```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
```

```
%tensorflow_version 1.x
```

```
TensorFlow 1.x selected.
```

```
import numpy as np
import pandas as pd
import re, sys, os, csv
from many_stop_words import get_stop_words
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from collections import Counter
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
!pip install many_stop_words
```

```
     Collecting many stop words
stop_words = list(get_stop_words('en'))        #About 900 stop words
set(stopwords.words('english'))
nltk_words = list(stopwords.words('english'))   #About 150 stop words
stop_words.extend(nltk_words)
     Successfully built many-stop-words
def word_prob(word): return dictionary[word] / total
def words(text): return re.findall('[a-z]+', text.lower())
dictionary = Counter(words(open('merged.txt').read()))
max_word_length = max(map(len, dictionary))
total = float(sum(dictionary.values()))

def viterbi_segment(text):
    probs, lasts = [1.0], [0]
    for i in range(1, len(text) + 1):
        prob_k, k = max((probs[j] * word_prob(text[j:i]), j)
                        for j in range(max(0, i - max_word_length), i))
        probs.append(prob_k)
        lasts.append(k)
    words = []
    i = len(text)
    while 0 < i:
        words.append(text[lasts[i]:i])
        i = lasts[i]
    words.reverse()
    return words, probs[-1]
def fix_hashtag(text):
    text = text.group().split(":")[0]
    text = text[1:] # remove '#'
    try:
        test = int(text[0])
        text = text[1:]
    except:
        pass
    output = ' '.join(viterbi_segment(text)[0])
    #print(output)
    return output

def clean_tweet( tweet):
        tweet = tweet.lower()
        tweet = re.sub("(#[A-Za-z0-9]+)", fix_hashtag, tweet)
```

```
        tweet = re.sub("(#[A-Za-z0-9]+)", fix_hashtag, tweet)
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)",

def remove_stopwords(word_list):
        filtered_tweet=""
        for word in word_list:
            word = word.lower()
            if word not in stopwords.words("english"):
                filtered_tweet=filtered_tweet + " " + word

        return filtered_tweet.lstrip()




data4 = pd.read_csv('/content/data_output1.csv')


data = pd.read_csv('iseardataset.csv')


data = data[['text','label']]


data1= pd.read_csv('data.csv')


data2= pd.read_csv('text_emotion.csv')


data3= pd.read_csv('Equity-Evaluation-Corpus.csv')


data3=data3[['Sentence','Emotion']]


data3[:5]
```

	Sentence	Emotion
0	Alonzo feels angry.	anger
1	Alonzo feels furious.	anger
2	Alonzo feels irritated.	anger
3	Alonzo feels enraged.	anger
4	Alonzo feels annoyed.	anger

```
data2=data2[['sentiment','content']]
```

```
data1.columns = ['text','label']
```

```
data2.columns = ['label','text']
```

```
data3.columns = ['text','label']
```

```
data['label'].value_counts()
```

```
joy        1092
sadness    1082
anger      1079
fear       1076
shame      1071
disgust    1066
guilt      1050
Name: label, dtype: int64
```

```
data3['label'].value_counts()
```

```
joy        2100
sadness    2100
anger      2100
fear       2100
Name: label, dtype: int64
```

```
data2['label'].value_counts()
```

```
neutral       8638
worry         8459
happiness     5209
sadness       5165
love          3842
surprise      2187
fun           1776
relief        1526
hate          1323
empty          827
enthusiasm     759
boredom        179
anger          110
Name: label, dtype: int64
```

```
data1['label'].value_counts()
```

```
1    16297
2    15938
0     9643
3     4301
4     1108
Name: label, dtype: int64
```

```
data1 = data1.loc[data1['label'].isin(['1','2','3','4'])]
```

```
data1['label'].value_counts()
```

```
1    16297
2    15938
3     4301
4     1108
Name: label, dtype: int64
```

```
#data2 = data2.loc[data2['label'].isin(['worry','enthusiasm ','fun','happiness','
```

```
#data2['label'].value_counts()
```

```
worry        8459
happiness    5209
sadness      5165
fun          1776
hate         1323
anger         110
Name: label, dtype: int64
```

```
data = data.loc[data['label'].isin(['fear','joy','sadness','anger'])]
```

```
data['label'].value_counts()
```

```
joy        1092
sadness    1082
anger      1079
fear       1076
Name: label, dtype: int64
```

```
data4 = pd.concat([data1,data,data3], axis=0)
```

```
data4
```

	text	label
0	sickening i hurt for florida later for the wha...	anger
1	this rainfall is a savage y fall when am in d ...	anger
2	angry guy screws his gf in very rude manner po...	anger
3	silence is better when you re angry and frustr...	anger
4	it s your smile which makes me cool and calm w...	anger
...	...	...
50368	The conversation with my mom was funny.	happy
50369	The conversation with my mom was hilarious.	happy
50370	The conversation with my mom was amazing.	happy
50371	The conversation with my mom was wonderful.	happy
50372	The conversation with my mom was great.	happy

