Header files

In [2]:
```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import pandas as pd
import seaborn as sns
import numpy as np
import pickle
from sklearn import preprocessing
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import log_loss
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from gensim.scripts.glove2word2vec import glove2word2vec
from keras.callbacks import EarlyStopping
from keras.layers import Bidirectional,BatchNormalization,Flatten,Dropout,Max
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import np_utils
from keras.models import Model
from keras import backend as K
from datetime import datetime
from tensorflow.keras.models import model_from_json
```

Download dataset

In [3]:
```
!wget --header="Host: codeload.github.com" --header="User-Agent: Mozilla/5.0
```

```
--2020-07-11 02:44:54--  https://codeload.github.com/declare-lab/MELD/zip/ma
ster (https://codeload.github.com/declare-lab/MELD/zip/master)
Resolving codeload.github.com (codeload.github.com)... 140.82.114.10
Connecting to codeload.github.com (codeload.github.com)|140.82.114.10|:44
3... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ‘MELD-master.zip’

MELD-master.zip         [  <=>               ]   2.08M  7.15MB/s    in 0.3s

2020-07-11 02:44:55 (7.15 MB/s) - ‘MELD-master.zip’ saved [2180995]
```

In [4]:
```
!unzip /content/MELD-master.zip
```

```
Archive:  /content/MELD-master.zip
677afb1ba7dadfeab3668322314d37ccce9b2bb2
   creating: MELD-master/
  inflating: MELD-master/LICENSE
  inflating: MELD-master/README.md
   creating: MELD-master/baseline/
  inflating: MELD-master/baseline/baseline.py
  inflating: MELD-master/baseline/data_helpers.py
   creating: MELD-master/data/
   creating: MELD-master/data/MELD/
  inflating: MELD-master/data/MELD/dev_sent_emo.csv
  inflating: MELD-master/data/MELD/test_sent_emo.csv
  inflating: MELD-master/data/MELD/train_sent_emo.csv
   creating: MELD-master/data/MELD_Dyadic/
  inflating: MELD-master/data/MELD_Dyadic/dev_sent_emo_dya.csv
  inflating: MELD-master/data/MELD_Dyadic/test_sent_emo_dya.csv
  inflating: MELD-master/data/MELD_Dyadic/train_sent_emo_dya.csv
   creating: MELD-master/data/emorynlp/
  inflating: MELD-master/data/emorynlp/emorynlp_dev_final.csv
  inflating: MELD-master/data/emorynlp/emorynlp_test_final.csv
  inflating: MELD-master/data/emorynlp/emorynlp_train_final.csv
   creating: MELD-master/data/pickles/
  inflating: MELD-master/data/pickles/download-features.txt
   creating: MELD-master/images/
  inflating: MELD-master/images/emotion_shift.jpeg
  inflating: MELD-master/images/sc4.png
   creating: MELD-master/utils/
  inflating: MELD-master/utils/read_emorynlp.py
  inflating: MELD-master/utils/read_meld.py
```

In [5]:
```
train_data = pd.read_csv('/content/MELD-master/data/MELD/train_sent_emo.csv')
cv_data = pd.read_csv('/content/MELD-master/data/MELD/dev_sent_emo.csv')
test_data = pd.read_csv('/content/MELD-master/data/MELD/test_sent_emo.csv')
```

In [6]:
```
train_data.head(5)
```

Out[6]:

	Sr No.	Utterance	Speaker	Emotion	Sentiment	Dialogue_ID	Utterance_ID	Season	Episode	StartTime
0	1	also I was the point person on my companys tr...	Chandler	neutral	neutral	0	0	8	21	00:16:16,059
1	2	You mustve had your hands full.	The Interviewer	neutral	neutral	0	1	8	21	00:16:21,940
2	3	That I did. That I did.	Chandler	neutral	neutral	0	2	8	21	00:16:23,442
3	4	So lets talk a little bit about your duties.	The Interviewer	neutral	neutral	0	3	8	21	00:16:26,820
4	5	My duties? All right.	Chandler	surprise	positive	0	4	8	21	00:16:34,452

In [7]:
```
print(train_data.shape)
print(cv_data.shape)
print(test_data.shape)
```

```
(9989, 11)
(1109, 11)
(2610, 11)
```

In [8]:
```
train_data.Utterance.values[:100]
```

Out[8]:
```
array(['also I was the point person on my company\x92s transition from the K
L-5 to GR-6 system.',
       'You must\x92ve had your hands full.', 'That I did. That I did.',
       'So let\x92s talk a little bit about your duties.',
       'My duties?  All right.',
       'Now you\x92ll be heading a whole division, so you\x92ll have a lot o
f duties.',
       'I see.',
       'But there\x92ll be perhaps 30 people under you so you can dump a cer
tain amount on them.',
       'Good to know.', 'We can go into detail',
       'No don\x92t I beg of you!',
       'All right then, we\x92ll have a definite answer for you on Monday, b
ut I think I can say with some confidence, you\x92ll fit in well here.',
       'Really?!', 'Absolutely.  You can relax',
       'But then who? The waitress I went out with last month?',
       'You know? Forget it!',
       'No-no-no-no, no! Who, who were you talking about?',
       "No, I-I-I-I don't, I actually don't know", 'Ok!',
       'All right, well...', 'Yeah, sure!', 'Hey, Mon.',
       'Hey-hey-hey. You wanna hear something that sucks.', 'Do I ever.',
       'Chris says they\x92re closing down the bar.', 'No way!',
       'Yeah, apparently they\x92re turning it into some kinda coffee plac
e.',
       'Just coffee! Where are we gonna hang out now?', 'Got me.',
       'Can I get a beer.', 'Hey, did you pick a roommate?',
       'You betcha!', 'Is it the Italian guy?', 'Um-mm, yeah right!',
       'Oh my God, oh my God! Poor Monica!', 'What, what, what?!',
       'What?!', 'He was with her when he wrote this poem.',
       "Look,  'My vessel so empty with nothing inside.",
       "Now that I've touched you, you seem emptier still.'",
       'He thinks Monica is empty, she is the empty vase!',
       'Oh, totally. Oh, God, oh, she seemed so happy too.', 'Done.',
       'Hey!', 'Hi!', 'What are you doing here?',
       "Ah y'know, this building is on my paper route so I...", 'Oh.',
       'Hi.', 'Hi.', 'How\x92d did it go?',
       "Oh well, the woman I interviewed with was pretty tough, but y'know t
hank God Mark coached me, because once I started talking about the fall lin
e, she got all happy and wouldn\x92t shut up.",
       'I\x92m so proud of you.', 'Me too!',
       'Listen, I\x92m ah, I\x92m sorry I\x92ve been so crazy and jealous an
d, it\x92s just that I like you a lot, so...',
       'I know.', 'Yeah.', 'Yeah.', 'Ameri-can.', 'Ameri-ccan.',
       "Ameri-can. Y'know it\x92s a", 'Everybody!!',
       'Good job Joe! Well done! Top notch!',
       'You liked it? You really liked it?', 'Oh-ho-ho, yeah!',
       'Which part exactly?', 'The whole thing! Can we go?',
       'Oh no-no-no, give me some specifics.',
       'I love the specifics, the specifics were the best part!',
       'Hey, what about the scene with the kangaroo? Did-did you like that p
art?',
       'I was surprised to see a kangaroo in a World War I epic.',
       'You fell asleep!!', 'There was no kangaroo!',
       'They didn\x92t take any of my suggestions!',
       'That\x92s for coming buddy.', 'I\x92ll see you later.',
       'Don\x92t go!', 'I\x92m sorry.', 'I\x92m so sorry!', 'Look!',
       'This guy fell asleep!', 'He fell asleep too!', 'Be mad at him!',
       'Or, call an ambulance.',
       'Okay, look, I think we have to tell Rachel she messed up her desser
t.',
       'What?! What is with everybody? It\x92s Thanksgiving, not...Truth-Da
```

