 1.29 min
Part 186 of 500 done. in 1.27 min
Part 187 of 500 done. in 1.23 min
Part 188 of 500 done. in 1.26 min
Part 189 of 500 done. in 1.24 min
Part 190 of 500 done. in 1.22 min
Part 191 of 500 done. in 1.25 min
Part 192 of 500 done. in 1.30 min
Part 193 of 500 done. in 1.24 min
Part 194 of 500 done. in 1.26 min
Part 195 of 500 done. in 1.25 min
Part 196 of 500 done. in 1.22 min
Part 197 of 500 done. in 1.18 min
Part 198 of 500 done. in 1.21 min
Part 199 of 500 done. in 1.20 min
Part 200 of 500 done. in 1.23 min
Part 201 of 500 done. in 1.24 min
Part 202 of 500 done. in 1.24 min
Part 203 of 500 done. in 1.23 min
Part 204 of 500 done. in 1.22 min
Part 205 of 500 done. in 1.19 min
Part 206 of 500 done. in 1.21 min
Part 207 of 500 done. in 1.23 min
Part 208 of 500 done. in 1.23 min
Part 209 of 500 done. in 1.26 min
Part 210 of 500 done. in 1.19 min
Part 211 of 500 done. in 1.24 min
Part 212 of 500 done. in 1.21 min
Part 213 of 500 done. in 1.23 min
Part 214 of 500 done. in 1.23 min
Part 215 of 500 done. in 1.19 min
Part 216 of 500 done. in 1.25 min
Part 217 of 500 done. in 1.23 min
Part 218 of 500 done. in 1.20 min
Part 219 of 500 done. in 1.19 min
Part 220 of 500 done. in 1.17 min
Part 221 of 500 done. in 1.17 min
Part 222 of 500 done. in 1.16 min
Part 223 of 500 done. in 1.17 min
Part 224 of 500 done. in 1.15 min
Part 225 of 500 done. in 1.18 min
Part 226 of 500 done. in 1.21 min
```

```
Part 227 of 500 done. in 1.22 min
Part 228 of 500 done. in 1.14 min
Part 229 of 500 done. in 1.17 min
Part 230 of 500 done. in 1.20 min
Part 231 of 500 done. in 1.25 min
Part 232 of 500 done. in 1.24 min
Part 233 of 500 done. in 1.22 min
Part 234 of 500 done. in 1.16 min
Part 235 of 500 done. in 1.20 min
Part 236 of 500 done. in 1.21 min
Part 237 of 500 done. in 1.19 min
Part 238 of 500 done. in 1.20 min
Part 239 of 500 done. in 1.18 min
Part 240 of 500 done. in 1.22 min
Part 241 of 500 done. in 1.20 min
Part 242 of 500 done. in 1.21 min
Part 243 of 500 done. in 1.27 min
Part 244 of 500 done. in 1.23 min
Part 245 of 500 done. in 1.19 min
Part 246 of 500 done. in 1.25 min
Part 247 of 500 done. in 1.21 min
Part 248 of 500 done. in 1.21 min
Part 249 of 500 done. in 1.22 min
Part 250 of 500 done. in 1.21 min
Part 251 of 500 done. in 1.20 min
Part 252 of 500 done. in 1.21 min
Part 253 of 500 done. in 1.22 min
Part 254 of 500 done. in 1.17 min
Part 255 of 500 done. in 1.21 min
Part 256 of 500 done. in 1.18 min
Part 257 of 500 done. in 1.16 min
Part 258 of 500 done. in 1.19 min
Part 259 of 500 done. in 1.19 min
Part 260 of 500 done. in 1.22 min
Part 261 of 500 done. in 1.19 min
Part 262 of 500 done. in 1.20 min
Part 263 of 500 done. in 1.20 min
Part 264 of 500 done. in 1.21 min
Part 265 of 500 done. in 1.23 min
Part 266 of 500 done. in 1.21 min
Part 267 of 500 done. in 1.20 min
Part 268 of 500 done. in 1.21 min
Part 269 of 500 done. in 1.23 min
Part 270 of 500 done. in 1.21 min
Part 271 of 500 done. in 1.19 min
Part 272 of 500 done. in 1.20 min
Part 273 of 500 done. in 1.19 min
Part 274 of 500 done. in 1.23 min
```

```
Part 275 of 500 done. in 1.28 min
Part 276 of 500 done. in 1.23 min
Part 277 of 500 done. in 1.25 min
Part 278 of 500 done. in 1.24 min
Part 279 of 500 done. in 1.23 min
Part 280 of 500 done. in 1.20 min
Part 281 of 500 done. in 1.19 min
Part 282 of 500 done. in 1.16 min
Part 283 of 500 done. in 1.20 min
Part 284 of 500 done. in 1.19 min
Part 285 of 500 done. in 1.21 min
Part 286 of 500 done. in 1.18 min
Part 287 of 500 done. in 1.20 min
Part 288 of 500 done. in 1.20 min
Part 289 of 500 done. in 1.15 min
Part 290 of 500 done. in 1.20 min
Part 291 of 500 done. in 1.20 min
Part 292 of 500 done. in 1.20 min
Part 293 of 500 done. in 1.18 min
Part 294 of 500 done. in 1.22 min
Part 295 of 500 done. in 1.22 min
Part 296 of 500 done. in 1.16 min
Part 297 of 500 done. in 1.19 min
Part 298 of 500 done. in 1.21 min
Part 299 of 500 done. in 1.21 min
Part 300 of 500 done. in 1.19 min
Part 301 of 500 done. in 1.14 min
Part 302 of 500 done. in 1.18 min
Part 303 of 500 done. in 1.19 min
Part 304 of 500 done. in 1.21 min
Part 305 of 500 done. in 1.17 min
Part 306 of 500 done. in 1.16 min
Part 307 of 500 done. in 1.20 min
Part 308 of 500 done. in 1.16 min
Part 309 of 500 done. in 1.19 min
Part 310 of 500 done. in 1.15 min
Part 311 of 500 done. in 1.16 min
Part 312 of 500 done. in 1.13 min
Part 313 of 500 done. in 1.17 min
Part 314 of 500 done. in 1.20 min
Part 315 of 500 done. in 1.16 min
Part 316 of 500 done. in 1.17 min
Part 317 of 500 done. in 1.16 min
Part 318 of 500 done. in 1.13 min
Part 319 of 500 done. in 1.20 min
Part 320 of 500 done. in 1.19 min
Part 321 of 500 done. in 1.16 min
Part 322 of 500 done. in 1.15 min
```