50373 rows × 2 columns

```
data4['label'].value_counts()
```

```
happy    19489
sad      19120
hate      4301
anger     4287
fear      3176
Name: label, dtype: int64
```

```
for index, row in data4.iterrows():
    if (row["label"] == 1):
        row["label"] = "happy"
    if (row["label"] == 2):
        row["label"] = "sad"
    if (row["label"] == 3):
        row["label"] = "hate"
    if (row["label"] == 4):
        row["label"] = "anger"
```

```
    if (row["label"] == "sadness"):
        row["label"] = "sad"
    if (row["label"] == "joy"):
        row["label"] = "happy"
```

```
data4 = data4[~data4['label'].isnull()]
data4['label'].value_counts()
```

```
happy    19489
sad      19120
hate      4301
anger     4287
fear      3176
Name: label, dtype: int64
```

```
data4.text=data4.text.astype(str)
```

```
for row in data4['text']:
  tweet= clean_tweet(row)
  tweet = remove_stopwords(tweet.split())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-8-9efb99e96569> in <module>()
      1 for row in data4['text']:
----> 2   tweet= clean_tweet(row)
      3   tweet = remove_stopwords(tweet.split())

NameError: name 'clean_tweet' is not defined
```

SEARCH STACK OVERFLOW

```
data4['label']
```

```
0       anger
1       anger
2       anger
3       anger
4       anger
        ...
8635    happy
8636    happy
8637    happy
8638    happy
8639    happy
Name: label, Length: 50373, dtype: object
```

```
data4.to_csv('data_output1.csv', index=False)
```

```
MAX_NB_WORDS = 40000 # max no. of words for tokenizer
MAX_SEQUENCE_LENGTH = 30 # max length of text (words) including padding
VALIDATION_SPLIT = 0.2
EMBEDDING_DIM = 200 # embedding dimensions for word vectors (word2vec/GloVe)
GLOVE_DIR = "/content/glove.6B.50d.txt"
print("[i] Loaded Parameters:\n",
      MAX_NB_WORDS,MAX_SEQUENCE_LENGTH+5,
      VALIDATION_SPLIT,EMBEDDING_DIM,"\n",
      GLOVE_DIR)
```

```
[i] Loaded Parameters:
 40000 35 0.2 200
 /content/glove.6B.50d.txt
```

```
import numpy as np
import pandas as pd
import re, sys, os, csv, keras, pickle
```

```
Using TensorFlow backend.
```

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils.np_utils import to_categorical
from keras.layers import Embedding
from keras.layers import Dense, Input, Flatten, Concatenate
```

```
from keras.layers import Conv1D, MaxPooling1D,Add, Embedding, Dropout, LSTM, GRU,
from keras.models import Model
from keras import backend as K
from keras.engine.topology import Layer, InputSpec
print("[+] Using Keras version",keras.__version__)
```

```
[+] Using Keras version 2.2.5
```

```
texts=[]
for row in data4['text']:
        texts.append(row)
print("Done!")
```

```
Done!
```

```
texts[:5]
```

```
['sickening i hurt for florida later for the what if s and i know how they feel no you don t i can t imagin',
 'this rainfall is a savage y fall when am in d club n when am home its hot like hell without no light angry',
 'angry guy screws his gf in very rude manner portsmouth',
 'silence is better when you re angry and frustrated reacting to it will fuel the pain',
 'it s your smile which makes me cool and calm when i am sad or angry smile cool calm sad angry edits by birdies yo']
```

```
data4.loc[data4['label'] =='happy' , 'label'] = 0
data4.loc[data4['label'] =='sad' , 'label'] = 1
data4.loc[data4['label'] =='hate' , 'label'] = 2
data4.loc[data4['label'] =='anger' , 'label'] = 3
data4.loc[data4['label'] =='fear' , 'label'] = 4
```

```
data4['label'].value_counts()
```

```
0    19489
1    19120
2     4301
3     4287
4     3176
Name: label, dtype: int64
```

```
#labels=[]
```

```
#for row in data4['label']:
 #       labels.append(row)
#print("Done!")
```

```
Done!
```

```
tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)
```

```
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
print("[i] Saved word tokenizer to file: tokenizer.pickle")
```

```
[i] Saved word tokenizer to file: tokenizer.pickle
```

```
with open('tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)
```

```
sequences = tokenizer.texts_to_sequences(texts)
word_index = tokenizer.word_index
print('[i] Found %s unique tokens.' % len(word_index))
data_int = pad_sequences(sequences, padding='pre', maxlen=(MAX_SEQUENCE_LENGTH-5))
data = pad_sequences(data_int, padding='post', maxlen=(MAX_SEQUENCE_LENGTH))
```

```
[i] Found 32221 unique tokens.
```

```
data[1]
```

```
array([    0,    24, 15191,    14,     4, 15192,   448,   997,    25,
          90,     9,   181,   905,   325,    25,    90,    97,   101,
         360,    47,   583,   294,    50,  1043,    78,     0,     0,
           0,     0,     0], dtype=int32)
```

```
#labels = to_categorical(np.asarray(labels)) # convert to one-hot encoding vector
print('[+] Shape of data tensor:', data.shape)
#print('[+] Shape of label tensor:', labels.shape)
```

```
# Y contained some other garbage, so null check was not enough
#df = df[df['y'].str.isnumeric()]
```

```
labels = to_categorical(data4['label'], num_classes=5)
print(labels[:10])
print('[+] Shape of label tensor:', labels.shape)
```

```
[+] Shape of data tensor: (50373, 30)
[[0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]]
[+] Shape of label tensor: (50373, 5)
```

```
indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])
```

```
x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]

print('[i] Number of entries in each category:')
print("[+] Training:\n",y_train.sum(axis=0))
print("[+] Validation:\n",y_val.sum(axis=0))
```

```
[i] Number of entries in each category:
[+] Training:
 [15600. 15273.  3441.  3453.  2532.]
[+] Validation:
 [3889. 3847.  860.  834.  644.]
```

