

5.SciBert400k

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 462405
-rw----- 1 root root    207564 Dec 25 10:00 '2 SciBert400k.ipynb'
-rw----- 1 root root    208923 Dec 24 22:28 '3 SciBert400k.ipynb'
-rw----- 1 root root   233350 Dec 24 18:11 '4 SciBert400k.ipynb'
drwx----- 2 root root     4096 Dec  3 16:27 clr
-rw----- 1 root root   349041 Dec 24 18:32 Compare.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 18 07:30 Distllbert400000.ipynb
-rw----- 1 root root     97583 Dec 24 22:28 'EDA on results.ipynb'
drwx----- 2 root root     4096 Dec 18 07:14 'misc model'
-rw----- 1 root root    43839 Dec 24 17:19 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    86347 Dec 19 06:43 Roberta.ipynb
-rw----- 1 root root    208993 Dec 25 10:25 SciBert400k.ipynb
drwx----- 2 root root     4096 Dec 24 17:39 scibert_scivocab_uncased
-rw----- 1 root root 442460160 Oct 24 2019 scibert_scivocab_uncased.tar.2
```

```
[ ]: 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)

```

Sat Dec 25 10:26:01 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   41C    P0      29W / 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 9.4 MB/s

Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (1.19.5)

Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-

```

packages (from transformers) (3.4.0)
Collecting sacremoses
  Downloading sacremoses-0.0.46-py3-none-any.whl (895 kB)
    || 895 kB 27.9 MB/s
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers) (4.8.2)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.2.1-py3-none-any.whl (61 kB)
    || 61 kB 427 kB/s
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 37.2 MB/s
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 28.5 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7
/dist-packages (from transformers) (4.62.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers) (2.23.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7
/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-
packages (from transformers) (21.3)
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: certifi>=2017.4.17 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (2021.10.8)
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: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
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)
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 7.0 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 7.7 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 BertModel, BertTokenizer, 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

```
[ ]: # ! wget "https://s3-us-west-2.amazonaws.com/ai2-s2-research/scibert/
→huggingface_pytorch/scibert_scivocab_uncased.tar"
# ! tar -xvf scibert_scivocab_uncased.tar
```

```
[ ]: BERT = BertModel.from_pretrained("./scibert_scivocab_uncased")

tokenizer = BertTokenizer.from_pretrained("./scibert_scivocab_uncased",
                                         do_lower_case=True,
                                         use_fast=True,
                                         max_length=128,
                                         truncation=True,
                                         pad_to_max_length=True)

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

Some weights of the model checkpoint at ./scibert_scivocab_uncased were not used when initializing BertModel: ['cls.predictions.decoder.weight', 'cls.seq_relationship.weight', 'cls.seq_relationship.bias', 'cls.predictions.bias', 'cls.predictions.transform.dense.bias', 'cls.predictions.transform.LayerNorm.bias', 'cls.predictions.transform.dense.weight', 'cls.predictions.transform.LayerNorm.weight']

- This IS expected if you are initializing BertModel 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 BertModel from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).

The tokenizer class you load from this checkpoint is not the same type as the class this function is called from. It may result in unexpected tokenization. The tokenizer class you load from this checkpoint is 'RobertaTokenizer'. The class this function is called from is 'BertTokenizer'.

```

[: batch=20
df = df.iloc[160000:200000,:]
step = int(df.shape[0]/batch)
step

[: 2000

[: ##### 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 sentence
# 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 = 80
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//embeddingSb//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 20 done in 3.68 min
Part 2 of 20 done in 4.25 min
Part 3 of 20 done in 3.50 min
Part 4 of 20 done in 3.28 min
Part 5 of 20 done in 3.38 min

Part 6 of 20 done in 3.30 min
Part 7 of 20 done in 3.24 min
Part 8 of 20 done in 3.25 min
Part 9 of 20 done in 2.98 min
Part 10 of 20 done in 2.96 min
Part 11 of 20 done in 2.96 min
Part 12 of 20 done in 3.00 min
Part 13 of 20 done in 2.99 min
Part 14 of 20 done in 2.93 min
Part 15 of 20 done in 2.96 min
Part 16 of 20 done in 3.07 min
Part 17 of 20 done in 3.42 min
Part 18 of 20 done in 3.29 min
Part 19 of 20 done in 3.27 min
Part 20 of 20 done in 3.23 min

```
[ ]: num = len(os.listdir('Data//embeddingSb//'))

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

for idx in range(1,num):

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

[ ]: np.shape(dataS)

[ ]: (400000, 768)

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