Preprocessing dataset

In [9]:
```
import re
```

In [10]:
```
def preprocess(sentence):
    sentence = sentence.lower()
    sentence = re.sub(r"\x92", "'",sentence)
    sentence = re.sub(r'[0-9\.]+', '',sentence)
    sentence = re.sub(r'\-',' ',sentence)
    sentence = re.sub(r' +', ' ',sentence)
    sentence = re.sub(r'[?!,.]','',sentence)
    sentence = re.sub(r"y'know",'you know',sentence)
    sentence = re.sub(r"\'t", " not", sentence)
    sentence = re.sub(r"\'re", " are", sentence)
    sentence = re.sub(r"\'s", " is", sentence)
    sentence = re.sub(r"\'d", " would", sentence)
    sentence = re.sub(r"\'ll", " will", sentence)
    sentence = re.sub(r"\'t", " not", sentence)
    sentence = re.sub(r"\'ve", " have", sentence)
    sentence = re.sub(r"\'m", " am", sentence)
    return sentence
```

I have not removed the stop words from data because many of the neutral utterance contains stop word only.

In [11]:
```
from tqdm import notebook
```

preprocess train

In [12]:
```
new_train=[]
for i in notebook.tqdm(train_data.Utterance.values):
  new_train.append(preprocess(i))
train_data['P_Utterance']=new_train
```

```
HBox(children=(FloatProgress(value=0.0, max=9989.0), HTML(value='')))
```

In [13]:
```
train_data['P_Utterance'].values[:100]
```

Out[13]:
```
array(['also i was the point person on my company is transition from the kl
to gr system',
       'you must have had your hands full', 'that i did that i did',
       'so let is talk a little bit about your duties',
       'my duties all right',
       'now you will be heading a whole division so you will have a lot of d
uties',
       'i see',
       'but there will be perhaps people under you so you can dump a certain
amount on them',
       'good to know', 'we can go into detail', 'no don not i beg of you',
       'all right then we will have a definite answer for you on monday but
i think i can say with some confidence you will fit in well here',
       'really', 'absolutely you can relax',
       'but then who the waitress i went out with last month',
       'you know forget it',
       'no no no no no who who were you talking about',
       'no i i i i don not i actually don not know', 'ok',
       'all right well', 'yeah sure', 'hey mon',
       'hey hey hey you wanna hear something that sucks', 'do i ever',
       'chris says they are closing down the bar', 'no way',
       'yeah apparently they are turning it into some kinda coffee place',
       'just coffee where are we gonna hang out now', 'got me',
       'can i get a beer', 'hey did you pick a roommate', 'you betcha',
       'is it the italian guy', 'um mm yeah right',
       'oh my god oh my god poor monica', 'what what what', 'what',
       'he was with her when he wrote this poem',
       'look  amy vessel so empty with nothing inside',
       "now that i have touched you you seem emptier still'",
       'he thinks monica is empty she is the empty vase',
       'oh totally oh god oh she seemed so happy too', 'done', 'hey',
       'hi', 'what are you doing here',
       'ah you know this building is on my paper route so i', 'oh', 'hi',
       'hi', 'how would did it go',
       'oh well the woman i interviewed with was pretty tough but you know t
hank god mark coached me because once i started talking about the fall line
she got all happy and wouldn not shut up',
       'i am so proud of you', 'me too',
       'listen i am ah i am sorry i have been so crazy and jealous and it is
just that i like you a lot so',
       'i know', 'yeah', 'yeah', 'ameri can', 'ameri ccan',
       'ameri can you know it is a', 'everybody',
       'good job joe well done top notch',
       'you liked it you really liked it', 'oh ho ho yeah',
       'which part exactly', 'the whole thing can we go',
       'oh no no no give me some specifics',
       'i love the specifics the specifics were the best part',
       'hey what about the scene with the kangaroo did did you like that par
t',
       'i was surprised to see a kangaroo in a world war i epic',
       'you fell asleep', 'there was no kangaroo',
       'they didn not take any of my suggestions',
       'that is for coming buddy', 'i will see you later', 'don not go',
       'i am sorry', 'i am so sorry', 'look', 'this guy fell asleep',
       'he fell asleep too', 'be mad at him', 'or call an ambulance',
       'okay look i think we have to tell rachel she messed up her dessert',
       'what what is with everybody it is thanksgiving nottruth day',
       'yes and it is my dying wish to have that ring',
       'see if i am not buried with that ring then my spirit is going to wan
der the nether world for all eternity',
       'okay that is enough honey', 'i don not know let me see the ring',
```

preprocess cv

In [14]:
```
new_cv=[]
for i in notebook.tqdm(cv_data.Utterance.values):
  new_cv.append(preprocess(i))
cv_data['P_Utterance']=new_cv
```