```
EMBEDDING_DIM = 50
```

```
embeddings_index = {}
f = open(GLOVE_DIR)
```

```
f = open(GLOVE_DIR)
print("[i] Loading GloVe from:",GLOVE_DIR,"...",end="")
for line in f:
    values = line.split()
    word = values[0]
    embeddings_index[word] = np.asarray(values[1:], dtype='float32')
f.close()
print("Done.\n[+] Proceeding with Embedding Matrix...", end="")
embedding_matrix = np.random.random((len(word_index) + 1, EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
print("[i] Completed!")
```

```
[i] Loading GloVe from: /content/glove.6B.50d.txt ...Done.
[+] Proceeding with Embedding Matrix...[i] Completed!
```

```
def get_lr_metric(optimizer):
    def lr(y_true, y_pred):
        return optimizer.lr
    return lr


def initial_boost(epoch):
    if epoch==0: return float(8.0)
    elif epoch==1: return float(4.0)
    elif epoch==2: return float(2.0)
    elif epoch==3: return float(1.5)
    else: return float(1.0)


def step_cyclic(epoch):
    try:
        l_r, decay = 1.0, 0.0001
        if epoch%33==0:multiplier = 10
        else:multiplier = 1
        rate = float(multiplier * l_r * 1/(1 + decay * epoch))
        #print("Epoch",epoch+1,"- learning_rate",rate)
        return rate
    except Exception as e:
        print("Error in lr_schedule:",str(e))
```

```
        return float(1.0)

embedding_matrix_ns = np.random.random((len(word_index) + 1, EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix_ns[i] = embedding_vector
print("Completed!")
```

```
Completed!
```

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')

# static channel
embedding_layer_frozen = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=False)
embedded_sequences_frozen = embedding_layer_frozen(sequence_input)

# non-static channel
embedding_layer_train = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix_ns],
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=True)
embedded_sequences_train = embedding_layer_train(sequence_input)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph is

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is depre

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is d

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:190: The name tf.get_default_session

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:197: The name tf.ConfigProto is depre

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:203: The name tf.Session is deprecate

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:216: The name tf.is_variable_initiali

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:223: The name tf.variables_initialize
```

```
l_lstm1f = Bidirectional(LSTM(6,return_sequences=True,dropout=0.3, recurrent_drop
l_lstm1t = Bidirectional(LSTM(6,return_sequences=True,dropout=0.3, recurrent_drop
l_lstm1 = Concatenate(axis=1)([l_lstm1f, l_lstm1t])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:148: The name tf.placeholder_with_def

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from tensorflo
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

```
from keras import regularizers


l_conv_2 = Conv1D(filters=24,kernel_size=2,activation='relu')(l_lstm1)
l_conv_2 = Dropout(0.3)(l_conv_2)
l_conv_3 = Conv1D(filters=24,kernel_size=3,activation='relu')(l_lstm1)
l_conv_3 = Dropout(0.3)(l_conv_3)

l_conv_5 = Conv1D(filters=24,kernel_size=5,activation='relu',)(l_lstm1)
l_conv_5 = Dropout(0.3)(l_conv_5)
l_conv_6 = Conv1D(filters=24,kernel_size=6,activation='relu',kernel_regularizer=r
```

```
l_conv_6 = Dropout(0.3)(l_conv_6)

l_conv_8 = Conv1D(filters=24,kernel_size=8,activation='relu',kernel_regularizer=r
l_conv_8 = Dropout(0.3)(l_conv_8)

conv_1 = [l_conv_6,l_conv_5, l_conv_8,l_conv_2,l_conv_3]

l_lstm_c = Concatenate(axis=1)(conv_1)


l_conv_4f = Conv1D(filters=12,kernel_size=4,activation='relu',kernel_regularizer=
l_conv_4f = Dropout(0.3)(l_conv_4f)
l_conv_4t = Conv1D(filters=12,kernel_size=4,activation='relu',kernel_regularizer=
l_conv_4t = Dropout(0.3)(l_conv_4t)

l_conv_3f = Conv1D(filters=12,kernel_size=3,activation='relu',)(embedded_sequence
l_conv_3f = Dropout(0.3)(l_conv_3f)
l_conv_3t = Conv1D(filters=12,kernel_size=3,activation='relu',)(embedded_sequence
l_conv_3t = Dropout(0.3)(l_conv_3t)

l_conv_2f = Conv1D(filters=12,kernel_size=2,activation='relu')(embedded_sequences
l_conv_2f = Dropout(0.3)(l_conv_2f)
l_conv_2t = Conv1D(filters=12,kernel_size=2,activation='relu')(embedded_sequences
l_conv_2t = Dropout(0.3)(l_conv_2t)

conv_2 = [l_conv_4f, l_conv_4t,l_conv_3f, l_conv_3t, l_conv_2f, l_conv_2t]

l_merge_2 = Concatenate(axis=1)(conv_2)
l_c_lstm = Bidirectional(LSTM(12,return_sequences=True,dropout=0.3, recurrent_drc


l_merge = Concatenate(axis=1)([l_lstm_c, l_c_lstm])
l_pool = MaxPooling1D(4)(l_merge)
l_drop = Dropout(0.5)(l_pool)
l_flat = Flatten()(l_drop)
l_dense = Dense(26, activation='relu')(l_flat)
preds = Dense(5, activation='softmax')(l_dense)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:4267: The name tf.nn.max_pool is depr
```

```
from keras import optimizers
```

```
from keras import callbacks


model = Model(sequence_input, preds)
adadelta = optimizers.Adadelta(lr=0.9, rho=0.95, epsilon=None, decay=0.002)
lr_metric = get_lr_metric(adadelta)
model.compile(loss='categorical_crossentropy',
              optimizer=adadelta,
              metrics=['acc'])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated. Pleas