```
[ ]: train_text, temp_text, train_labels, temp_labels = train_test_split(dataS,
    ↳datay['Label'],
    ↳random_state=2018,
    ↳3,
    ↳stratify=datay['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.shuffle(5000).batch(128)
```

```
val_data = tf.data.Dataset.from_tensor_slices((val_text, val_labels))
val_data = val_data.shuffle(5000).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.BatchNormalization()(X)
X = tf.keras.layers.Dense(512, activation='relu')(X)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.BatchNormalization()(X)
X = tf.keras.layers.Dense(128, activation='relu')(X)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.BatchNormalization()(X)
X = tf.keras.layers.Dense(2, activation='softmax')(X)
model = tf.keras.Model(inputs=input, outputs = X)
```

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

Model: "model"

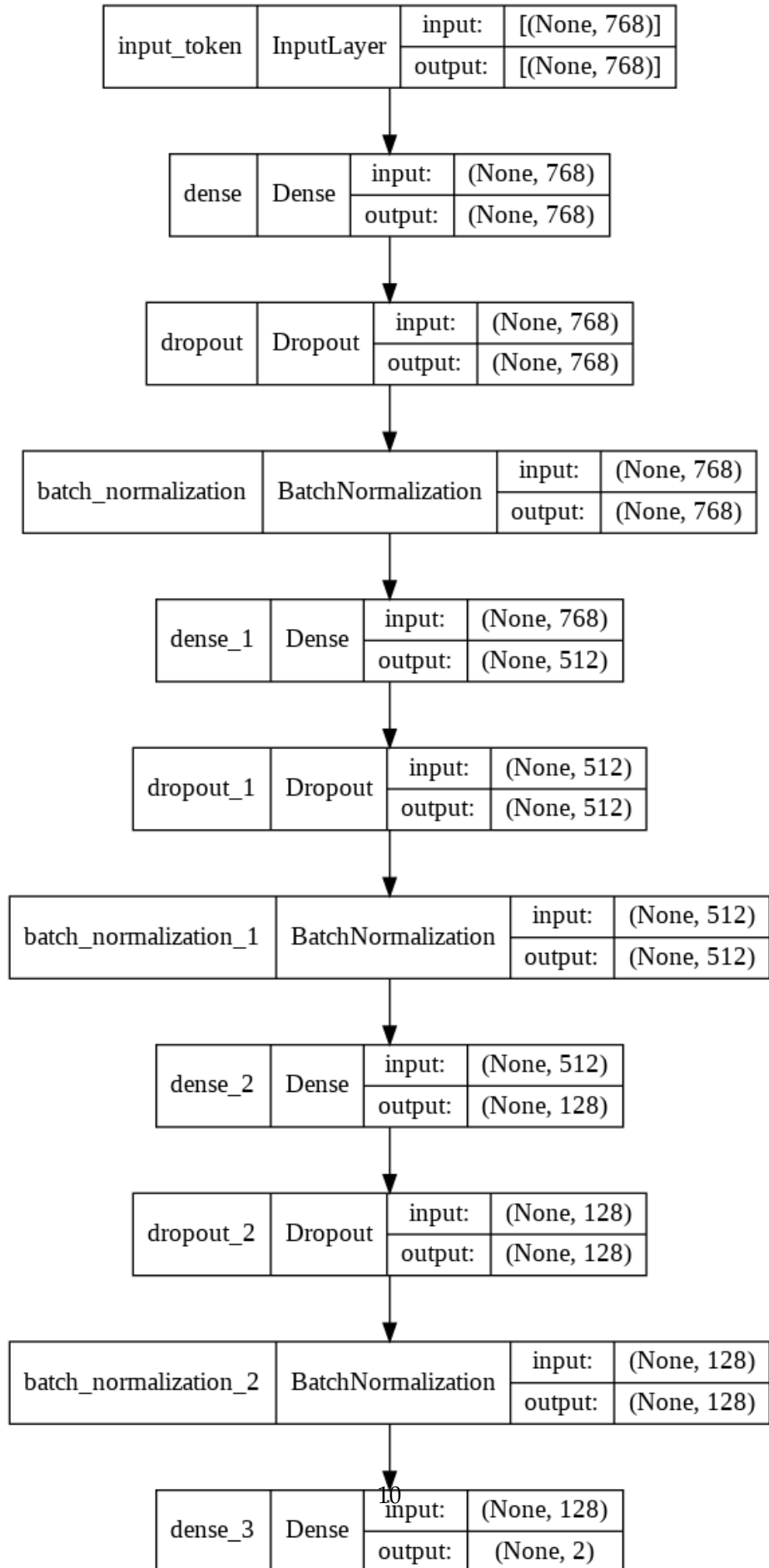
Layer (type)	Output Shape	Param #
input_token (InputLayer)	[(None, 768)]	0
dense (Dense)	(None, 768)	590592
dropout (Dropout)	(None, 768)	0
batch_normalization (Batch Normalization)	(None, 768)	3072
dense_1 (Dense)	(None, 512)	393728
dropout_1 (Dropout)	(None, 512)	0
batch_normalization_1 (Batch Normalization)	(None, 512)	2048

dense_2 (Dense)	(None, 128)	65664
dropout_2 (Dropout)	(None, 128)	0
batch_normalization_2 (Batch Normalization)	(None, 128)	512
dense_3 (Dense)	(None, 2)	258