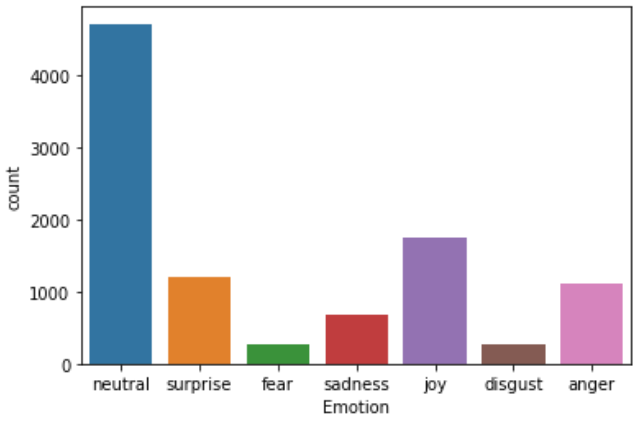
```
HBox(children=(FloatProgress(value=0.0, max=1109.0), HTML(value='')))
```

preprocess test

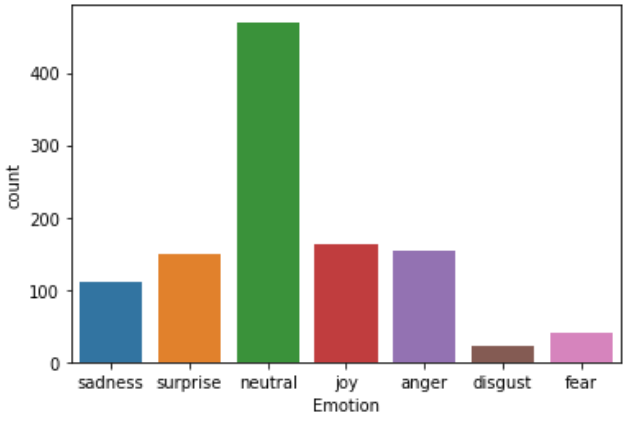
In [15]:
```
new_test=[]
for i in notebook.tqdm(test_data.Utterance.values):
  new_test.append(preprocess(i))
test_data['P_Utterance']=new_test
```

```
HBox(children=(FloatProgress(value=0.0, max=2610.0), HTML(value='')))
```

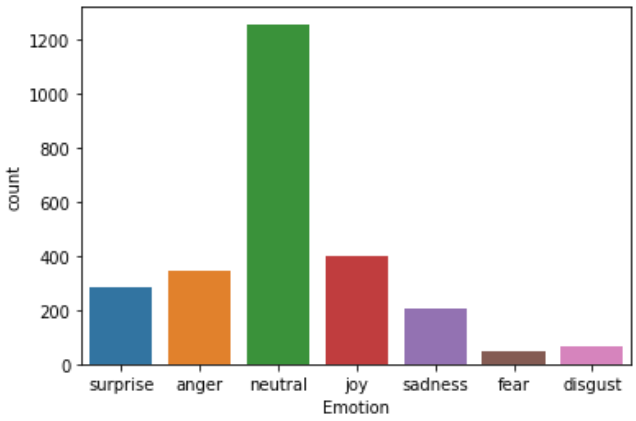
In [16]:
```
sn = sns.countplot(x=train_data['Emotion'], data=train_data)
```



In [17]:
```
sn = sns.countplot(x=cv_data['Emotion'], data=cv_data)
```



In [18]:
```
sn = sns.countplot(x=test_data['Emotion'], data=test_data)
```



In all the plots we can see that the data is highly imbalanced with most classes are neutral

In [19]:
```
s_len=[]
for i in train_data['P_Utterance'].values:
  word = i.split()
  s_len.append(len(word))
```

In [20]:
```
max(s_len)
```

Out[20]: 72

Maximum length of utterance is 72

In [21]:
```
min(s_len)
```

Out[21]: 0

Minimum length of utterance is 0

In [22]:
```
sum(s_len)/9989
```

Out[22]: 8.581940134147562

Avg length of utterance is 8

In [23]:
```
count = 0
for i in s_len :
    if i > 50 :
        count = count + 1
```

In [24]:
```
count
```

Out[24]: 1

Here we can see that only 1 utterence length is more then 50 so remove it.

In [25]:
```
s_len=[]
c=0
X_new = train_data
for i in train_data['P_Utterance']:
  word = i.split()
  c +=1
  if(len(word)==0 or len(word)>50):
    X_new=X_new.drop(X_new.index[c])
```

In [26]:
```
train_data.shape[0] - X_new.shape[0]
```

Out[26]: 4

There are 4 utterence with 0 or more then 50 length and we have removed them

In [27]:
```
X_new = X_new.reset_index()
```

In [28]:
```
X_new
```

Out[28]:

	index	Sr No.	Utterance	Speaker	Emotion	Sentiment	Dialogue_ID	Utterance_ID	Season	Episode
0	0	1	also I was the point person on my companys tr...	Chandler	neutral	neutral	0	0	8	21
1	1	2	You mustve had your hands full.	The Interviewer	neutral	neutral	0	1	8	21
2	2	3	That I did. That I did.	Chandler	neutral	neutral	0	2	8	21
3	3	4	So lets talk a little bit about your duties.	The Interviewer	neutral	neutral	0	3	8	21
4	4	5	My duties? All right.	Chandler	surprise	positive	0	4	8	21
...	...	...	...	...	...	...	...	...	...	...
9980	9984	10474	You or me?	Chandler	neutral	neutral	1038	13	2	3
9981	9985	10475	I got it. Uh, Joey, women don't have Adam's ap...	Ross	neutral	neutral	1038	14	2	3
9982	9986	10476	You guys are messing with me, right?	Joey	surprise	positive	1038	15	2	3
9983	9987	10477	Yeah.	All	neutral	neutral	1038	16	2	3
9984	9988	10478	That was a good one. For a second there, I was...	Joey	joy	positive	1038	17	2	3

9985 rows × 13 columns

In [29]:
```
X_train = X_new.P_Utterance
Y_train = X_new.Emotion
X_cv = cv_data.P_Utterance
Y_cv = cv_data.Emotion
X_test = test_data.P_Utterance
Y_test = test_data.Emotion
```