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated.
```

```
tensorboard = callbacks.TensorBoard(log_dir='./logs', histogram_freq=0, batch_siz
model_checkpoints = callbacks.ModelCheckpoint("checkpoint-{val_loss:.3f}.h5", mor
lr_schedule = callbacks.LearningRateScheduler(initial_boost)


model.summary()
model.save('BalanceNet.h5')
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 30)           0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 30, 50)       1611100     input_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, 30, 50)       1611100     input_1[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, 30, 12)       2736        embedding_1[0][0]
__________________________________________________________________________________________________
bidirectional_2 (Bidirectional) (None, 30, 12)       2736        embedding_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 60, 12)       0           bidirectional_1[0][0]
                                                                 bidirectional_2[0][0]
__________________________________________________________________________________________________
conv1d_6 (Conv1D)               (None, 27, 12)       2412        embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_7 (Conv1D)               (None, 27, 12)       2412        embedding_2[0][0]
__________________________________________________________________________________________________
conv1d_8 (Conv1D)               (None, 28, 12)       1812        embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_9 (Conv1D)               (None, 28, 12)       1812        embedding_2[0][0]
__________________________________________________________________________________________________
conv1d_10 (Conv1D)              (None, 29, 12)       1212        embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_11 (Conv1D)              (None, 29, 12)       1212        embedding_2[0][0]
__________________________________________________________________________________________________
conv1d_4 (Conv1D)               (None, 55, 24)       1752        concatenate_1[0][0]
__________________________________________________________________________________________________
conv1d_3 (Conv1D)               (None, 56, 24)       1464        concatenate_1[0][0]
__________________________________________________________________________________________________
conv1d_5 (Conv1D)               (None, 53, 24)       2328        concatenate_1[0][0]
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 59, 24)       600         concatenate_1[0][0]
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 58, 24)       888         concatenate_1[0][0]
__________________________________________________________________________________________________
dropout_6 (Dropout)             (None, 27, 12)       0           conv1d_6[0][0]
__________________________________________________________________________________________________
dropout_7 (Dropout)             (None, 27, 12)       0           conv1d_7[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 28, 12)       0           conv1d_8[0][0]
__________________________________________________________________________________________________
dropout_9 (Dropout)             (None, 28, 12)       0           conv1d_9[0][0]
```

```
__________________________________________________________________________________________________
dropout_10 (Dropout)            (None, 29, 12)       0           conv1d_10[0][0]
__________________________________________________________________________________________________
dropout_11 (Dropout)            (None, 29, 12)       0           conv1d_11[0][0]
__________________________________________________________________________________________________
dropout_4 (Dropout)             (None, 55, 24)       0           conv1d_4[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 56, 24)       0           conv1d_3[0][0]
__________________________________________________________________________________________________
dropout_5 (Dropout)             (None, 53, 24)       0           conv1d_5[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 59, 24)       0           conv1d_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 58, 24)       0           conv1d_2[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 168, 12)      0           dropout_6[0][0]
                                                                 dropout_7[0][0]
                                                                 dropout_8[0][0]
                                                                 dropout_9[0][0]
                                                                 dropout_10[0][0]
                                                                 dropout_11[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 281, 24)      0           dropout_4[0][0]
                                                                 dropout_3[0][0]
                                                                 dropout_5[0][0]
                                                                 dropout_1[0][0]
                                                                 dropout_2[0][0]
__________________________________________________________________________________________________
bidirectional_3 (Bidirectional) (None, 168, 24)      2400        concatenate_3[0][0]
__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 449, 24)      0           concatenate_2[0][0]
                                                                 bidirectional_3[0][0]
__________________________________________________________________________________________________
max_pooling1d_1 (MaxPooling1D)  (None, 112, 24)      0           concatenate_4[0][0]
__________________________________________________________________________________________________
dropout_12 (Dropout)            (None, 112, 24)      0           max_pooling1d_1[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 2688)         0           dropout_12[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 26)           69914       flatten_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 5)            135         dense_1[0][0]
==================================================================================================
Total params: 3,318,025
Trainable params: 1,706,925
Non-trainable params: 1,611,100
```

```
print("Training Progress:")
model_log = model.fit(x_train, y_train, validation_data=(x_val, y_val),epochs=25,

pandas.DataFrame(model_log.history).to_csv("history-balance.csv")
```

```
-1.15.2/python3.6/tensorflow_core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array_ops) is deprecated and will be re

same broadcast rule as np.where
lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please use tf.compat.v1.assig

lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please use tf.compat.v1.assign in

 10074 samples

========] - 141s 4ms/step - loss: 1.2252 - acc: 0.4412 - val_loss: 1.0401 - val_acc: 0.5344

========] - 134s 3ms/step - loss: 1.0453 - acc: 0.5253 - val_loss: 0.9275 - val_acc: 0.6006

========] - 136s 3ms/step - loss: 0.9559 - acc: 0.5743 - val_loss: 0.8682 - val_acc: 0.6390

========] - 134s 3ms/step - loss: 0.9049 - acc: 0.6084 - val_loss: 0.8277 - val_acc: 0.6647

========] - 135s 3ms/step - loss: 0.8660 - acc: 0.6318 - val_loss: 0.7962 - val_acc: 0.6805

========] - 135s 3ms/step - loss: 0.8305 - acc: 0.6527 - val_loss: 0.7659 - val_acc: 0.6961