```
=====
Total params: 1,055,874
Trainable params: 1,053,058
Non-trainable params: 2,816
-----
```

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

```
[ ]:
```



```
[ ]: filepath="SBert.hdf5"
checkpoint = ModelCheckpoint(filepath,
    ↳monitor='val_loss',verbose=1,save_best_only=True, mode='min')
ES =tf.keras.callbacks.
    ↳EarlyStopping(monitor="val_loss",patience=10,verbose=1,mode="min",restore_best_weights=True)
# 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=30,verbose=1,
    ↳callbacks = callbacks_list)
```

```
Epoch 1/30
8740/8750 [=====>.] - ETA: 0s - loss: 0.6416 - f1_score:
0.5651
Epoch 00001: val_loss improved from inf to 0.62894, saving model to SBert.hdf5
8750/8750 [=====] - 47s 5ms/step - loss: 0.6416 -
f1_score: 0.5651 - val_loss: 0.6289 - val_f1_score: 0.5406
Epoch 2/30
8745/8750 [=====>.] - ETA: 0s - loss: 0.6324 - f1_score:
0.5687
Epoch 00002: val_loss improved from 0.62894 to 0.62660, saving model to
SBert.hdf5
8750/8750 [=====] - 41s 5ms/step - loss: 0.6324 -
f1_score: 0.5688 - val_loss: 0.6266 - val_f1_score: 0.5607
Epoch 3/30
8747/8750 [=====>.] - ETA: 0s - loss: 0.6301 - f1_score:
0.5732
Epoch 00003: val_loss improved from 0.62660 to 0.62623, saving model to
SBert.hdf5
8750/8750 [=====] - 41s 5ms/step - loss: 0.6301 -
f1_score: 0.5732 - val_loss: 0.6262 - val_f1_score: 0.5756
Epoch 4/30
8740/8750 [=====>.] - ETA: 0s - loss: 0.6283 - f1_score:
0.5753
Epoch 00004: val_loss did not improve from 0.62623
8750/8750 [=====] - 41s 5ms/step - loss: 0.6283 -
f1_score: 0.5754 - val_loss: 0.6264 - val_f1_score: 0.5867
Epoch 5/30
8741/8750 [=====>.] - ETA: 0s - loss: 0.6269 - f1_score:
0.5766
Epoch 00005: val_loss did not improve from 0.62623
8750/8750 [=====] - 40s 5ms/step - loss: 0.6269 -
f1_score: 0.5766 - val_loss: 0.6266 - val_f1_score: 0.5632
Epoch 6/30
```