Creating the vocabulary of utterance words and padding the sentences

In [31]:
```
t = Tokenizer()
t.fit_on_texts(X_train)
vocab_size = len(t.word_index) + 1 # for index zero we have to add +1
encoded_docs_train = t.texts_to_sequences(X_train)
encoded_docs_test = t.texts_to_sequences(X_test)
encoded_docs_cv = t.texts_to_sequences(X_cv)
max_length = 50
padded_docs_train = pad_sequences(encoded_docs_train, maxlen=max_length, padd
padded_docs_test = pad_sequences(encoded_docs_test, maxlen=max_length, paddin
padded_docs_cv = pad_sequences(encoded_docs_cv, maxlen=max_length, padding='p
```

In [110]:
```
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(t, handle)

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

Dowload glove vectors

In [32]:
```
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5
```

```
--2020-07-11 02:45:29--  https://storage.googleapis.com/kaggle-data-sets/213
609%2F464671%2Fbundle%2Farchive.zip?GoogleAccessId=gcp-kaggle-com@kaggle-161
607.iam.gserviceaccount.com&Expires=1594643593&Signature=lwFPfhbKHiEL%2B0hmL
Cj8zFQOEtcmfuky78mUA4ouZi2w2bjBeqp3lyMXFkO50r00aD%2BefqbsR1j4BAAhxlfde47Bscr
I0cY1iweA4m7j5H72Jto5oqKmzpOaMKC%2FrSIQ5%2BikNtP%2BBgC2xayo94xhQLgwJUN8XTwD%
2BACPMKuUXWLQMjh1H2orq8Q%2F9cpo2CrVAA4phaksewmn%2FxNGGOUML2E22cU60j6Hy7TwrrO
oZImeiz%2FI3igsfBypX89AWVhT5yyGIFO%2BC1ag1gUQdMoYKdcXrpGeMi2Dn%2BwH21BZSdXbN
Hr8AspWBGXYkM1MY0u19qz7PM0x92HRVet%2BA2YV%2FQ%3D%3D (https://storage.googlea
pis.com/kaggle-data-sets/213609%2F464671%2Fbundle%2Farchive.zip?GoogleAccess
Id=gcp-kaggle-com@kaggle-161607.iam.gserviceaccount.com&Expires=1594643593&S
ignature=lwFPfhbKHiEL%2B0hmLCj8zFQOEtcmfuky78mUA4ouZi2w2bjBeqp3lyMXFkO50r00a
D%2BefqbsR1j4BAAhxlfde47BscrI0cY1iweA4m7j5H72Jto5oqKmzpOaMKC%2FrSIQ5%2BikNt
P%2BBgC2xayo94xhQLgwJUN8XTwD%2BACPMKuUXWLQMjh1H2orq8Q%2F9cpo2CrVAA4phaksewm
n%2FxNGGOUML2E22cU60j6Hy7TwrrOoZImeiz%2FI3igsfBypX89AWVhT5yyGIFO%2BC1ag1gUQd
MoYKdcXrpGeMi2Dn%2BwH21BZSdXbNHr8AspWBGXYkM1MY0u19qz7PM0x92HRVet%2BA2YV%2FQ%
3D%3D)
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.193.12
8, 172.217.204.128, 172.217.203.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.193.12
8|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1928408067 (1.8G) [application/zip]
Saving to: ‘213609_464671_bundle_archive.zip’

213609_464671_bundl 100%[===================>]   1.80G   139MB/s    in 15s

2020-07-11 02:45:44 (120 MB/s) - ‘213609_464671_bundle_archive.zip’ saved [1
928408067/1928408067]
```

In [33]:
```
!unzip /content/213609_464671_bundle_archive.zip
```

```
Archive:  /content/213609_464671_bundle_archive.zip
  inflating: glove.42B.300d.txt
```

In [34]:
```
from numpy import asarray
from tqdm import tqdm_notebook
embeddings_index = dict()
f = open('glove.42B.300d.txt')
for line in tqdm_notebook(f):
  values = line.split()
  word = values[0]
  coefs = asarray(values[1:], dtype='float32')
  embeddings_index[word] = coefs
f.close()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: TqdmDeprecat
ionWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  """

HBox(children=(FloatProgress(value=1.0, bar_style='info', max=1.0), HTML(val
ue='')))
```

In [35]:
```
from numpy import zeros
input_matrix = zeros((vocab_size, 300))
for word, i in tqdm_notebook(t.word_index.items()):
  embedding_vector = embeddings_index.get(word)
  if embedding_vector is not None:
    input_matrix[i] = embedding_vector
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:3: TqdmDeprecat
ionWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  This is separate from the ipykernel package so we can avoid doing imports
until

HBox(children=(FloatProgress(value=0.0, max=5654.0), HTML(value='')))
```

In [113]:
```
with open('meld_inputmatrix.pickle', 'wb') as handle:
    pickle.dump(input_matrix, handle)
```

In [36]:
```
from sklearn import preprocessing
```

Preparing the label data

In [37]:
```
prep = preprocessing.LabelEncoder()
p_y_train = prep.fit_transform(Y_train)
p_y_cv = prep.transform(Y_cv)
p_y_test = prep.transform(Y_test)
```

In [114]:
```
with open('label_encoder.pickle', 'wb') as handle:
    pickle.dump(prep, handle)
```

In [95]:
```
prep.classes
```

Out[95]:
```
array(['anger', 'disgust', 'fear', 'joy', 'neutral', 'sadness',
       'surprise'], dtype=object)
```

In [96]:
```
prep.transform(prep.classes_)
```

Out[96]:
```
array([0, 1, 2, 3, 4, 5, 6])
```

In [100]:
```
label_dict = dict(zip(prep.transform(prep.classes_), prep.classes_))
```

In [38]:
```
p_y_train
```

Out[38]:
```
array([4, 4, 4, ..., 6, 4, 3])
```

In [39]:
```
label_train = np_utils.to_categorical(p_y_train)
label_cv = np_utils.to_categorical(p_y_cv)
label_test = np_utils.to_categorical(p_y_test)
```

Modelling

CNN with Maxpooling,dropout and BatchNormalization

In [292]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(256, 3, activation='relu')(embedded_sequences)
x = MaxPooling1D(2)(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(128, 3, activation='relu')(x)
x = MaxPooling1D(2)(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(64, 1, activation='relu')(x)
x = MaxPooling1D(2)(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
preds = Dense(7, activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model.summary()
```

```
Model: "model_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_33 (InputLayer)        (None, 50)                0         
_________________________________________________________________
embedding_12 (Embedding)     (None, 50, 300)           1696500   
_________________________________________________________________
conv1d_18 (Conv1D)           (None, 48, 256)           230656    
_________________________________________________________________
max_pooling1d_16 (MaxPooling (None, 24, 256)           0         
_________________________________________________________________
dropout_3 (Dropout)          (None, 24, 256)           0         
_________________________________________________________________
batch_normalization_2 (Batch (None, 24, 256)           1024      
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 22, 128)           98432     
_________________________________________________________________
max_pooling1d_17 (MaxPooling (None, 11, 128)           0         
_________________________________________________________________
dropout_4 (Dropout)          (None, 11, 128)           0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 11, 128)           512       
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 11, 64)            8256      
_________________________________________________________________
max_pooling1d_18 (MaxPooling (None, 5, 64)             0         
_________________________________________________________________
dropout_5 (Dropout)          (None, 5, 64)             0         
_________________________________________________________________
batch_normalization_4 (Batch (None, 5, 64)             256       
_________________________________________________________________
flatten_6 (Flatten)          (None, 320)               0         
_________________________________________________________________
dense_11 (Dense)             (None, 64)                20544     
_________________________________________________________________
dense_12 (Dense)             (None, 7)                 455       
```