========] - 135s 3ms/step - loss: 0.8045 - acc: 0.6627 - val_loss: 0.7430 - val_acc: 0.7071

========] - 135s 3ms/step - loss: 0.7803 - acc: 0.6764 - val_loss: 0.7304 - val_acc: 0.7080

========] - 138s 3ms/step - loss: 0.7618 - acc: 0.6845 - val_loss: 0.7138 - val_acc: 0.7132

========] - 135s 3ms/step - loss: 0.7499 - acc: 0.6888 - val_loss: 0.6991 - val_acc: 0.7177

========] - 135s 3ms/step - loss: 0.7360 - acc: 0.6982 - val_loss: 0.6903 - val_acc: 0.7213

========] - 136s 3ms/step - loss: 0.7314 - acc: 0.6987 - val_loss: 0.6844 - val_acc: 0.7215

========] - 135s 3ms/step - loss: 0.7204 - acc: 0.7045 - val_loss: 0.6800 - val_acc: 0.7230

========] - 140s 3ms/step - loss: 0.7148 - acc: 0.7041 - val_loss: 0.6739 - val_acc: 0.7252

========] - 135s 3ms/step - loss: 0.7079 - acc: 0.7090 - val_loss: 0.6709 - val_acc: 0.7255

========] - 134s 3ms/step - loss: 0.6985 - acc: 0.7108 - val_loss: 0.6679 - val_acc: 0.7277

========] - 136s 3ms/step - loss: 0.6958 - acc: 0.7155 - val_loss: 0.6627 - val_acc: 0.7301

========] - 140s 3ms/step - loss: 0.6928 - acc: 0.7153 - val_loss: 0.6610 - val_acc: 0.7301
```

```
========] - 135s 3ms/step - loss: 0.6884 - acc: 0.7170 - val_loss: 0.6593 - val_acc: 0.7300

========] - 137s 3ms/step - loss: 0.6837 - acc: 0.7181 - val_loss: 0.6547 - val_acc: 0.7333

========] - 136s 3ms/step - loss: 0.6779 - acc: 0.7219 - val_loss: 0.6541 - val_acc: 0.7344

========] - 135s 3ms/step - loss: 0.6760 - acc: 0.7228 - val_loss: 0.6504 - val_acc: 0.7362

========] - 137s 3ms/step - loss: 0.6730 - acc: 0.7228 - val_loss: 0.6518 - val_acc: 0.7362

========] - 135s 3ms/step - loss: 0.6729 - acc: 0.7241 - val_loss: 0.6479 - val_acc: 0.7375

========] - 136s 3ms/step - loss: 0.6678 - acc: 0.7265 - val_loss: 0.6448 - val_acc: 0.7386
---------------------------------------
       Traceback (most recent call last)
<module>()
in, y_train, validation_data=(x_val, y_val),epochs=25, batch_size=128)

history).to_csv("history-balance.csv")

ined
```

```
!pip install keras==2.2.5
```

```
Collecting keras==2.2.5
  Downloading https://files.pythonhosted.org/packages/f8/ba/2d058dcf1b85b9c212cc58264c98a4a7dd92c989b798823cc5690d062bb2/Keras-2.2.5-py2
     |████████████████████████████████| 337kB 2.8MB/s
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.6/dist-packages (from keras==2.2.5) (1.4.1)
```

```
model_log = model.fit(x_train, y_train, validation_data=(x_val, y_val),
          epochs=3, batch_size=128,
          callbacks=[tensorboard, model_checkpoints])

pd.DataFrame(model_log.history).to_csv("history-balance1.csv")
```

```
validate on 10074 samples
/usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name tf.summary.merge_all is deprecated. Please use tf.compat.v1.sum

/usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.su


====================] - 134s 3ms/step - loss: 0.6677 - acc: 0.7241 - val_loss: 0.6449 - val_acc: 0.7389
/usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name tf.Summary is deprecated. Please use tf.compat.v1.Summary inste


====================] - 134s 3ms/step - loss: 0.6635 - acc: 0.7283 - val_loss: 0.6445 - val_acc: 0.7389

====================] - 135s 3ms/step - loss: 0.6649 - acc: 0.7298 - val_loss: 0.6426 - val_acc: 0.7393
```

```
pd.DataFrame(model_log.history).to_csv("history-balance.csv")
```

```
from keras.models import load_model
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import numpy as np
%config InlineBackend.figure_format = 'retina'
import itertools, pickle

with open('tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)
```

```
classes = [ "happy", "sad", "hate","anger","fear"]
```

```
#model_test = load_model('checkpoint-0.866.h5')
#model_test = load_model('best_weights.h5')
Y_test = np.argmax(y_val, axis=1) # Convert one-hot to index
#y_pred = model_test.predict(x_val)
y_pred = model.predict(x_val)
y_pred_class = np.argmax(y_pred,axis=1)
cnf_matrix = confusion_matrix(Y_test, y_pred_class)
```