```

8745/8750 [=====>.] - ETA: 0s - loss: 0.6255 - f1_score:
0.5773
Epoch 00006: val_loss improved from 0.62623 to 0.62601, saving model to
SBert.hdf5
8750/8750 [=====] - 41s 5ms/step - loss: 0.6255 -
f1_score: 0.5773 - val_loss: 0.6260 - val_f1_score: 0.5364
Epoch 7/30
8746/8750 [=====>.] - ETA: 0s - loss: 0.6244 - f1_score:
0.5821
Epoch 00007: val_loss did not improve from 0.62601
8750/8750 [=====] - 41s 5ms/step - loss: 0.6244 -
f1_score: 0.5821 - val_loss: 0.6284 - val_f1_score: 0.5745
Epoch 8/30
8739/8750 [=====>.] - ETA: 0s - loss: 0.6227 - f1_score:
0.5838
Epoch 00008: val_loss did not improve from 0.62601
8750/8750 [=====] - 40s 5ms/step - loss: 0.6227 -
f1_score: 0.5838 - val_loss: 0.6316 - val_f1_score: 0.5546
Epoch 9/30
8739/8750 [=====>.] - ETA: 0s - loss: 0.6216 - f1_score:
0.5861
Epoch 00009: val_loss did not improve from 0.62601
8750/8750 [=====] - 41s 5ms/step - loss: 0.6216 -
f1_score: 0.5861 - val_loss: 0.6269 - val_f1_score: 0.5866
Epoch 10/30
8740/8750 [=====>.] - ETA: 0s - loss: 0.6202 - f1_score:
0.5895
Epoch 00010: val_loss did not improve from 0.62601
8750/8750 [=====] - 40s 5ms/step - loss: 0.6202 -
f1_score: 0.5895 - val_loss: 0.6343 - val_f1_score: 0.5757
Epoch 11/30
8742/8750 [=====>.] - ETA: 0s - loss: 0.6188 - f1_score:
0.5903
Epoch 00011: val_loss did not improve from 0.62601
8750/8750 [=====] - 40s 5ms/step - loss: 0.6188 -
f1_score: 0.5903 - val_loss: 0.6325 - val_f1_score: 0.5571
Epoch 12/30
8741/8750 [=====>.] - ETA: 0s - loss: 0.6173 - f1_score:
0.5931
Epoch 00012: val_loss did not improve from 0.62601
8750/8750 [=====] - 41s 5ms/step - loss: 0.6172 -
f1_score: 0.5931 - val_loss: 0.6424 - val_f1_score: 0.5626
Epoch 13/30
8746/8750 [=====>.] - ETA: 0s - loss: 0.6155 - f1_score:
0.5963
Epoch 00013: val_loss did not improve from 0.62601
8750/8750 [=====] - 41s 5ms/step - loss: 0.6155 -
f1_score: 0.5962 - val_loss: 0.6361 - val_f1_score: 0.5658

```

```

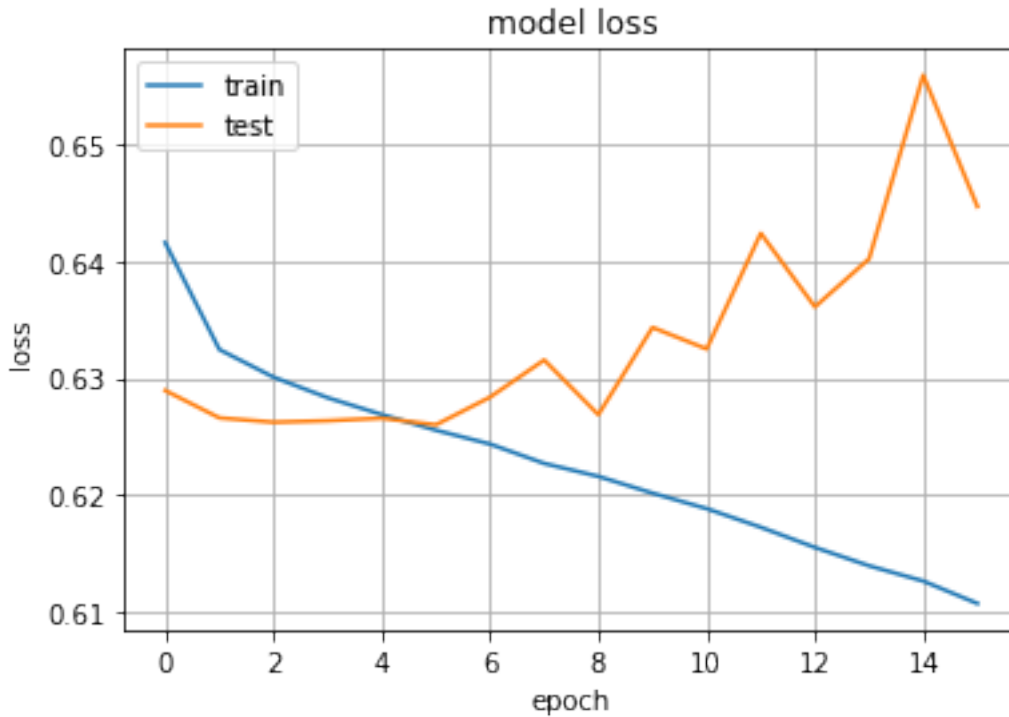
Epoch 14/30
8749/8750 [=====>.] - ETA: 0s - loss: 0.6140 - f1_score:
0.5985
Epoch 00014: val_loss did not improve from 0.62601
8750/8750 [=====] - 41s 5ms/step - loss: 0.6139 -
f1_score: 0.5985 - val_loss: 0.6402 - val_f1_score: 0.5749
Epoch 15/30
8747/8750 [=====>.] - ETA: 0s - loss: 0.6127 - f1_score:
0.6011
Epoch 00015: val_loss did not improve from 0.62601
8750/8750 [=====] - 40s 5ms/step - loss: 0.6126 -
f1_score: 0.6011 - val_loss: 0.6559 - val_f1_score: 0.5789
Epoch 16/30
8750/8750 [=====] - ETA: 0s - loss: 0.6107 - f1_score:
0.6029
Epoch 00016: val_loss did not improve from 0.62601
Restoring model weights from the end of the best epoch: 6.
8750/8750 [=====] - 41s 5ms/step - loss: 0.6107 -
f1_score: 0.6029 - val_loss: 0.6447 - val_f1_score: 0.5755
Epoch 00016: early stopping

```