```
Part 323 of 500 done. in 1.15 min
Part 324 of 500 done. in 1.17 min
Part 325 of 500 done. in 1.14 min
Part 326 of 500 done. in 1.13 min
Part 327 of 500 done. in 1.20 min
Part 328 of 500 done. in 1.20 min
Part 329 of 500 done. in 1.24 min
Part 330 of 500 done. in 1.20 min
Part 331 of 500 done. in 1.18 min
Part 332 of 500 done. in 1.24 min
Part 333 of 500 done. in 1.22 min
Part 334 of 500 done. in 1.20 min
Part 335 of 500 done. in 1.22 min
Part 336 of 500 done. in 1.25 min
Part 337 of 500 done. in 1.20 min
Part 338 of 500 done. in 1.20 min
Part 339 of 500 done. in 1.19 min
Part 340 of 500 done. in 1.27 min
Part 341 of 500 done. in 1.28 min
Part 342 of 500 done. in 1.25 min
Part 343 of 500 done. in 1.25 min
Part 344 of 500 done. in 1.22 min
Part 345 of 500 done. in 1.22 min
Part 346 of 500 done. in 1.17 min
Part 347 of 500 done. in 1.19 min
Part 348 of 500 done. in 1.18 min
Part 349 of 500 done. in 1.17 min
Part 350 of 500 done. in 1.22 min
Part 351 of 500 done. in 1.19 min
Part 352 of 500 done. in 1.18 min
Part 353 of 500 done. in 1.25 min
Part 354 of 500 done. in 1.26 min
Part 355 of 500 done. in 1.20 min
Part 356 of 500 done. in 1.20 min
Part 357 of 500 done. in 1.22 min
Part 358 of 500 done. in 1.22 min
Part 359 of 500 done. in 1.21 min
Part 360 of 500 done. in 1.23 min
Part 361 of 500 done. in 1.21 min
Part 362 of 500 done. in 1.23 min
Part 363 of 500 done. in 1.20 min
Part 364 of 500 done. in 1.19 min
Part 365 of 500 done. in 1.23 min
Part 366 of 500 done. in 1.25 min
Part 367 of 500 done. in 1.21 min
Part 368 of 500 done. in 1.30 min
Part 369 of 500 done. in 1.21 min
Part 370 of 500 done. in 1.25 min
```

```
Part 371 of 500 done. in 1.20 min
Part 372 of 500 done. in 1.23 min
Part 373 of 500 done. in 1.24 min
Part 374 of 500 done. in 1.20 min
Part 375 of 500 done. in 1.17 min
Part 376 of 500 done. in 1.19 min
Part 377 of 500 done. in 1.25 min
Part 378 of 500 done. in 1.23 min
Part 379 of 500 done. in 1.24 min
Part 380 of 500 done. in 1.20 min
Part 381 of 500 done. in 1.20 min
Part 382 of 500 done. in 1.15 min
Part 383 of 500 done. in 1.19 min
Part 384 of 500 done. in 1.22 min
Part 385 of 500 done. in 1.18 min
Part 386 of 500 done. in 1.13 min
Part 387 of 500 done. in 1.20 min
Part 388 of 500 done. in 1.22 min
Part 389 of 500 done. in 1.23 min
Part 390 of 500 done. in 1.19 min
Part 391 of 500 done. in 1.22 min
Part 392 of 500 done. in 1.21 min
Part 393 of 500 done. in 1.23 min
Part 394 of 500 done. in 1.23 min
Part 395 of 500 done. in 1.23 min
Part 396 of 500 done. in 1.21 min
Part 397 of 500 done. in 1.17 min
Part 398 of 500 done. in 1.18 min
Part 399 of 500 done. in 1.20 min
Part 400 of 500 done. in 1.19 min
Part 401 of 500 done. in 1.17 min
Part 402 of 500 done. in 1.17 min
Part 403 of 500 done. in 1.20 min
Part 404 of 500 done. in 1.17 min
Part 405 of 500 done. in 1.19 min
Part 406 of 500 done. in 1.21 min
Part 407 of 500 done. in 1.19 min
Part 408 of 500 done. in 1.19 min
Part 409 of 500 done. in 1.20 min
Part 410 of 500 done. in 1.19 min
Part 411 of 500 done. in 1.21 min
Part 412 of 500 done. in 1.21 min
Part 413 of 500 done. in 1.26 min
Part 414 of 500 done. in 1.26 min
Part 415 of 500 done. in 1.40 min
Part 416 of 500 done. in 1.27 min
Part 417 of 500 done. in 1.25 min
Part 418 of 500 done. in 1.30 min
```