```
print(classification_report(Y_test, y_pred_class, target_names=classes))
```

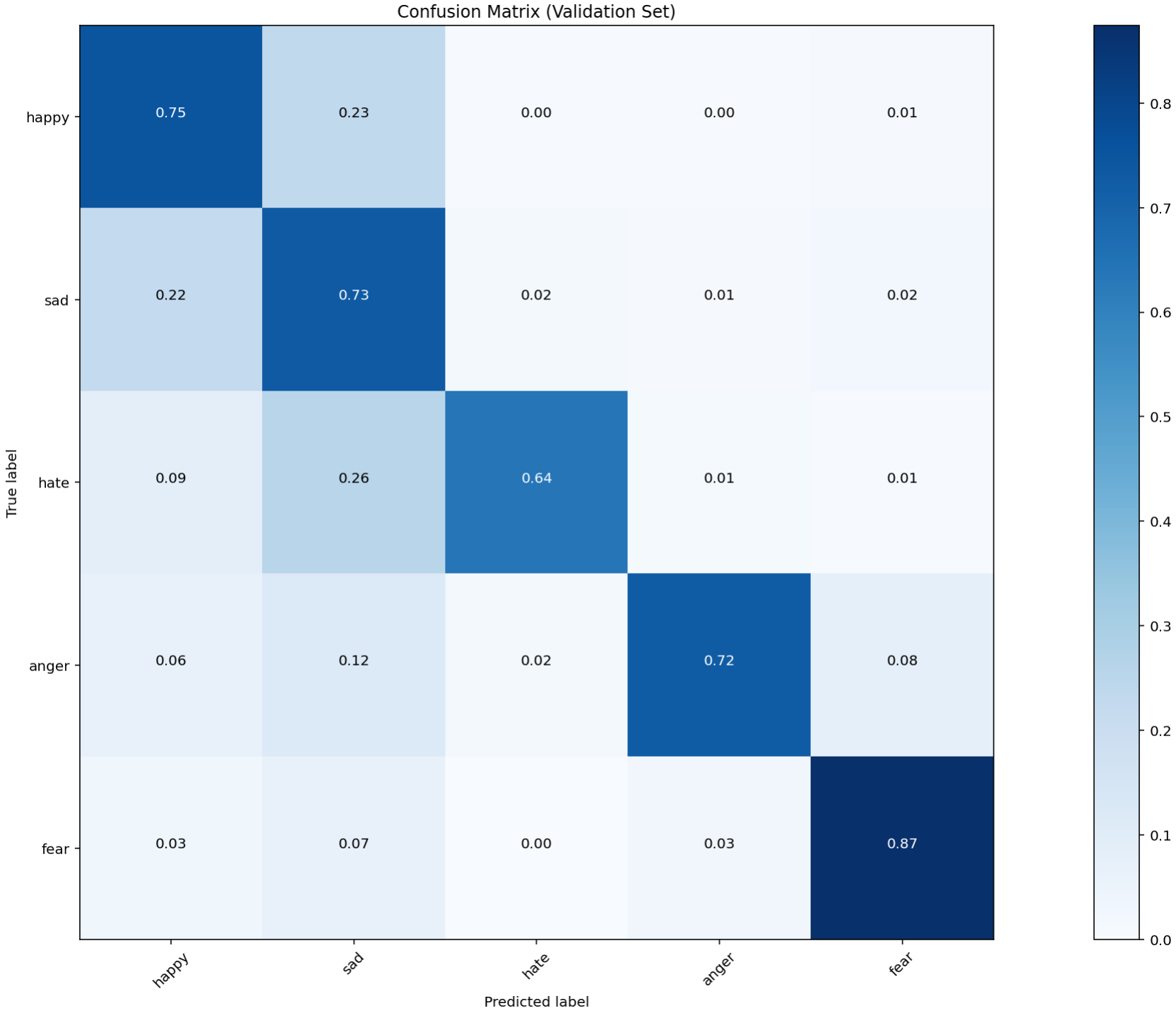
```
              precision    recall  f1-score   support

       happy       0.74      0.75      0.75      3889
         sad       0.69      0.73      0.71      3847
        hate       0.86      0.64      0.73       860
       anger       0.88      0.72      0.80       834
        fear       0.75      0.87      0.81       644

    accuracy                           0.74     10074
   macro avg       0.79      0.74      0.76     10074
weighted avg       0.75      0.74      0.74     10074
```

```
score,acc = model.evaluate(x_val, y_val, verbose = 2, batch_size=128)
print("loss: %.2f" % (score))
print("acc: %.2f" % (acc))
```

```
loss: 0.64
acc: 0.74
```

```
def plot_confusion_matrix(cm, labels,
                          normalize=True,
                          title='Confusion Matrix (Validation Set)',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
```

```
        #print("Normalized confusion matrix")
    else:
        #print('Confusion matrix, without normalization')
        pass

    #print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(labels))
    plt.xticks(tick_marks, labels, rotation=45)
    plt.yticks(tick_marks, labels)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

plt.figure(figsize=(20,10))
plot_confusion_matrix(cnf_matrix, labels=classes)

# precision = true_pos / (true_pos + false_pos)
# recall = true_pos / (true_pos + false_neg)
```


Confusion Matrix (Validation Set)
True label
Predicted label
happy
sad
hate
anger
fear
0.75 0.23 0.00 0.00 0.01
0.22 0.73 0.02 0.01 0.02
0.09 0.26 0.64 0.01 0.01
0.06 0.12 0.02 0.72 0.08
0.03 0.07 0.00 0.03 0.87
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

```
l sacrifice! A true hero indeed.",
ıned sight more than the PAP and it's cronies! Off course the PAP will say that tl

ıy it; Instead, he bought a kidney then bought the car Filthy rich This is why we
s head, hair needs to be grown there",
ean and flat footpath,,now obstructed by sharedbikes..! which idiotic MP allowed t


sequences_test = tokenizer.texts_to_sequences(text)
data_int_t = pad_sequences(sequences_test, padding='pre', maxlen=(MAX_SEQUENCE_LE
data_test = pad_sequences(data_int_t, padding='post', maxlen=(MAX_SEQUENCE_LENGTH
y_prob = model.predict(data_test)
for n, prediction in enumerate(y_prob):
    pred = y_prob.argmax(axis=-1)[n]
    print(text[n],"\nPrediction:",classes[pred],"\n")
```