```

[ ]: 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()

```

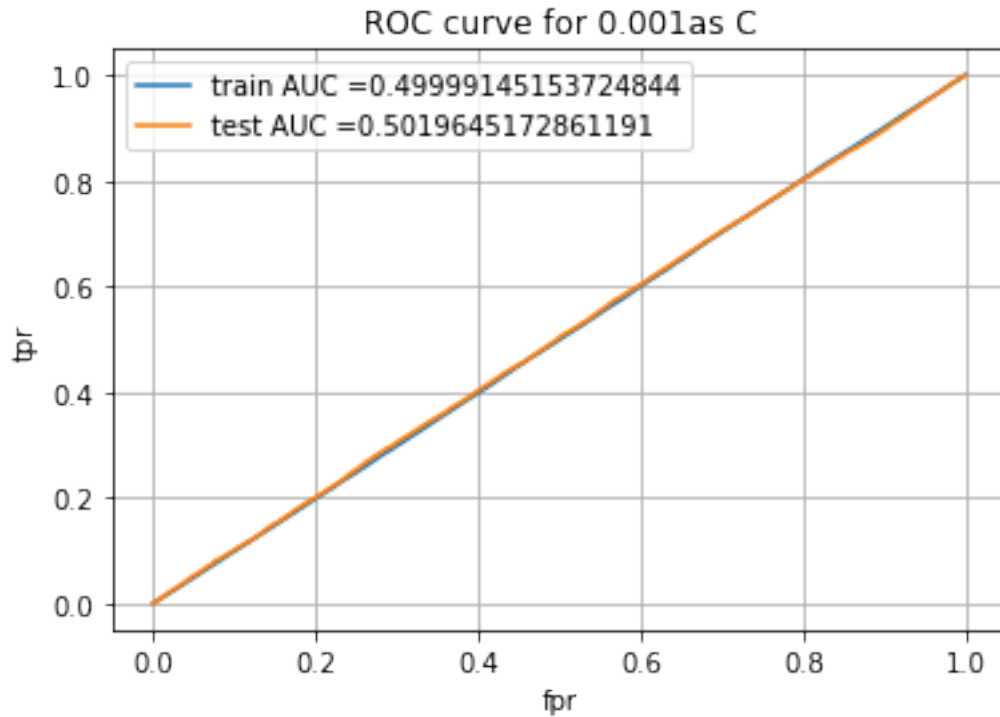


```
[ ]: from keras.models import load_model
model = load_model("SBert.hdf5")

[ ]: 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(threshold, fpr, tpr):
    t = threshold[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, threshold):
    predictions = []
    for i in proba:
        if i>=threshold:
            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 $tpr \cdot (1 - fpr)$ 0.25205766737743474 for threshold 0.712