```
Part 419 of 500 done. in 1.26 min
Part 420 of 500 done. in 1.29 min
Part 421 of 500 done. in 1.28 min
Part 422 of 500 done. in 1.32 min
Part 423 of 500 done. in 1.28 min
Part 424 of 500 done. in 1.30 min
Part 425 of 500 done. in 1.33 min
Part 426 of 500 done. in 1.35 min
Part 427 of 500 done. in 1.27 min
Part 428 of 500 done. in 1.26 min
Part 429 of 500 done. in 1.26 min
Part 430 of 500 done. in 1.25 min
Part 431 of 500 done. in 1.25 min
Part 432 of 500 done. in 1.26 min
Part 433 of 500 done. in 1.27 min
Part 434 of 500 done. in 1.27 min
Part 435 of 500 done. in 1.34 min
Part 436 of 500 done. in 1.35 min
Part 437 of 500 done. in 1.27 min
Part 438 of 500 done. in 1.23 min
Part 439 of 500 done. in 1.28 min
Part 440 of 500 done. in 1.28 min
Part 441 of 500 done. in 1.24 min
Part 442 of 500 done. in 1.26 min
Part 443 of 500 done. in 1.30 min
Part 444 of 500 done. in 1.29 min
Part 445 of 500 done. in 1.24 min
Part 446 of 500 done. in 1.20 min
Part 447 of 500 done. in 1.23 min
Part 448 of 500 done. in 1.26 min
Part 449 of 500 done. in 1.26 min
Part 450 of 500 done. in 1.30 min
Part 451 of 500 done. in 1.27 min
Part 452 of 500 done. in 1.24 min
Part 453 of 500 done. in 1.28 min
Part 454 of 500 done. in 1.26 min
Part 455 of 500 done. in 1.24 min
Part 456 of 500 done. in 1.25 min
Part 457 of 500 done. in 1.29 min
Part 458 of 500 done. in 1.26 min
Part 459 of 500 done. in 1.25 min
Part 460 of 500 done. in 1.23 min
Part 461 of 500 done. in 1.20 min
Part 462 of 500 done. in 1.24 min
Part 463 of 500 done. in 1.29 min
Part 464 of 500 done. in 1.33 min
Part 465 of 500 done. in 1.18 min
Part 466 of 500 done. in 1.24 min
```

```
Part 467 of 500 done. in 1.29 min
Part 468 of 500 done. in 1.30 min
Part 469 of 500 done. in 1.23 min
Part 470 of 500 done. in 1.30 min
Part 471 of 500 done. in 1.27 min
Part 472 of 500 done. in 1.24 min
Part 473 of 500 done. in 1.24 min
Part 474 of 500 done. in 1.19 min
Part 475 of 500 done. in 1.16 min
Part 476 of 500 done. in 1.18 min
Part 477 of 500 done. in 1.28 min
Part 478 of 500 done. in 1.26 min
Part 479 of 500 done. in 1.25 min
Part 480 of 500 done. in 1.24 min
Part 481 of 500 done. in 1.23 min
Part 482 of 500 done. in 1.23 min
Part 483 of 500 done. in 1.23 min
Part 484 of 500 done. in 1.24 min
Part 485 of 500 done. in 1.25 min
Part 486 of 500 done. in 1.29 min
Part 487 of 500 done. in 1.28 min
Part 488 of 500 done. in 1.26 min
Part 489 of 500 done. in 1.26 min
Part 490 of 500 done. in 1.22 min
Part 491 of 500 done. in 1.20 min
Part 492 of 500 done. in 1.23 min
Part 493 of 500 done. in 1.22 min
Part 494 of 500 done. in 1.23 min
Part 495 of 500 done. in 1.25 min
Part 496 of 500 done. in 1.23 min
Part 497 of 500 done. in 1.23 min
Part 498 of 500 done. in 1.18 min
Part 499 of 500 done. in 1.18 min
Part 500 of 500 done. in 1.20 min
```

[ ]:

```
num = len(os.listdir('Data//embeddingRo//'))

with open('Data//embeddingRo//embeddings'+str(0),'rb') as f:
    dataD = pickle.load(f)

for idx in range(1,num):

  with open('Data//embeddingRo//embeddings'+str(idx),'rb') as f:
    mat = pickle.load(f)
    dataD=np.concatenate([dataD,mat],axis=0)
```

[ ]:

```
np.shape(dataD)
```

[ ]: (400000, 768)

[ ]:
```
df = df.iloc[:400000,:]
```

[ ]:
```
train_text, temp_text, train_labels, temp_labels = train_test_split(dataD,␣
 ↪df['Label'],
                                                                    ␣
 ↪random_state=2018,
                                                                    test_size=0.
 ↪3,
                                                                    ␣
 ↪stratify=df['Label'])