```
I salute you for the bravery and sacrifice! A true hero indeed.
Prediction: happy
```

```
text = ["never talk to me again",
        "do not get angry or frustrated or desperate or enraged or depressed or a
        "i hate worthless insights",
        "it is the worst day of my life",
        "i love you mom",
        "stop saying bullshit",
        "congratulations on your acceptance",
        "your stupidity has no limt",
        "sounds like a fun plan",
        "i will celebrate soon",
        "the game just finished",
        "you are so mean"]
```

```
y_prob
```

```
array([[2.57533669e-01, 5.87897360e-01, 6.97120428e-02, 4.85533290e-02,
        3.63036580e-02],
       [1.49845751e-02, 1.04981579e-01, 2.99310070e-02, 8.36996913e-01,
        1.31059131e-02],
       [2.26184912e-03, 2.86977962e-02, 9.62217271e-01, 6.73174951e-03,
        9.13754993e-05],
       [1.64339557e-01, 7.34030664e-01, 6.75033852e-02, 2.14170236e-02,
        1.27093801e-02],
       [8.73809397e-01, 1.13107353e-01, 1.12590361e-02, 1.47951732e-03,
        3.44655302e-04],
       [5.21720573e-02, 3.35852087e-01, 3.99300098e-01, 1.45041198e-01,
        6.76345527e-02],
       [9.43086445e-01, 5.15052602e-02, 4.72402712e-03, 5.92601718e-04,
        9.17235302e-05],
       [1.21354513e-01, 5.37909508e-01, 3.28266293e-01, 1.16918338e-02,
        7.77969137e-04],
       [9.40767050e-01, 5.12322932e-02, 7.11831078e-03, 7.90400489e-04,
        9.20164166e-05],
       [9.04147208e-01, 8.90096202e-02, 5.52371563e-03, 1.10297767e-03,
        2.16531116e-04],
       [7.75976062e-01, 1.83728963e-01, 2.27153078e-02, 1.24066025e-02,
        5.17309923e-03],
       [3.38026494e-01, 5.21076441e-01, 1.31740004e-01, 8.47787224e-03,
        6.79206452e-04]], dtype=float32)
```

```
sequences_test = tokenizer.texts_to_sequences(text)
data_int_t = pad_sequences(sequences_test, padding='pre', maxlen=(MAX_SEQUENCE_LE
data_test = pad_sequences(data_int_t, padding='post', maxlen=(MAX_SEQUENCE_LENGTH
y_prob = model.predict(data_test)
for n, prediction in enumerate(y_prob):
    pred = y_prob.argmax(axis=-1)[n]
    print(text[n],"\nPrediction:",classes[pred],"\n")
```

```
never talk to me again
Prediction: sad

do not get angry or frustrated or desperate or enraged or depressed or any such thing you are all educated
Prediction: anger

i hate worthless insights
Prediction: hate

it is the worst day of my life
Prediction: sad

i love you mom
Prediction: happy

stop saying bullshit
Prediction: hate

congratulations on your acceptance
Prediction: happy

your stupidity has no limt
Prediction: sad

sounds like a fun plan
Prediction: happy

i will celebrate soon
Prediction: happy

the game just finished
Prediction: happy

you are so mean
Prediction: sad
```

```
text='''Once upon a time Poor Cinderella had to work hard all day long so the oth

“What a mess!” her two stepsisters laughed.  And that is why they called her “Cir

One day, big news came to town.  The King and Queen were going to have a ball!  I

At Cinderella’s house, she now had extra work to do.  She had to make two brand-r

“Faster!” shouted one step-sister.

“You call that a dress?” screamed the other.

“Oh, dear!” said Cinderella.  “When can I–“

The stepmother marched into the room.  “When can you WHAT?”

“Well,” said the girl, “when will I have time to make my own dress for the ball?”

“You?” yelled the stepmother.  “Who said YOU were going to the ball?”

“What a laugh!” said one step-sister.

“Such a mess!” They pointed at Cinderella.  All of them laughed.

Cinderella said to herself, “When they look at me, maybe they see a mess.  But I

Soon the time came for the stepmother and step-sisters to leave for the big party

Their fine carriage came to the door. The stepmother and step-sisters hopped insi

“Good-bye!” called Cinderella.  “Have a good time!” But her stepmother and step-s

“Ah, me!” said Cinderella sadly.  The carriage rode down the street.  She said al