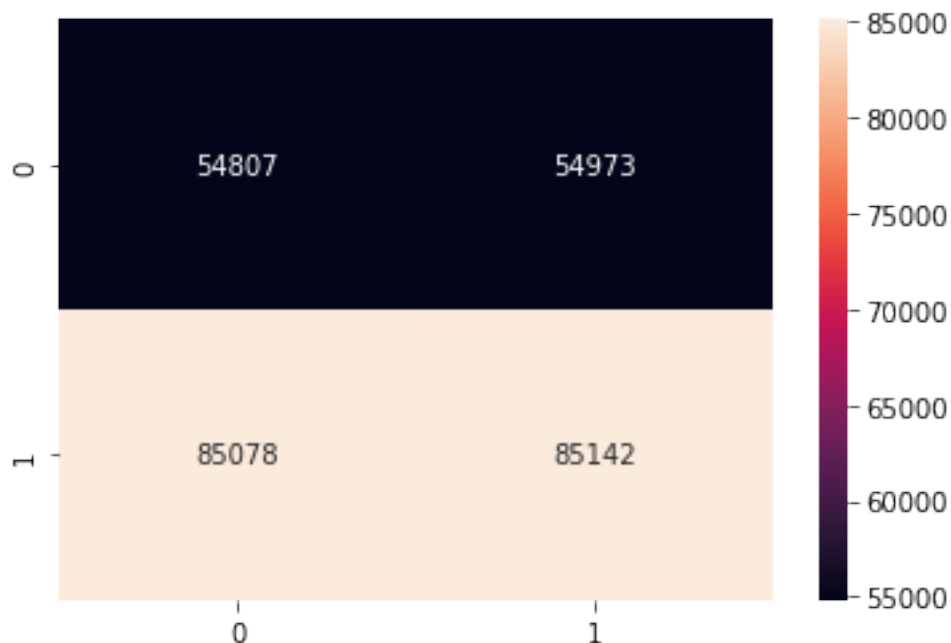
train

the maximum value of $tpr \cdot (1 - fpr)$ 0.24971582508754495 for threshold 0.713

```
[ ]: 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

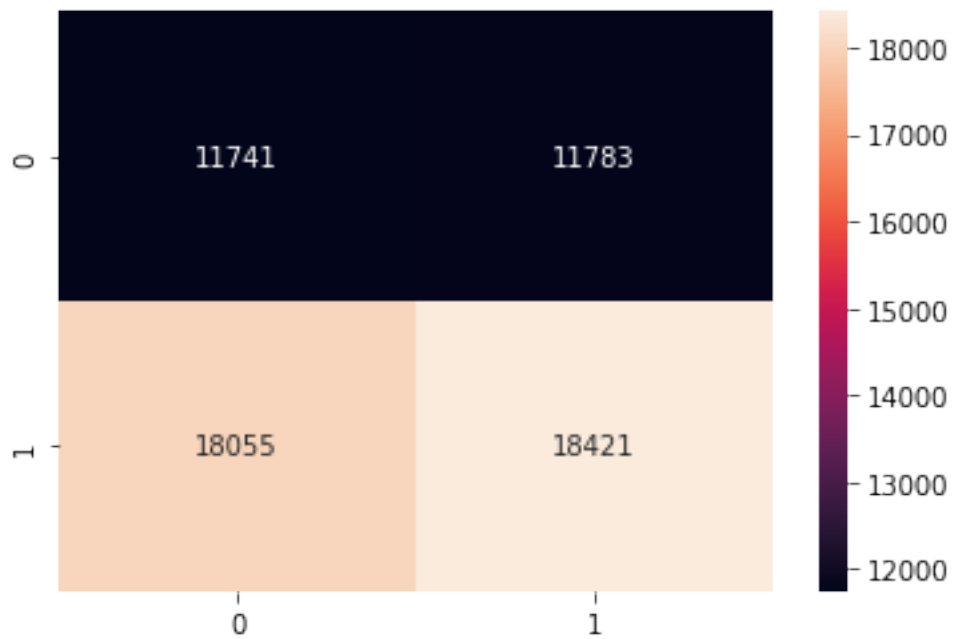
```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f62f8c05410>
```



```
[ ]: 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 0x7f62f8c3c690>
```



```
[ ]: 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: 50.27%
Precision on test set: 60.99%
recall score on test set: 50.50%
f1 score on test set: 55.25%