# we will use temp_text and temp_labels to create validation and test set
val_text, test_text, val_labels, test_labels = train_test_split(temp_text,␣
 ↪temp_labels,
                                                                ␣
 ↪random_state=2018,
                                                                test_size=0.5,
                                                                ␣
 ↪stratify=temp_labels)
```

[ ]:
```
train_labels = tf.keras.utils.to_categorical(train_labels)
val_labels = tf.keras.utils.to_categorical(val_labels)
test_labels = tf.keras.utils.to_categorical(test_labels)
```

[ ]:
```
train_data = tf.data.Dataset.from_tensor_slices((train_text, train_labels))
train_data = train_data.batch(128)


val_data = tf.data.Dataset.from_tensor_slices((val_text, val_labels))
val_data = val_data.batch(128)
```

[ ]:
```
input = tf.keras.layers.Input(shape=(768,), name='input_token', dtype='int32')
X = tf.keras.layers.Dense(768, activation='relu')(input)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.Dense(512, activation='relu')(input)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.Dense(128, activation='relu')(X)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.Dense(2, activation='softmax')(X)
model = tf.keras.Model(inputs=input, outputs = X)
```

[ ]:
```
model.summary()
```

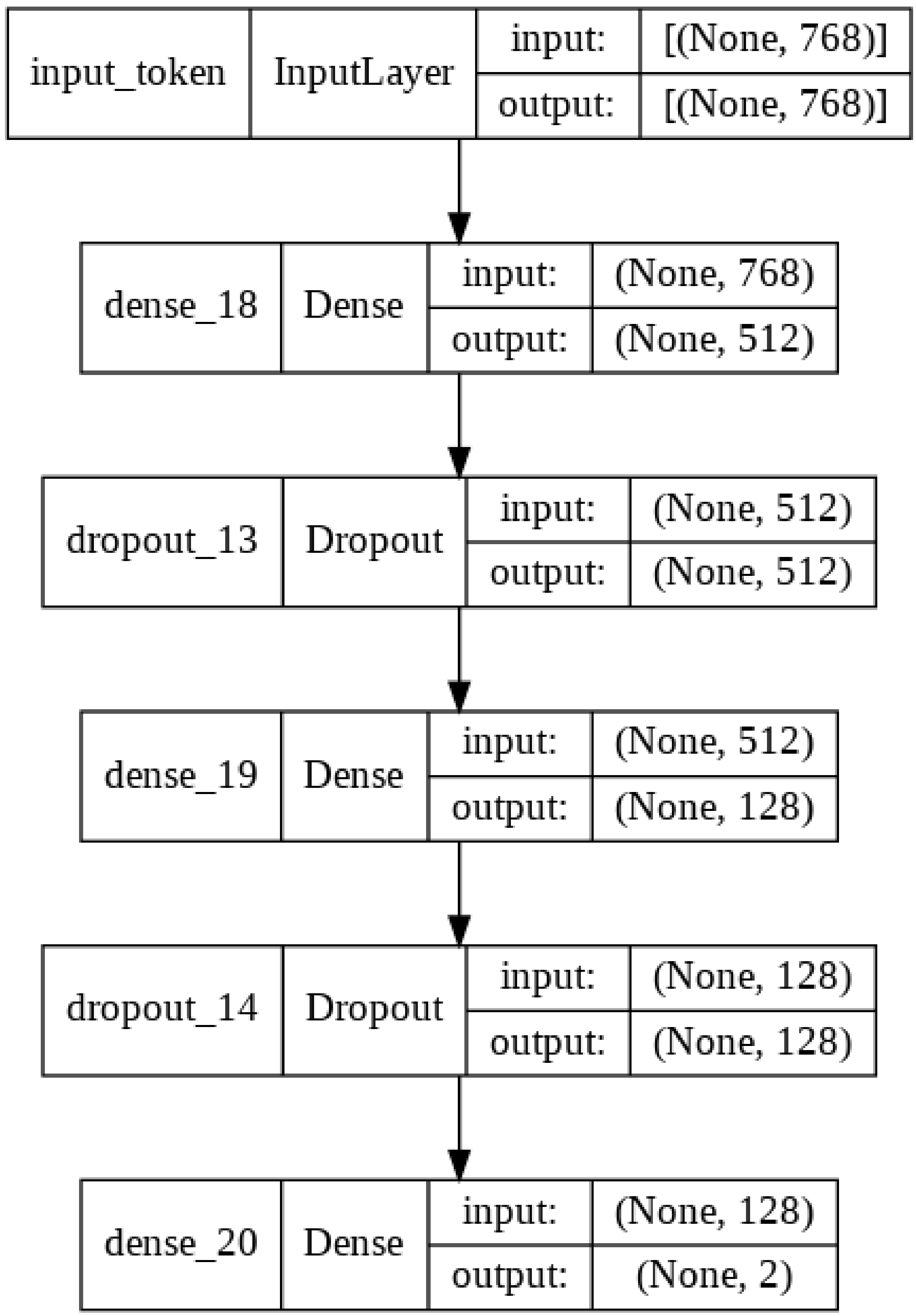
```
Model: "model_5"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_token (InputLayer)    [(None, 768)]             0
```

```
 dense_18 (Dense)            (None, 512)               393728

 dropout_13 (Dropout)        (None, 512)               0

 dense_19 (Dense)            (None, 128)               65664

 dropout_14 (Dropout)        (None, 128)               0

 dense_20 (Dense)            (None, 2)                 258

=================================================================
Total params: 459,650
Trainable params: 459,650
Non-trainable params: 0
_________________________________________________________________
```

[ ]:

```python
from keras.utils.vis_utils import plot_model
plot_model(model, show_shapes=True, show_layer_names=True)
```

[ ]:


input_token
InputLayer
input:
[(None, 768)]
output:
[(None, 768)]
dense_18
Dense
input:
(None, 768)
output:
(None, 512)
dropout_13
Dropout
input:
(None, 512)
output:
(None, 512)
dense_19
Dense
input:
(None, 512)
output:
(None, 128)
dropout_14
Dropout
input:
(None, 128)
output:
(None, 128)
dense_20
Dense
input:
(None, 128)
output:
(None, 2)

[ ]:
```
filepath="roBERT.hdf5"
checkpoint = ModelCheckpoint(filepath,␣
 ↪monitor='val_loss',verbose=1,save_best_only=True, mode='min')
ES =tf.keras.callbacks.
 ↪EarlyStopping(monitor="val_loss",patience=patience,verbose=1,mode="min",restore_best_weight
# pre = tf.keras.metrics.Precision()
f1 = tfa.metrics.F1Score(num_classes=2, average="macro")
callbacks_list = [checkpoint,ES]
model.compile(loss='binary_crossentropy', optimizer='adam' ,metrics=[f1])
```

[ ]:
```
history = model.fit(train_data, validation_data=val_data,␣
 ↪epochs=epoch,verbose=1, callbacks = callbacks_list)
```

```
Epoch 1/30
2185/2188 [============================>.] - ETA: 0s - loss: 0.6529 - f1_score:
0.4582
Epoch 00001: val_loss improved from inf to 0.65063, saving model to roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6529 -
f1_score: 0.4582 - val_loss: 0.6506 - val_f1_score: 0.4436
Epoch 2/30
2180/2188 [============================>.] - ETA: 0s - loss: 0.6506 - f1_score:
0.4540
Epoch 00002: val_loss improved from 0.65063 to 0.64889, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6506 -
f1_score: 0.4540 - val_loss: 0.6489 - val_f1_score: 0.4474
Epoch 3/30
2187/2188 [============================>.] - ETA: 0s - loss: 0.6502 - f1_score:
0.4530
Epoch 00003: val_loss improved from 0.64889 to 0.64838, saving model to
roBERT.hdf5
2188/2188 [==============================] - 9s 4ms/step - loss: 0.6502 -
f1_score: 0.4530 - val_loss: 0.6484 - val_f1_score: 0.4510
Epoch 4/30
2186/2188 [============================>.] - ETA: 0s - loss: 0.6501 - f1_score:
0.4527
Epoch 00004: val_loss improved from 0.64838 to 0.64780, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6501 -
f1_score: 0.4527 - val_loss: 0.6478 - val_f1_score: 0.4539
Epoch 5/30
2173/2188 [============================>.] - ETA: 0s - loss: 0.6500 - f1_score:
0.4546
Epoch 00005: val_loss improved from 0.64780 to 0.64767, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6499 -
f1_score: 0.4549 - val_loss: 0.6477 - val_f1_score: 0.4487
Epoch 6/30
```

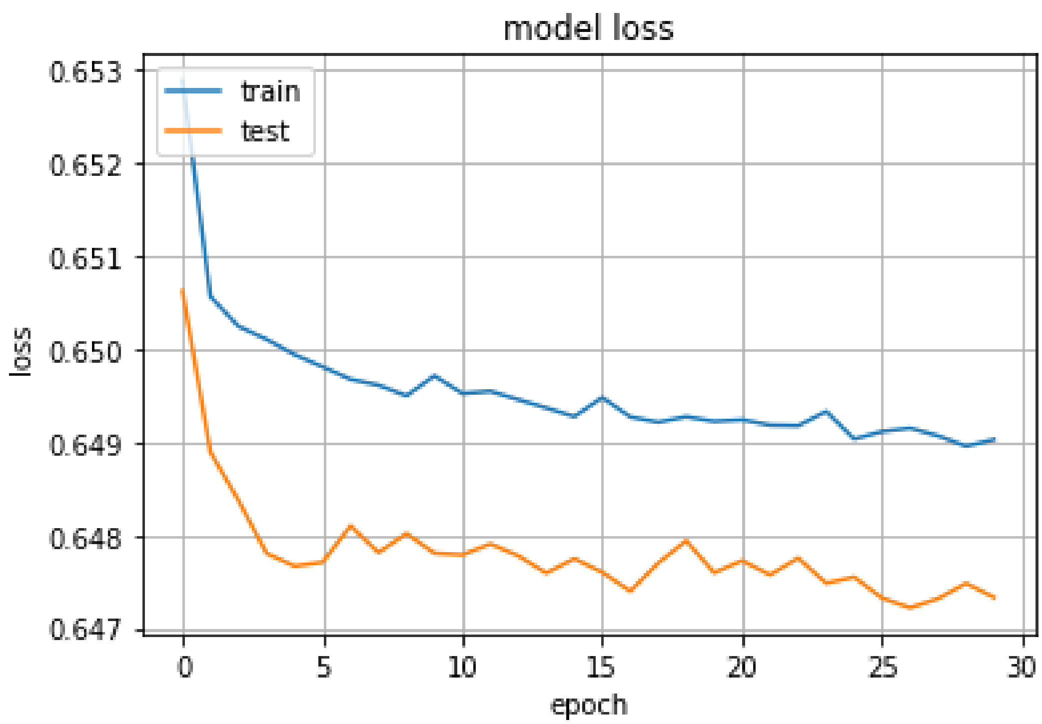
```
2178/2188 [============================>.] - ETA: 0s - loss: 0.6498 - f1_score:
0.4555
Epoch 00006: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6498 -
f1_score: 0.4556 - val_loss: 0.6477 - val_f1_score: 0.4505
Epoch 7/30
2176/2188 [============================>.] - ETA: 0s - loss: 0.6497 - f1_score:
0.4548
Epoch 00007: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6497 -
f1_score: 0.4551 - val_loss: 0.6481 - val_f1_score: 0.4557
Epoch 8/30
2179/2188 [============================>.] - ETA: 0s - loss: 0.6496 - f1_score:
0.4562
Epoch 00008: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6496 -
f1_score: 0.4562 - val_loss: 0.6478 - val_f1_score: 0.4495
Epoch 9/30
2172/2188 [============================>.] - ETA: 0s - loss: 0.6495 - f1_score:
0.4551
Epoch 00009: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6495 -
f1_score: 0.4552 - val_loss: 0.6480 - val_f1_score: 0.4476
Epoch 10/30
2174/2188 [============================>.] - ETA: 0s - loss: 0.6498 - f1_score:
0.4550
Epoch 00010: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6497 -
f1_score: 0.4551 - val_loss: 0.6478 - val_f1_score: 0.4493
Epoch 11/30
2178/2188 [============================>.] - ETA: 0s - loss: 0.6496 - f1_score:
0.4531
Epoch 00011: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6495 -
f1_score: 0.4532 - val_loss: 0.6478 - val_f1_score: 0.4504
Epoch 12/30
2178/2188 [============================>.] - ETA: 0s - loss: 0.6496 - f1_score:
0.4551
Epoch 00012: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6495 -
f1_score: 0.4552 - val_loss: 0.6479 - val_f1_score: 0.4495
Epoch 13/30
2177/2188 [============================>.] - ETA: 0s - loss: 0.6495 - f1_score:
0.4555
Epoch 00013: val_loss did not improve from 0.64767
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6495 -
f1_score: 0.4558 - val_loss: 0.6478 - val_f1_score: 0.4529
Epoch 14/30
```