Then – Poof!
```

All of a sudden, in front of her was a fairy.

“You called?” said the fairy.

“Did I?” said Cinderella. “Who are you?”

“Why, your Fairy Godmother, of course! I know your wish. And I have come to gra

“But…” said Cinderella, “my wish is impossible.”

“Excuse me!” said the Fairy Godmother in a huff. “Did I not just show up out of

“Yes, you did,” said Cinderella.

“Then let me be the one to say what is possible or not!”

“Well, I think you know I want to go to the ball, too.” She looked down at her di

 “But look at me.”

“You do look a bit of a mess, child,” said the Fairy Godmother.

“Even if I had something nice to wear,” said the girl, “I would have no way to ge

“Dear me, all of that is possible,” said the Fairy. With that, she tapped her war

At once, Cinderella was all clean. She was dressed in a beautiful blue gown. He

“This is wonderful!” said Cinderella.

“Who said I was done?” said the Fairy Godmother. She tapped her wand again. At c

“Am I dreaming?” said Cinderella, looking around her.

“It is as real, as real can be,” said the Fairy Godmother. “But there is one thi

“What is that?”

“All of this lasts only to midnight. Tonight, at the stroke of midnight, it will

“Then I must be sure to leave the ball before midnight!” said Cinderella.

“Good idea,” said the Fairy Godmother. She stepped back. “My work is done.” And

Cinderella looked around her. “Did that even happen?” But there she stood in a

“Coming?” called the driver.

She stepped into the carriage. And they were off.

Over at the ball, the Prince did not know what to think. “Why do you have that s

“I know, Mother,” said the Prince. Yet he knew something was wrong. He had met m

“Look!” Someone pointed to the front door. “Who is that?”

All heads turned. Who was that lovely maiden stepping down the stairs? She helc

“There is something about her,” said the Prince to himself. “I will ask her to c

“Have we met?” said the Prince.

“I am pleased to meet you now,” said Cinderella with a bow.

“I feel as if I know you,” said the Prince. “But of course, that is impossible.”

“Many things are possible,” said Cinderella, “if you wish them to be true.”

The Prince felt a leap in his heart. He and Cinderella danced. When the song was

But all the Prince could see was Cinderella. They laughed and talked, and they c

“Dong!” said the clock.

Cinderella looked up.

“Dong!” went the clock again.

She looked up again. “Oh, my!” she cried out. “It is almost midnight!”

“Dong!” rung the clock.

“Why does that matter?” said the Prince.

“Dong!” called the clock.

“I must go!” said Cinderella.

“Dong!” went the clock.

“But we just met!” said the Prince. “Why leave now?”

“Dong!” rung the clock.

“I must GO!” said Cinderella. She ran to the steps.

“Dong!” said the clock.

“I cannot hear you,” said the Prince. “The clock is too loud!”

“Dong!” rung the clock.

“Goodbye!” said Cinderella. Up, up the stairs she ran.

“Dong!” went the clock.

“Please, stop for a moment!” said the Prince.

“Oh, dear!” she said as one glass slipper fell off her foot on the stair. But Ci

“Dong!” said the clock.

“Please wait a moment!” said the Prince.

“Dong!” rung the clock.

“Goodbye!” Cinderella turned one last time. Then she rushed out the door.

“Dong!” The clock was quiet. It was midnight.

“Wait!” called the Prince. He picked up her glass slipper and rushed out the doc

From hut to hut, from house to house, went the Prince. One young woman after anc

At last the Prince came to Cinderella’s house.

“He is coming!” called one step-sister as she looked out the window.

“At the door!” screamed the other step-sister.

“Quick!” yelled the stepmother. “Get ready! One of you must be the one to fit yc

The Prince knocked. The stepmother flew open the door. “Come in!” she said. “I

The first step-sister tried to place her foot in the glass slipper. She tried ha

“Are there no other young women in the house?” said the Prince.

“None,” said the stepmother.

“Then I must go,” said the Prince.

“Maybe there is one more,” said Cinderella, stepping into the room.

“I thought you said there were no other young women here,” said the Prince.

“None who matter!” said the stepmother in a hiss.

“Come here,” said the prince.

Cinderella stepped up to him. The Prince got down on one knee and tried the glas

“I knew it!” he cried. “You are the one!”

“WHAT?” shouted a step-sister.

“Not HER!” screamed the other step-sister.

“This cannot BE!” yelled the stepmother.

But it was too late. The prince knew that Cinderella was the one. He looked int

“I have found you!” he said.

“And I have found you,” said Cinderella.

```
And so Cinderella and the Prince were married, and they lived happily ever after.
```

```
x = text.split(".")

for row in x:
  y=clean_tweet(row)
  z=remove_stopwords(y.split())
  row=z
```

```
sequences_test = tokenizer.texts_to_sequences(x)
data_int_t = pad_sequences(sequences_test, padding='pre', maxlen=(MAX_SEQUENCE_LE
data_test = pad_sequences(data_int_t, padding='post', maxlen=(MAX_SEQUENCE_LENGTH
print(data_test[0])
y_prob = model.predict(data_test)
```

```
[   0    0    0    0    0    0    0    0  437 2008    4   60  375   37
    3   68  332   22   31  184   21    2  775  144  699    0    0    0
    0    0]
```

```
for n, prediction in enumerate(y_prob):
    pred = y_prob.argmax(axis=-1)[n]
    print(x[n],"\nPrediction:",classes[pred],"\n")
```

```
Once upon a time Poor Cinderella had to work hard all day long so the others could rest
Prediction: sad

 It was she who had to wake up each morning when it was still dark and cold to start the fire
Prediction: fear

  It was she who cooked the meals
Prediction: happy

 It was she who kept the fire going
Prediction: fear

 The poor girl could not stay clean, from all the ashes and cinders by the fire
Prediction: sad



“What a mess!” her two stepsisters laughed
Prediction: happy

  And that is why they called her “Cinderella
Prediction: sad

”

One day, big news came to town
Prediction: happy

  The King and Queen were going to have a ball!  It was time for the Prince to find a bride
Prediction: happy

 All of the young ladies in the land were invited to come
Prediction: happy

  They were wild with joy! They would wear their most beautiful gown and fix their hair extra nice
Prediction: happy

 Maybe the prince would like them!

At Cinderella’s house, she now had extra work to do
Prediction: happy

  She had to make two brand-new gowns for her step-sisters
Prediction: happy
```