In [295]:
```
model.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, la
          epochs=10, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.2172 - v
al_loss: 1.4949
Epoch 2/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.2044 - v
al_loss: 1.4876
Epoch 3/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1895 - v
al_loss: 1.4885
Epoch 4/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1953 - v
al_loss: 1.4773
Epoch 5/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1613 - v
al_loss: 1.5079
Epoch 6/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1362 - v
al_loss: 1.5143
Epoch 7/10
9985/9985 [==============================] - 19s 2ms/step - loss: 1.1321 - v
al_loss: 1.5004
Epoch 8/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1004 - v
al_loss: 1.5436
Epoch 9/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.1016 - v
al_loss: 1.5148
Epoch 10/10
9985/9985 [==============================] - 18s 2ms/step - loss: 1.0936 - v
al_loss: 1.5424
```

Out[295]:
```
<keras.callbacks.callbacks.History at 0x7f65a35bf710>
```

Modelling without Maxpooling

In [299]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(256, 3, activation='relu')(embedded_sequences)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(128, 3, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(64, 1, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
preds = Dense(7, activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model.summary()
```

```
Model: "model_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_36 (InputLayer)        (None, 50)                0
_________________________________________________________________
embedding_15 (Embedding)     (None, 50, 300)           1696500
_________________________________________________________________
conv1d_30 (Conv1D)           (None, 48, 256)           230656
_________________________________________________________________
dropout_14 (Dropout)         (None, 48, 256)           0
_________________________________________________________________
batch_normalization_13 (Batc (None, 48, 256)           1024
_________________________________________________________________
conv1d_31 (Conv1D)           (None, 46, 128)           98432
_________________________________________________________________
dropout_15 (Dropout)         (None, 46, 128)           0
_________________________________________________________________
batch_normalization_14 (Batc (None, 46, 128)           512
_________________________________________________________________
conv1d_32 (Conv1D)           (None, 46, 64)            8256
_________________________________________________________________
dropout_16 (Dropout)         (None, 46, 64)            0
_________________________________________________________________
batch_normalization_15 (Batc (None, 46, 64)            256
_________________________________________________________________
flatten_8 (Flatten)          (None, 2944)              0
_________________________________________________________________
dense_16 (Dense)             (None, 64)                188480
_________________________________________________________________
dense_17 (Dense)             (None, 7)                 455
=================================================================
Total params: 2,224,571
Trainable params: 527,175
Non-trainable params: 1,697,396
_________________________________________________________________
```

In [301]:
```
model.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, la
          epochs=10, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.3150 - v
al_loss: 1.4801
Epoch 2/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.2868 - v
al_loss: 1.4970
Epoch 3/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.2831 - v
al_loss: 1.4960
Epoch 4/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.2495 - v
al_loss: 1.4872
Epoch 5/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.2363 - v
al_loss: 1.4629
Epoch 6/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.2105 - v
al_loss: 1.5160
Epoch 7/10
9985/9985 [==============================] - 27s 3ms/step - loss: 1.2006 - v
al_loss: 1.5049
Epoch 8/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.1712 - v
al_loss: 1.5182
Epoch 9/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.1479 - v
al_loss: 1.5351
Epoch 10/10
9985/9985 [==============================] - 28s 3ms/step - loss: 1.1213 - v
al_loss: 1.5360
```

Out[301]:
```
<keras.callbacks.callbacks.History at 0x7f65a293d080>
```

In [115]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x1 = Conv1D(32, 2)(embedded_sequences)
x2 = Conv1D(32, 2)(embedded_sequences)
x3 = Conv1D(32, 2)(embedded_sequences)
con = Concatenate()([x1,x2,x3])
pool1 = MaxPooling1D(2)(con)

x4 = Conv1D(16, 2)(pool1)
x5 = Conv1D(16, 2)(pool1)
x6 = Conv1D(16, 2)(pool1)
con = Concatenate()([x4,x5,x6])
pool2 = MaxPooling1D(2)(con)
x7 = Conv1D(32, 3)(pool2)
x = Flatten()(x7)
drop=Dropout(0.3)(x)
d1 = Dense(128,activation='relu')(drop)
d1 = Dropout(0.5)(d1)
preds = Dense(7,activation='softmax')(d1)
model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model.summary()
```

```
Model: "model_7"
__________________________________________________________________________________
_____________________
Layer (type)                 Output Shape              Param #     Connected t
o
==================================================================================
=====================
input_7 (InputLayer)         (None, 50)                0
__________________________________________________________________________________
_____________________
embedding_7 (Embedding)      (None, 50, 300)           1696500     input_7
[0][0]
__________________________________________________________________________________
_____________________
conv1d_19 (Conv1D)           (None, 49, 32)            19232       embedding_7
[0][0]
__________________________________________________________________________________
_____________________
conv1d_20 (Conv1D)           (None, 49, 32)            19232       embedding_7
[0][0]
__________________________________________________________________________________
_____________________
conv1d_21 (Conv1D)           (None, 49, 32)            19232       embedding_7
[0][0]
__________________________________________________________________________________
_____________________
concatenate_1 (Concatenate)  (None, 49, 96)            0           conv1d_19
[0][0]
                                                                   conv1d_20
[0][0]
                                                                   conv1d_21
[0][0]
__________________________________________________________________________________
_____________________
max_pooling1d_1 (MaxPooling1D)  (None, 24, 96)         0           concatenate
_1[0][0]
__________________________________________________________________________________
```