```
2171/2188 [============================>.] - ETA: 0s - loss: 0.6494 - f1_score:
0.4551
Epoch 00014: val_loss improved from 0.64767 to 0.64759, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6494 -
f1_score: 0.4552 - val_loss: 0.6476 - val_f1_score: 0.4509
Epoch 15/30
2181/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4555
Epoch 00015: val_loss did not improve from 0.64759
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6493 -
f1_score: 0.4555 - val_loss: 0.6477 - val_f1_score: 0.4512
Epoch 16/30
2181/2188 [============================>.] - ETA: 0s - loss: 0.6495 - f1_score:
0.4553
Epoch 00016: val_loss did not improve from 0.64759
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6495 -
f1_score: 0.4554 - val_loss: 0.6476 - val_f1_score: 0.4508
Epoch 17/30
2177/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4572
Epoch 00017: val_loss improved from 0.64759 to 0.64740, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6493 -
f1_score: 0.4572 - val_loss: 0.6474 - val_f1_score: 0.4475
Epoch 18/30
2184/2188 [============================>.] - ETA: 0s - loss: 0.6492 - f1_score:
0.4561
Epoch 00018: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6492 -
f1_score: 0.4561 - val_loss: 0.6477 - val_f1_score: 0.4513
Epoch 19/30
2173/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4555
Epoch 00019: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6493 -
f1_score: 0.4556 - val_loss: 0.6479 - val_f1_score: 0.4494
Epoch 20/30
2172/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4554
Epoch 00020: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6492 -
f1_score: 0.4555 - val_loss: 0.6476 - val_f1_score: 0.4496
Epoch 21/30
2179/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4557
Epoch 00021: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6492 -
```

```
f1_score: 0.4558 - val_loss: 0.6477 - val_f1_score: 0.4491
Epoch 22/30
2173/2188 [============================>.] - ETA: 0s - loss: 0.6493 - f1_score:
0.4554
Epoch 00022: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6492 -
f1_score: 0.4555 - val_loss: 0.6476 - val_f1_score: 0.4487
Epoch 23/30
2187/2188 [============================>.] - ETA: 0s - loss: 0.6492 - f1_score:
0.4546
Epoch 00023: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6492 -
f1_score: 0.4546 - val_loss: 0.6478 - val_f1_score: 0.4500
Epoch 24/30
2183/2188 [============================>.] - ETA: 0s - loss: 0.6494 - f1_score:
0.4537
Epoch 00024: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6493 -
f1_score: 0.4537 - val_loss: 0.6475 - val_f1_score: 0.4483
Epoch 25/30
2185/2188 [============================>.] - ETA: 0s - loss: 0.6490 - f1_score:
0.4568
Epoch 00025: val_loss did not improve from 0.64740
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6490 -
f1_score: 0.4568 - val_loss: 0.6475 - val_f1_score: 0.4491
Epoch 26/30
2182/2188 [============================>.] - ETA: 0s - loss: 0.6491 - f1_score:
0.4577
Epoch 00026: val_loss improved from 0.64740 to 0.64732, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6491 -
f1_score: 0.4577 - val_loss: 0.6473 - val_f1_score: 0.4492
Epoch 27/30
2183/2188 [============================>.] - ETA: 0s - loss: 0.6492 - f1_score:
0.4546
Epoch 00027: val_loss improved from 0.64732 to 0.64722, saving model to
roBERT.hdf5
2188/2188 [==============================] - 8s 4ms/step - loss: 0.6492 -
f1_score: 0.4547 - val_loss: 0.6472 - val_f1_score: 0.4493
Epoch 28/30
2179/2188 [============================>.] - ETA: 0s - loss: 0.6491 - f1_score:
0.4542
Epoch 00028: val_loss did not improve from 0.64722
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6491 -
f1_score: 0.4542 - val_loss: 0.6473 - val_f1_score: 0.4467
Epoch 29/30
2180/2188 [============================>.] - ETA: 0s - loss: 0.6490 - f1_score:
0.4547
```

```
Epoch 00029: val_loss did not improve from 0.64722
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6490 -
f1_score: 0.4548 - val_loss: 0.6475 - val_f1_score: 0.4517
Epoch 30/30
2183/2188 [============================>.] - ETA: 0s - loss: 0.6490 - f1_score:
0.4552
Epoch 00030: val_loss did not improve from 0.64722
2188/2188 [==============================] - 8s 3ms/step - loss: 0.6490 -
f1_score: 0.4552 - val_loss: 0.6473 - val_f1_score: 0.4470
```

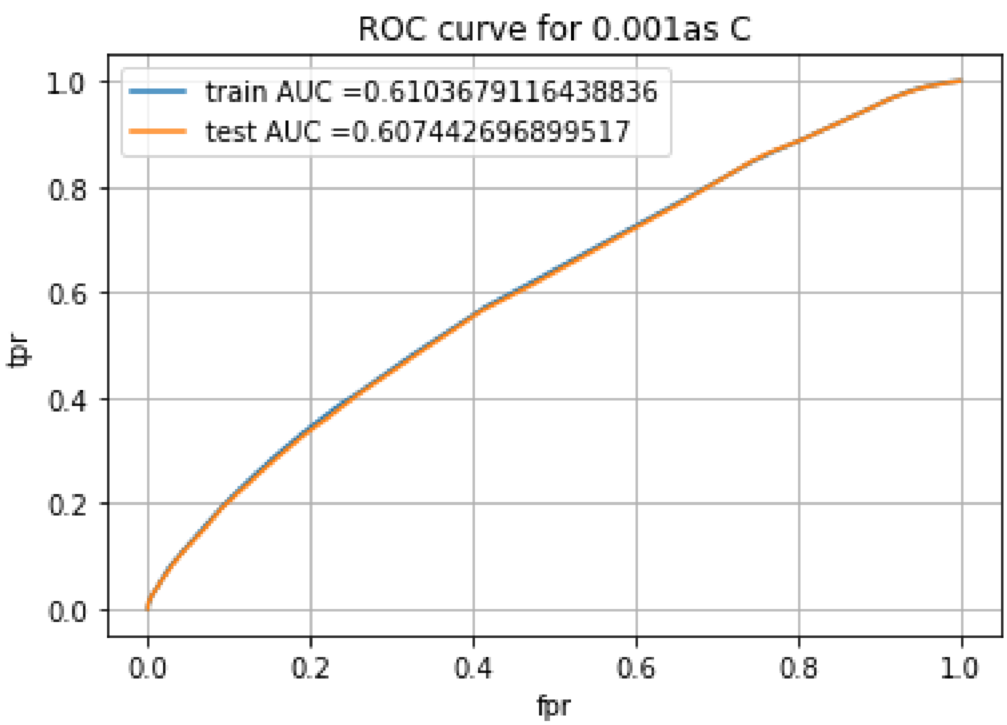
[ ]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.grid()
plt.show()
```



[ ]:
```
from keras.models import load_model
model = load_model("roBERT.hdf5")
```

[ ]:
```
test_data = tf.data.Dataset.from_tensor_slices((test_text))
test_data = test_data.shuffle(5000).batch(128)
```

```
[ ]: y_pr_ts = model.predict(test_data)[:,0]
y_pred_tr = model.predict(train_data)[:,0]
y_ts = test_labels[:,0]
y_tr = train_labels[:,0]
from sklearn.metrics import␣
 ↪roc_curve,auc,confusion_matrix,accuracy_score,precision_score,recall_score,f1_score

train_fpr, train_tpr, tr_thresholds = roc_curve(y_tr, y_pred_tr)
test_fpr, test_tpr, te_thresholds = roc_curve(y_ts, y_pr_ts)
plt.plot(train_fpr, train_tpr, label="train AUC␣
 ↪="+str(auc(train_fpr,train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.xlabel("fpr")
plt.ylabel("tpr")
plt.title('ROC curve for '+str (0.001)+'as C')
plt.legend()
plt.grid()
plt.show()
```




```
[ ]: # This section of code where ever implemented is taken from sample kNN python␣
 ↪notebook

def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
```