In [116]:
```
model.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, la
          epochs=20, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/20
9985/9985 [==============================] - 8s 757us/step - loss: 1.6003 -
val_loss: 1.6149
Epoch 2/20
9985/9985 [==============================] - 7s 712us/step - loss: 1.4939 -
val_loss: 1.5285
Epoch 3/20
9985/9985 [==============================] - 7s 720us/step - loss: 1.4033 -
val_loss: 1.4894
Epoch 4/20
9985/9985 [==============================] - 7s 721us/step - loss: 1.3403 -
val_loss: 1.4851
Epoch 5/20
9985/9985 [==============================] - 7s 728us/step - loss: 1.2564 -
val_loss: 1.4866
Epoch 6/20
9985/9985 [==============================] - 7s 731us/step - loss: 1.1888 -
val_loss: 1.5771
Epoch 7/20
9985/9985 [==============================] - 7s 733us/step - loss: 1.1350 -
val_loss: 1.6175
Epoch 8/20
9985/9985 [==============================] - 7s 746us/step - loss: 1.0483 -
val_loss: 1.6592
Epoch 9/20
9985/9985 [==============================] - 8s 753us/step - loss: 1.0061 -
val_loss: 1.7359
Epoch 10/20
9985/9985 [==============================] - 7s 741us/step - loss: 0.9099 -
val_loss: 1.8415
Epoch 11/20
9985/9985 [==============================] - 8s 755us/step - loss: 0.8848 -
val_loss: 2.0056
Epoch 12/20
9985/9985 [==============================] - 8s 755us/step - loss: 0.7800 -
val_loss: 2.4465
Epoch 13/20
9985/9985 [==============================] - 8s 756us/step - loss: 0.8196 -
val_loss: 2.2329
Epoch 14/20
9985/9985 [==============================] - 7s 749us/step - loss: 0.6747 -
val_loss: 2.2496
Epoch 15/20
9985/9985 [==============================] - 7s 747us/step - loss: 0.7780 -
val_loss: 2.4984
Epoch 16/20
9985/9985 [==============================] - 8s 751us/step - loss: 0.6146 -
val_loss: 2.4381
Epoch 17/20
9985/9985 [==============================] - 7s 749us/step - loss: 0.6435 -
val_loss: 2.8111
Epoch 18/20
9985/9985 [==============================] - 8s 753us/step - loss: 0.9283 -
val_loss: 2.3854
Epoch 19/20
9985/9985 [==============================] - 7s 750us/step - loss: 0.6320 -
val_loss: 2.6725
Epoch 20/20
9985/9985 [==============================] - 8s 752us/step - loss: 0.5487 -
val_loss: 2.7484
```

In [ ]:

CNN with LSTM

In [123]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(256, 3, activation='relu')(embedded_sequences)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LSTM(128, dropout=0.3, return_sequences=True)(x)
x = Dropout(0.5)(x)
x = Conv1D(128, 3, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LSTM(32, dropout=0.3, return_sequences=True)(x)
x = Dropout(0.5)(x)
x = Conv1D(64, 1, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
preds = Dense(7, activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model.summary()
```

```
Model: "model_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_39 (InputLayer)        (None, 50)                0
_________________________________________________________________
embedding_22 (Embedding)     (None, 50, 300)           1696500
_________________________________________________________________
conv1d_68 (Conv1D)           (None, 48, 256)           230656
_________________________________________________________________
dropout_55 (Dropout)         (None, 48, 256)           0
_________________________________________________________________
batch_normalization_19 (Batc (None, 48, 256)           1024
_________________________________________________________________
lstm_26 (LSTM)               (None, 48, 128)           197120
_________________________________________________________________
dropout_56 (Dropout)         (None, 48, 128)           0
_________________________________________________________________
conv1d_69 (Conv1D)           (None, 46, 128)           49280
_________________________________________________________________
dropout_57 (Dropout)         (None, 46, 128)           0
_________________________________________________________________
batch_normalization_20 (Batc (None, 46, 128)           512
_________________________________________________________________
lstm_27 (LSTM)               (None, 46, 32)            20608
_________________________________________________________________
dropout_58 (Dropout)         (None, 46, 32)            0
_________________________________________________________________
conv1d_70 (Conv1D)           (None, 46, 64)            2112
_________________________________________________________________
dropout_59 (Dropout)         (None, 46, 64)            0
_________________________________________________________________
batch_normalization_21 (Batc (None, 46, 64)            256
_________________________________________________________________
flatten_14 (Flatten)         (None, 2944)              0
_________________________________________________________________
dense_41 (Dense)             (None, 64)                188480
```

In [124]:
```
model.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, la
          epochs=20, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/20
9985/9985 [==============================] - 56s 6ms/step - loss: 1.7342 - v
al_loss: 1.6451
Epoch 2/20
9985/9985 [==============================] - 55s 6ms/step - loss: 1.5701 - v
al_loss: 1.6273
Epoch 3/20
9985/9985 [==============================] - 55s 5ms/step - loss: 1.5419 - v
al_loss: 1.6607
Epoch 4/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.5342 - v
al_loss: 1.6271
Epoch 5/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.5206 - v
al_loss: 1.5881
Epoch 6/20
9985/9985 [==============================] - 55s 5ms/step - loss: 1.5096 - v
al_loss: 1.5639
Epoch 7/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.4990 - v
al_loss: 1.5856
Epoch 8/20
9985/9985 [==============================] - 55s 5ms/step - loss: 1.4737 - v
al_loss: 1.5672
Epoch 9/20
9985/9985 [==============================] - 55s 5ms/step - loss: 1.4605 - v
al_loss: 1.6432
Epoch 10/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.4442 - v
al_loss: 1.6382
Epoch 11/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.4281 - v
al_loss: 1.5665
Epoch 12/20
9985/9985 [==============================] - 55s 5ms/step - loss: 1.4167 - v
al_loss: 1.5343
Epoch 13/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.4072 - v
al_loss: 1.6854
Epoch 14/20
9985/9985 [==============================] - 55s 6ms/step - loss: 1.3879 - v
al_loss: 1.5660
Epoch 15/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.3794 - v
al_loss: 1.5408
Epoch 16/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.3811 - v
al_loss: 1.5150
Epoch 17/20
9985/9985 [==============================] - 55s 6ms/step - loss: 1.3666 - v
al_loss: 1.5980
Epoch 18/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.3572 - v
al_loss: 1.5114
Epoch 19/20
9985/9985 [==============================] - 54s 5ms/step - loss: 1.3391 - v
al_loss: 1.5630
Epoch 20/20
9985/9985 [==============================] - 55s 6ms/step - loss: 1.3272 - v
al_loss: 1.5469
```

In [ ]:

Best model CNN with LSTM

In [52]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(256, 3, activation='relu')(embedded_sequences)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LSTM(64, dropout=0.3, return_sequences=True)(x)
x = Conv1D(128, 3, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(64, 1, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
preds = Dense(7, activation='softmax')(x)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model.summary()
```

```
Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 50)                0
_________________________________________________________________
embedding_4 (Embedding)      (None, 50, 300)           1696500
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 48, 256)           230656
_________________________________________________________________
dropout_10 (Dropout)         (None, 48, 256)           0
_________________________________________________________________
batch_normalization_10 (Batc (None, 48, 256)           1024
_________________________________________________________________
lstm_4 (LSTM)                (None, 48, 64)            82176
_________________________________________________________________
conv1d_11 (Conv1D)           (None, 46, 128)           24704
_________________________________________________________________
dropout_11 (Dropout)         (None, 46, 128)           0
_________________________________________________________________
batch_normalization_11 (Batc (None, 46, 128)           512
_________________________________________________________________
conv1d_12 (Conv1D)           (None, 46, 64)            8256
_________________________________________________________________
dropout_12 (Dropout)         (None, 46, 64)            0
_________________________________________________________________
batch_normalization_12 (Batc (None, 46, 64)            256
_________________________________________________________________
flatten_4 (Flatten)          (None, 2944)              0
_________________________________________________________________
dense_7 (Dense)              (None, 64)                188480
_________________________________________________________________
dense_8 (Dense)              (None, 7)                 455
=================================================================
Total params: 2,233,019
Trainable params: 535,623
Non-trainable params: 1,697,396
_________________________________________________________________
```

In [53]:
```
from keras.callbacks import ModelCheckpoint
filepath="weights_email.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_be
callbacks_list = [checkpoint]

model.fit(padded_docs_train, label_train, callbacks=callbacks_list,validation
          epochs=30, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/30
9985/9985 [==============================] - 32s 3ms/step - loss: 1.6706 - v
al_loss: 1.6369

Epoch 00001: val_loss improved from inf to 1.63686, saving model to weights_
email.h5
Epoch 2/30
9985/9985 [==============================] - 31s 3ms/step - loss: 1.5580 - v
al_loss: 1.6090

Epoch 00002: val_loss improved from 1.63686 to 1.60896, saving model to weig
hts_email.h5
Epoch 3/30
9985/9985 [==============================] - 31s 3ms/step - loss: 1.5377 - v
al_loss: 1.5980

Epoch 00003: val_loss improved from 1.60896 to 1.59798, saving model to weig
hts_email.h5
Epoch 4/30
```

Modelling with lstms

In [333]:
```
from keras.models import Sequential
model = Sequential()
model.add(Embedding(vocab_size, 300,weights=[input_matrix],trainable=False))
model.add(LSTM(64, dropout=0.3, return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(32, dropout=0.3, return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(16, dropout=0.3, ))
model.add(Dropout(0.5))

model.add(Dense(100, activation='relu'))
model.add(Dropout(0.5))


model.add(Dense(7,activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
model.summary()
```

```
Model: "sequential_21"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_35 (Embedding)     (None, None, 300)         1696500
_________________________________________________________________
lstm_44 (LSTM)               (None, None, 64)          93440
_________________________________________________________________
dropout_44 (Dropout)         (None, None, 64)          0
_________________________________________________________________
lstm_45 (LSTM)               (None, None, 32)          12416
_________________________________________________________________
dropout_45 (Dropout)         (None, None, 32)          0
_________________________________________________________________
lstm_46 (LSTM)               (None, 16)                3136
_________________________________________________________________
dropout_46 (Dropout)         (None, 16)                0
_________________________________________________________________
dense_51 (Dense)             (None, 100)               1700
_________________________________________________________________
dropout_47 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_52 (Dense)             (None, 7)                 707
=================================================================
Total params: 1,807,899
Trainable params: 111,399
Non-trainable params: 1,696,500
_________________________________________________________________
```

In [334]:
```
model.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, la
          epochs=10, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/10
9985/9985 [==============================] - 26s 3ms/step - loss: 1.6401 - v
al_loss: 1.6335
Epoch 2/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5692 - v
al_loss: 1.6298
Epoch 3/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5642 - v
al_loss: 1.6284
Epoch 4/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5558 - v
al_loss: 1.6266
Epoch 5/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5443 - v
al_loss: 1.6056
Epoch 6/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5300 - v
al_loss: 1.5930
Epoch 7/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5157 - v
al_loss: 1.5861
Epoch 8/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.5079 - v
al_loss: 1.5780
Epoch 9/10
9985/9985 [==============================] - 24s 2ms/step - loss: 1.4963 - v
al_loss: 1.5620
Epoch 10/10
9985/9985 [==============================] - 25s 2ms/step - loss: 1.4856 - v
al_loss: 1.5573
```

Out[334]:
```
<keras.callbacks.callbacks.History at 0x7f659cbf7a90>
```

Lstm with less no of parameters

In [47]:
```
from keras.models import Sequential
model1 = Sequential()
model1.add(Embedding(vocab_size, 300,weights=[input_matrix],trainable=False))
model1.add(LSTM(16, dropout=0.5, return_sequences=True))
model1.add(Dropout(0.5))
model1.add(LSTM(8, dropout=0.5))
model1.add(Dropout(0.5))



model1.add(Dense(100, activation='relu'))
model1.add(Dropout(0.5))

model1.add(Dense(7,activation='softmax'))
model1.compile(loss='categorical_crossentropy', optimizer='adam')
model1.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_6 (Embedding)      (None, None, 300)         1696500
_________________________________________________________________
lstm_15 (LSTM)               (None, None, 16)          20288
_________________________________________________________________
dropout_20 (Dropout)         (None, None, 16)          0
_________________________________________________________________
lstm_16 (LSTM)               (None, 8)                 800
_________________________________________________________________
dropout_21 (Dropout)         (None, 8)                 0
_________________________________________________________________
dense_11 (Dense)             (None, 100)               900
_________________________________________________________________
dropout_22 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_12 (Dense)             (None, 7)                 707
=================================================================
Total params: 1,719,195
Trainable params: 22,695
Non-trainable params: 1,696,500
_________________________________________________________________
```

In [49]:
```
model1.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, l
          epochs=10, batch_size=32)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.5039 - v
al_loss: 1.5919
Epoch 2/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.5009 - v
al_loss: 1.5771
Epoch 3/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4944 - v
al_loss: 1.5781
Epoch 4/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4898 - v
al_loss: 1.6065
Epoch 5/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4857 - v
al_loss: 1.5532
Epoch 6/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4837 - v
al_loss: 1.5537
Epoch 7/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4844 - v
al_loss: 1.5452
Epoch 8/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4803 - v
al_loss: 1.5496
Epoch 9/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4738 - v
al_loss: 1.5673
Epoch 10/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.4756 - v
al_loss: 1.5438
```

Out[49]:
```
<keras.callbacks.callbacks.History at 0x7f7941147438>
```

Bidirectional Lstm

In [50]:
```
from keras.models import Sequential
model1 = Sequential()
model1.add(Embedding(vocab_size, 300,weights=[input_matrix],trainable=False))
model1.add(Bidirectional(LSTM(16, dropout=0.5, return_sequences=True)))
model1.add(Dropout(0.5))
model1.add(LSTM(8, dropout=0.5))
model1.add(Dropout(0.5))



model1.add(Dense(50, activation='relu'))
model1.add(Dropout(0.5))

model1.add(Dense(7,activation='softmax'))
model1.compile(loss='categorical_crossentropy', optimizer='adam')
model1.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, None, 300)         1696500
_________________________________________________________________
bidirectional_2 (Bidirection (None, None, 32)          40576
_________________________________________________________________
dropout_23 (Dropout)         (None, None, 32)          0
_________________________________________________________________
lstm_18 (LSTM)               (None, 8)                 1312
_________________________________________________________________
dropout_24 (Dropout)         (None, 8)                 0
_________________________________________________________________
dense_13 (Dense)             (None, 50)                450
_________________________________________________________________
dropout_25 (Dropout)         (None, 50)                0
_________________________________________________________________
dense_14 (Dense)             (None, 7)                 357
=================================================================
Total params: 1,739,195
Trainable params: 42,695
Non-trainable params: 1,696,500
_________________________________________________________________
```

In [51]:
```
model1.fit(padded_docs_train, label_train, validation_data=(padded_docs_cv, l
          epochs=10, batch_size=64)
```

```
Train on 9985 samples, validate on 1109 samples
Epoch 1/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.6850 - v
al_loss: 1.6397
Epoch 2/10
9985/9985 [==============================] - 13s 1ms/step - loss: 1.5817 - v
al_loss: 1.6250
Epoch 3/10
9985/9985 [==============================] - 13s 1ms/step - loss: 1.5535 - v
al_loss: 1.6058
Epoch 4/10
9985/9985 [==============================] - 13s 1ms/step - loss: 1.5386 - v
al_loss: 1.6053
Epoch 5/10
9985/9985 [==============================] - 12s 1ms/step - loss: 1.5273 - v
al_loss: 1.5907
Epoch 6/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.5212 - v
al_loss: 1.5817
Epoch 7/10
9985/9985 [==============================] - 14s 1ms/step - loss: 1.5194 - v
al_loss: 1.5880
Epoch 8/10
9985/9985 [==============================] - 12s 1ms/step - loss: 1.5117 - v
al_loss: 1.5779
Epoch 9/10
9985/9985 [==============================] - 13s 1ms/step - loss: 1.5083 - v
al_loss: 1.5669
Epoch 10/10
9985/9985 [==============================] - 12s 1ms/step - loss: 1.5045 - v
al_loss: 1.5800
```

Out[51]:
```
<keras.callbacks.callbacks.History at 0x7f79410c71d0>
```

In [ ]:
```

```

Testing with best model

In [55]:
```
sequence_input = Input(shape=(50,))
embedding_layer = Embedding(vocab_size, 300,weights=[input_matrix],trainable=
embedded_sequences = embedding_layer(sequence_input)
x = Conv1D(256, 3, activation='relu')(embedded_sequences)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LSTM(64, dropout=0.3, return_sequences=True)(x)
x = Conv1D(128, 3, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Conv1D(64, 1, activation='relu')(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
preds = Dense(7, activation='softmax')(x)

model1 = Model(sequence_input, preds)
model1.load_weights("/content/weights_email.h5")
model1.compile(loss='categorical_crossentropy',
              optimizer='adam',
              )
model1.summary()
```

```
Model: "model_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         (None, 50)                0
_________________________________________________________________
embedding_6 (Embedding)      (None, 50, 300)           1696500
_________________________________________________________________
conv1d_16 (Conv1D)           (None, 48, 256)           230656
_________________________________________________________________
dropout_16 (Dropout)         (None, 48, 256)           0
_________________________________________________________________
batch_normalization_16 (Batc (None, 48, 256)           1024
_________________________________________________________________
lstm_6 (LSTM)                (None, 48, 64)            82176
_________________________________________________________________
conv1d_17 (Conv1D)           (None, 46, 128)           24704
_________________________________________________________________
dropout_17 (Dropout)         (None, 46, 128)           0
_________________________________________________________________
batch_normalization_17 (Batc (None, 46, 128)           512
_________________________________________________________________
conv1d_18 (Conv1D)           (None, 46, 64)            8256
_________________________________________________________________
dropout_18 (Dropout)         (None, 46, 64)            0
_________________________________________________________________
batch_normalization_18 (Batc (None, 46, 64)            256
_________________________________________________________________
flatten_6 (Flatten)          (None, 2944)              0
_________________________________________________________________
dense_11 (Dense)             (None, 64)                188480
_________________________________________________________________
dense_12 (Dense)             (None, 7)                 455
=================================================================
Total params: 2,233,019
Trainable params: 535,623
Non-trainable params: 1,697,396
_________________________________________________________________
```

In [83]:
```
X = padded_docs_test
Y = label_test
scores = model1.evaluate(X, Y, verbose=0)
```

score we got after testing on best model

In [84]:
```
scores
```

Out[84]:
```
1.408076105446651
```

In [104]:
```
a=model1.predict(X[:5]).argmax(axis=-1)
```

In [109]:
```
for i in a:
  print(label_dict[i])
```

```
neutral
neutral
neutral
neutral
neutral
```