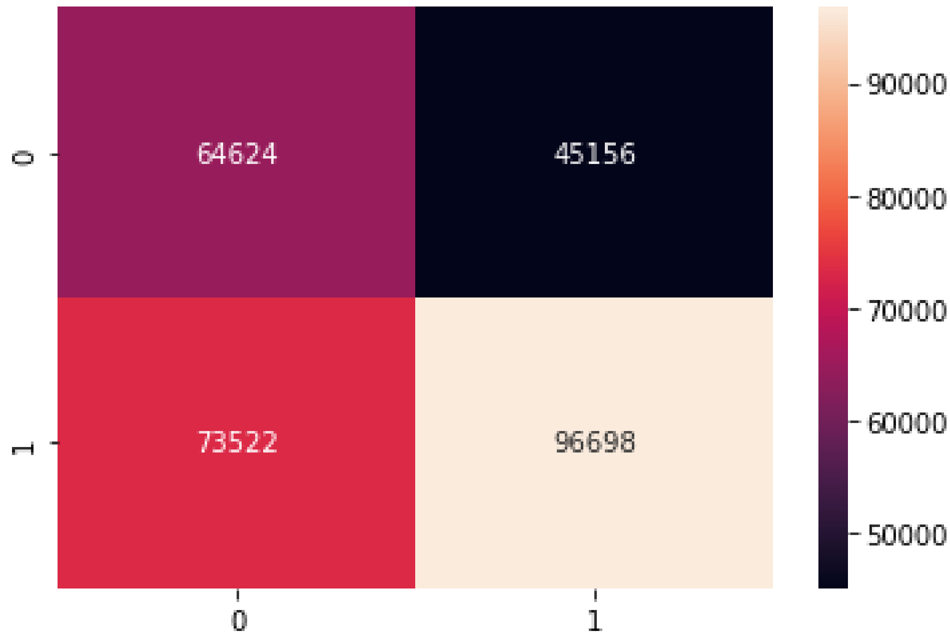
```
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very␣
↪high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for␣
↪threshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions


print('test')
best_ts_thres = find_best_threshold(te_thresholds, test_fpr, test_tpr)

print('train')
best_tr_thres = find_best_threshold(tr_thresholds, train_fpr, train_tpr)
```

```
test
the maximum value of tpr*(1-fpr) 0.33247305148634676 for threshold 0.663
train
the maximum value of tpr*(1-fpr) 0.3344086594483335 for threshold 0.667
```

[ ]:
```
print('Train Confusion Matrix')
cm2 = pd.DataFrame(confusion_matrix(y_tr, predict_with_best_t(y_pred_tr,␣
↪best_tr_thres)), range(2),range(2))
sns.heatmap(cm2, annot=True,fmt='g')
```

```
Train Confusion Matrix
```

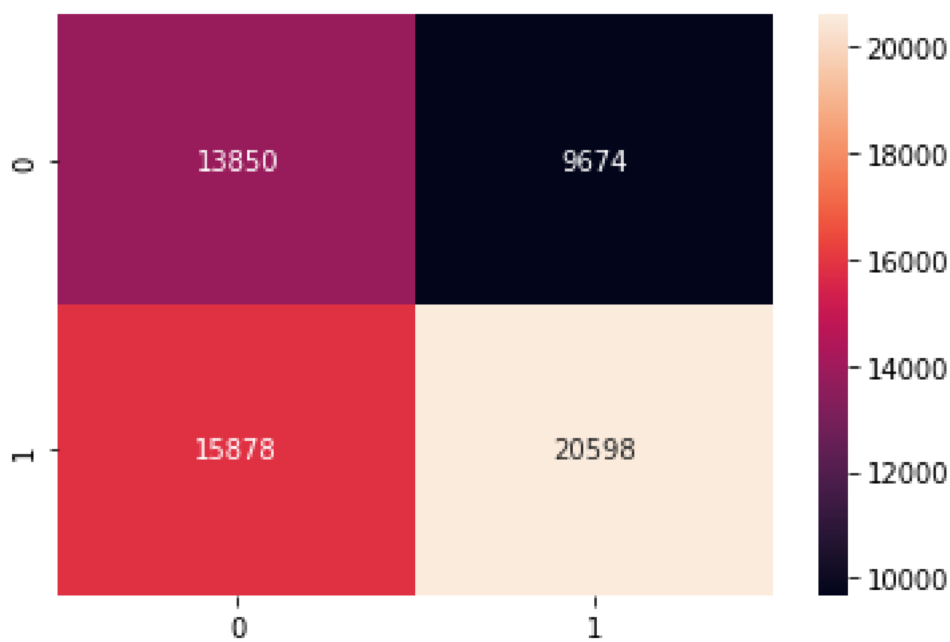
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f694651d8d0>
```



[ ]:
```
print('Test Confusion Matrix')
cm2 = pd.DataFrame(confusion_matrix(y_ts, predict_with_best_t(y_pr_ts,␣
 ↪best_ts_thres)), range(2),range(2))
sns.heatmap(cm2, annot=True,fmt='g')
```

```
Test Confusion Matrix
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f69463d2410>
```



[ ]:
```
acc=accuracy_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
ps=precision_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
rc=recall_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
f1=f1_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100

print("Accuracy on test set: %0.2f%%"%(acc))
print("Precision on test set: %0.2f%%"%(ps))
print("recall score on test set: %0.2f%%"%(rc))
print("f1 score on test set: %0.2f%%"%(f1))
```

```
Accuracy on test set: 57.41%
Precision on test set: 68.04%
recall score on test set: 56.47%
f1 score on test set: 61.72%
```

[ ]:

```
acc=accuracy_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
ps=precision_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
rc=recall_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
f1=f1_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100

print("Accuracy on test set: %0.2f%%"%(acc))
print("Precision on test set: %0.2f%%"%(ps))
print("recall score on test set: %0.2f%%"%(rc))
print("f1 score on test set: %0.2f%%"%(f1))
```

```
Accuracy on test set: 57.41%
Precision on test set: 68.04%
recall score on test set: 56.47%
f1 score on test set: 61.72%
```


[ ]: