Using ANN to show whether a movie review has a positive or a negative sentiment.

**NOTE:** All output is shown in blue letters or input images

Python code:

import numpy as np

import pandas as pd

*# Input data files are available in the "../input/" directory.*

*# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory*

import os

print(os.listdir('C:/Users/x/Desktop/ML3rdProject/submission\_ann.csv’))

*# Any results you write to the current directory are saved as output.*

['train.tsv', 'test.tsv', 'sampleSubmission.csv']

#Importing data Data

In [2]:

import warnings

warnings.filterwarnings('ignore')

import numpy as np

import pandas as pd

train = pd.read\_csv('C:/Users/x/Desktop/ML3rdProject/submission\_ann.csv /train.tsv',sep = '**\t**')

test = pd.read\_csv(‘C:/Users/x/Desktop/ML3rdProject/submission\_ann.csv /test.tsv', sep = '**\t**')

print("Train set: **{0}**".format(train.shape))

print("Test set: **{0}**".format(test.shape))

df = pd.concat([train, test])

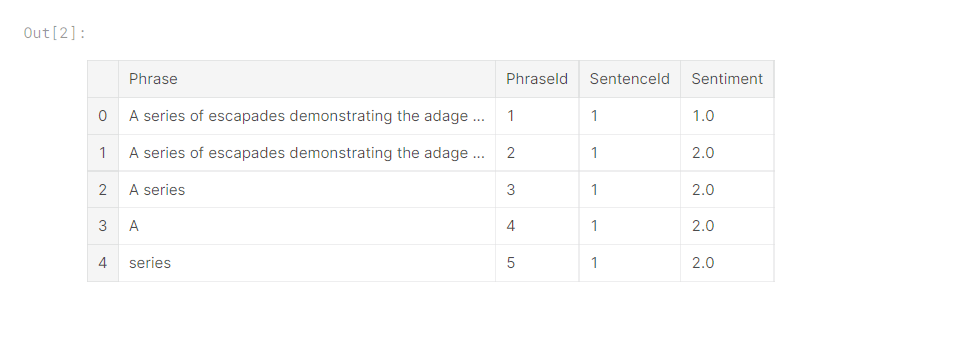
print("All df set: **{0}**".format(df.shape))

df.head()

Train set: (156060, 4)

Test set: (66292, 3)

All df set: (222352, 4)

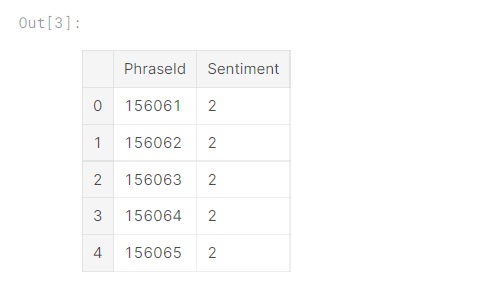
In [3]:

sub = pd.read\_csv('C:/Users/x/Desktop/ML3rdProject/submission\_ann.csv/sampleSubmission.csv', sep = ',')

print("Submission: **{0}**".format(sub.shape))

sub.head()

Submission: (66292, 2)



In [4]:

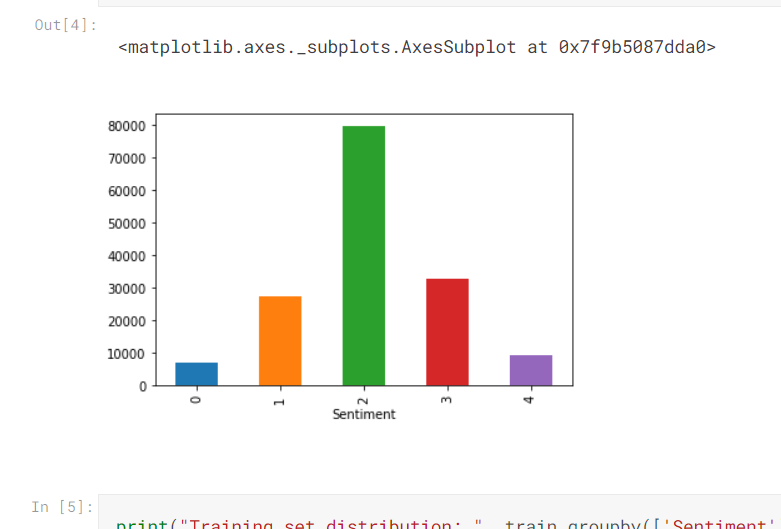
x = train.groupby(['Sentiment'])['PhraseId'].count()

x.plot.bar()

Out[4]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f9b5087dda0>

#Now we analyze the distribution of different groups



In [5]:

print("Training set distribution: ", train.groupby(['Sentiment']).size()/train.shape[0])

Training set distribution: Sentiment

0 0.045316

1 0.174760

2 0.509945

3 0.210989

4 0.058990

dtype: float64

#Cleaning data

In [6]:

import re

from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

from nltk.stem import WordNetLemmatizer

wordnet\_lemmatizer = WordNetLemmatizer()

In [7]:

def clean\_text(text):

text = text.lower()

text = re.sub(r"[-()**\"**#/@;:<>**{}**+=~|.?,]", "", text)

review\_lemma=[]

for word **in** text.split():

word\_lemma = wordnet\_lemmatizer.lemmatize(word)

review\_lemma.append(word\_lemma)

review\_lemma=' '.join(review\_lemma)

return review\_lemma

In [8]:

train['clean\_phrase'] = train['Phrase'].apply(clean\_text)

test['clean\_phrase'] = test['Phrase'].apply(clean\_text)

df['clean\_phrase'] = df['Phrase'].apply(clean\_text)

In [9]:

train.head()



#Counting different features from reviews

In [10]:

from keras.utils import to\_categorical

from sklearn.model\_selection import train\_test\_split

from nltk.tokenize import word\_tokenize

from nltk import FreqDist

Using TensorFlow backend.

In [11]:

train\_text=train.clean\_phrase.values

test\_text=test.clean\_phrase.values

target=train.Sentiment.values

y=to\_categorical(target)

print(train\_text.shape,target.shape,y.shape)

(156060,) (156060,) (156060, 5)

In [12]:

X\_train\_text,X\_val\_text,y\_train,y\_val=train\_test\_split(train\_text,y,test\_size=0.2,stratify=y,random\_state=123)

print(X\_train\_text.shape,y\_train.shape)

print(X\_val\_text.shape,y\_val.shape)

(124848,) (124848, 5)

(31212,) (31212, 5)

In [13]:

all\_words = ' '.join(X\_train\_text)

word2count = {}

for word **in** all\_words.split():

if word **not** **in** word2count:

word2count[word] = 1

else:

word2count[word] += 1

print("Number of unique words: ", len(word2count.keys()))

Number of unique words: 15070

In [14]:

df['length\_review'] = df['clean\_phrase'].apply(lambda x: len(x.split()))

print("Max phrase length: ", max(df['length\_review']))

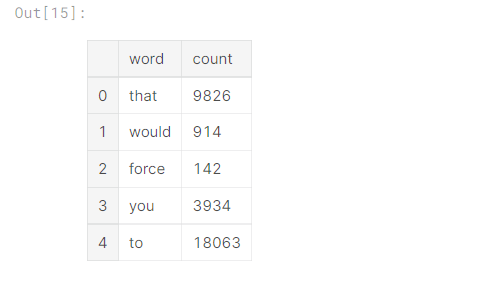
Max phrase length: 52

#Processing the common features of the reviews

In [15]:

d = pd.DataFrame(list(word2count.items()), columns=['word', 'count'])

d.head()



In [16]:

all\_phrases = [X\_train\_text]

all\_phrases

Out[16]:

[array(['that would force you to give it a millisecond of thought',

'joyful', 'might enjoy this', ..., 'c walsh', 'a while a meander',

"think you 've figured out late marriage"], dtype=object)]

In [17]:

*from sklearn.feature\_extraction.text import TfidfTransformer*

*sklearn\_tfidf = TfidfTransformer()*

*sklearn\_representation = sklearn\_tfidf.fit\_transform(all\_phrases)*

#Tokenizer and Sequence padding - Extracting and applying padding to sequences of tokens to use to train deep learning models with Tensorflow.

In [18]:

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad\_sequences

In [19]:

MAX\_REVIEW\_LENGTH = 49

FEATURE\_LENGTH = 12011

BATCH\_SIZE = 1000

EPOCHS = 100

NUM\_CLASSES = 5

In [20]:

tokenizer = Tokenizer(num\_words = FEATURE\_LENGTH)

tokenizer.fit\_on\_texts(list(np.concatenate((train\_text, test\_text), axis=0)))

X\_train = tokenizer.texts\_to\_sequences(X\_train\_text)

X\_val = tokenizer.texts\_to\_sequences(X\_val\_text)

X\_test = tokenizer.texts\_to\_sequences(test\_text)

In [21]:

X\_train = pad\_sequences(X\_train, maxlen=MAX\_REVIEW\_LENGTH)

X\_val = pad\_sequences(X\_val, maxlen=MAX\_REVIEW\_LENGTH)

X\_test= pad\_sequences(X\_test, maxlen=MAX\_REVIEW\_LENGTH)

#Using random Forest

In [22]:

from keras.models import Sequential

from keras.layers import Dense,Dropout,Embedding,LSTM,Conv1D,GlobalMaxPooling1D

from keras.losses import categorical\_crossentropy

from keras.optimizers import Adam

In [23]:

model=Sequential()

model.add(Embedding(FEATURE\_LENGTH,250,mask\_zero=True))

model.add(LSTM(128,dropout=0.4, recurrent\_dropout=0.4,return\_sequences=True))

model.add(LSTM(64,dropout=0.5, recurrent\_dropout=0.5,return\_sequences=False))

model.add(Dense(NUM\_CLASSES,activation='softmax'))

model.compile(loss='categorical\_crossentropy',optimizer=Adam(lr=0.001),metrics=['accuracy'])

model.summary()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_1 (Embedding) (None, None, 250) 3002750

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, None, 128) 194048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_2 (LSTM) (None, 64) 49408

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 5) 325

=================================================================

Total params: 3,246,531

Trainable params: 3,246,531

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [24]:

history = model.fit(X\_train, y\_train, validation\_data=(X\_val, y\_val),epochs=EPOCHS, batch\_size=BATCH\_SIZE, verbose=1)

Train on 124848 samples, validate on 31212 samples

Epoch 1/100

124848/124848 [==============================] - 51s 407us/step - loss: 1.2661 - acc: 0.5339 - val\_loss: 0.9983 - val\_acc: 0.6140

Epoch 2/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.9214 - acc: 0.6389 - val\_loss: 0.8908 - val\_acc: 0.6471

Epoch 3/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.8394 - acc: 0.6662 - val\_loss: 0.8719 - val\_acc: 0.6560

Epoch 4/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.8040 - acc: 0.6798 - val\_loss: 0.8653 - val\_acc: 0.6562

Epoch 5/100

124848/124848 [==============================] - 44s 356us/step - loss: 0.7805 - acc: 0.6878 - val\_loss: 0.8628 - val\_acc: 0.6604

Epoch 6/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.7595 - acc: 0.6968 - val\_loss: 0.8617 - val\_acc: 0.6606

Epoch 7/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.7436 - acc: 0.7011 - val\_loss: 0.8586 - val\_acc: 0.6625

Epoch 8/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.7272 - acc: 0.7074 - val\_loss: 0.8615 - val\_acc: 0.6630

Epoch 9/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.7122 - acc: 0.7125 - val\_loss: 0.8667 - val\_acc: 0.6630

Epoch 10/100

124848/124848 [==============================] - 44s 356us/step - loss: 0.7001 - acc: 0.7156 - val\_loss: 0.8672 - val\_acc: 0.6616

Epoch 11/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.6862 - acc: 0.7211 - val\_loss: 0.8703 - val\_acc: 0.6639

Epoch 12/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.6732 - acc: 0.7256 - val\_loss: 0.8705 - val\_acc: 0.6626

Epoch 13/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.6628 - acc: 0.7287 - val\_loss: 0.8764 - val\_acc: 0.6661

Epoch 14/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.6499 - acc: 0.7326 - val\_loss: 0.8828 - val\_acc: 0.6635

Epoch 15/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.6377 - acc: 0.7369 - val\_loss: 0.8921 - val\_acc: 0.6648

Epoch 16/100

124848/124848 [==============================] - 45s 358us/step - loss: 0.6276 - acc: 0.7398 - val\_loss: 0.8948 - val\_acc: 0.6638

Epoch 17/100

124848/124848 [==============================] - 45s 357us/step - loss: 0.6154 - acc: 0.7448 - val\_loss: 0.9075 - val\_acc: 0.6616

Epoch 18/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.6058 - acc: 0.7474 - val\_loss: 0.9106 - val\_acc: 0.6610

Epoch 19/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5955 - acc: 0.7499 - val\_loss: 0.9312 - val\_acc: 0.6612

Epoch 20/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.5868 - acc: 0.7543 - val\_loss: 0.9354 - val\_acc: 0.6615

Epoch 21/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5770 - acc: 0.7561 - val\_loss: 0.9487 - val\_acc: 0.6590

Epoch 22/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5694 - acc: 0.7604 - val\_loss: 0.9635 - val\_acc: 0.6577

Epoch 23/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5608 - acc: 0.7626 - val\_loss: 0.9668 - val\_acc: 0.6608

Epoch 24/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.5532 - acc: 0.7653 - val\_loss: 0.9893 - val\_acc: 0.6574

Epoch 25/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.5464 - acc: 0.7677 - val\_loss: 0.9978 - val\_acc: 0.6538

Epoch 26/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5387 - acc: 0.7717 - val\_loss: 1.0145 - val\_acc: 0.6542

Epoch 27/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5342 - acc: 0.7701 - val\_loss: 1.0291 - val\_acc: 0.6520

Epoch 28/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.5256 - acc: 0.7743 - val\_loss: 1.0432 - val\_acc: 0.6554

Epoch 29/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.5187 - acc: 0.7788 - val\_loss: 1.0571 - val\_acc: 0.6538

Epoch 30/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.5138 - acc: 0.7789 - val\_loss: 1.0665 - val\_acc: 0.6502

Epoch 31/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.5079 - acc: 0.7826 - val\_loss: 1.0888 - val\_acc: 0.6512

Epoch 32/100

124848/124848 [==============================] - 44s 354us/step - loss: 0.5011 - acc: 0.7849 - val\_loss: 1.1098 - val\_acc: 0.6522

Epoch 33/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4983 - acc: 0.7854 - val\_loss: 1.1069 - val\_acc: 0.6498

Epoch 34/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4934 - acc: 0.7878 - val\_loss: 1.1229 - val\_acc: 0.6505

Epoch 35/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4866 - acc: 0.7902 - val\_loss: 1.1407 - val\_acc: 0.6492

Epoch 36/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4833 - acc: 0.7926 - val\_loss: 1.1525 - val\_acc: 0.6463

Epoch 37/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.4791 - acc: 0.7937 - val\_loss: 1.1517 - val\_acc: 0.6467

Epoch 38/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.4734 - acc: 0.7956 - val\_loss: 1.1669 - val\_acc: 0.6484

Epoch 39/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.4692 - acc: 0.7977 - val\_loss: 1.1843 - val\_acc: 0.6463

Epoch 40/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4650 - acc: 0.7996 - val\_loss: 1.1945 - val\_acc: 0.6464

Epoch 41/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4611 - acc: 0.8007 - val\_loss: 1.2126 - val\_acc: 0.6475

Epoch 42/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.4572 - acc: 0.8018 - val\_loss: 1.2169 - val\_acc: 0.6439

Epoch 43/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.4527 - acc: 0.8043 - val\_loss: 1.2461 - val\_acc: 0.6435

Epoch 44/100

124848/124848 [==============================] - 44s 356us/step - loss: 0.4483 - acc: 0.8069 - val\_loss: 1.2594 - val\_acc: 0.6440

Epoch 45/100

124848/124848 [==============================] - 44s 356us/step - loss: 0.4463 - acc: 0.8065 - val\_loss: 1.2630 - val\_acc: 0.6438

Epoch 46/100

124848/124848 [==============================] - 44s 356us/step - loss: 0.4425 - acc: 0.8087 - val\_loss: 1.2657 - val\_acc: 0.6427

Epoch 47/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.4382 - acc: 0.8098 - val\_loss: 1.2981 - val\_acc: 0.6392

Epoch 48/100

124848/124848 [==============================] - 44s 352us/step - loss: 0.4368 - acc: 0.8111 - val\_loss: 1.2981 - val\_acc: 0.6400

Epoch 49/100

124848/124848 [==============================] - 44s 353us/step - loss: 0.4305 - acc: 0.8127 - val\_loss: 1.3089 - val\_acc: 0.6408

Epoch 50/100

124848/124848 [==============================] - 44s 355us/step - loss: 0.4279 - acc: 0.8134 - val\_loss: 1.3255 - val\_acc: 0.6371

Epoch 51/100

108000/124848 [========================>.....] - ETA: 5s - loss: 0.4225 - acc: 0.8178

In [25]:

y\_pred = model.predict\_classes(X\_test)

In [26]:

test["Sentiment"] = y\_pred

In [27]:

test[['PhraseId', 'Sentiment']].to\_csv('submission\_lstm.csv', index = False)

#ANN

In [28]:

from keras.layers import Flatten

In [29]:

ann\_model = Sequential()

ann\_model.add(Embedding(FEATURE\_LENGTH,250, input\_length=MAX\_REVIEW\_LENGTH))

ann\_model.add(Dense(output\_dim = 100, init = 'uniform', activation = 'relu'))

ann\_model.add(Flatten())

ann\_model.add(Dense(output\_dim = 50, activation='tanh'))

ann\_model.add(Dense(output\_dim = 10, activation = 'relu'))

ann\_model.add(Dense(NUM\_CLASSES,activation='softmax'))

ann\_model.compile(optimizer=Adam(lr=0.001), loss = 'categorical\_crossentropy', metrics = ['accuracy'])

ann\_model.summary()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_2 (Embedding) (None, 49, 250) 3002750

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 49, 100) 25100

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 4900) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 50) 245050

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_4 (Dense) (None, 10) 510

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 5) 55

=================================================================

Total params: 3,273,465

Trainable params: 3,273,465

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [30]:

ann\_history = ann\_model.fit(X\_train, y\_train, validation\_data=(X\_val, y\_val), batch\_size = BATCH\_SIZE, epochs = EPOCHS)

Train on 124848 samples, validate on 31212 samples

Epoch 1/100

124848/124848 [==============================] - 4s 28us/step - loss: 1.1862 - acc: 0.5273 - val\_loss: 0.9932 - val\_acc: 0.6050

Epoch 2/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.8747 - acc: 0.6527 - val\_loss: 0.8614 - val\_acc: 0.6557

Epoch 3/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.7659 - acc: 0.6915 - val\_loss: 0.8454 - val\_acc: 0.6611

Epoch 4/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.7104 - acc: 0.7115 - val\_loss: 0.8494 - val\_acc: 0.6607

Epoch 5/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.6694 - acc: 0.7267 - val\_loss: 0.8588 - val\_acc: 0.6535

Epoch 6/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.6367 - acc: 0.7399 - val\_loss: 0.8723 - val\_acc: 0.6589

Epoch 7/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.6099 - acc: 0.7491 - val\_loss: 0.8906 - val\_acc: 0.6588

Epoch 8/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5870 - acc: 0.7568 - val\_loss: 0.9114 - val\_acc: 0.6515

Epoch 9/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5664 - acc: 0.7632 - val\_loss: 0.9333 - val\_acc: 0.6527

Epoch 10/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5496 - acc: 0.7686 - val\_loss: 0.9550 - val\_acc: 0.6484

Epoch 11/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5344 - acc: 0.7744 - val\_loss: 0.9720 - val\_acc: 0.6481

Epoch 12/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5216 - acc: 0.7781 - val\_loss: 0.9933 - val\_acc: 0.6492

Epoch 13/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.5090 - acc: 0.7815 - val\_loss: 1.0172 - val\_acc: 0.6446

Epoch 14/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4975 - acc: 0.7851 - val\_loss: 1.0363 - val\_acc: 0.6448

Epoch 15/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4856 - acc: 0.7896 - val\_loss: 1.0562 - val\_acc: 0.6382

Epoch 16/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4747 - acc: 0.7925 - val\_loss: 1.0768 - val\_acc: 0.6421

Epoch 17/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4666 - acc: 0.7945 - val\_loss: 1.0993 - val\_acc: 0.6368

Epoch 18/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4574 - acc: 0.7978 - val\_loss: 1.1123 - val\_acc: 0.6372

Epoch 19/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4494 - acc: 0.8002 - val\_loss: 1.1399 - val\_acc: 0.6381

Epoch 20/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4411 - acc: 0.8037 - val\_loss: 1.1720 - val\_acc: 0.6377

Epoch 21/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4340 - acc: 0.8051 - val\_loss: 1.1884 - val\_acc: 0.6346

Epoch 22/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4265 - acc: 0.8081 - val\_loss: 1.2148 - val\_acc: 0.6345

Epoch 23/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4189 - acc: 0.8099 - val\_loss: 1.2338 - val\_acc: 0.6349

Epoch 24/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4131 - acc: 0.8123 - val\_loss: 1.2482 - val\_acc: 0.6367

Epoch 25/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.4064 - acc: 0.8149 - val\_loss: 1.2779 - val\_acc: 0.6301

Epoch 26/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3994 - acc: 0.8174 - val\_loss: 1.2907 - val\_acc: 0.6316

Epoch 27/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3937 - acc: 0.8199 - val\_loss: 1.3238 - val\_acc: 0.6354

Epoch 28/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3869 - acc: 0.8218 - val\_loss: 1.3420 - val\_acc: 0.6309

Epoch 29/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3806 - acc: 0.8239 - val\_loss: 1.3702 - val\_acc: 0.6275

Epoch 30/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3748 - acc: 0.8263 - val\_loss: 1.3916 - val\_acc: 0.6309

Epoch 31/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3692 - acc: 0.8288 - val\_loss: 1.4139 - val\_acc: 0.6308

Epoch 32/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3639 - acc: 0.8319 - val\_loss: 1.4357 - val\_acc: 0.6257

Epoch 33/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3580 - acc: 0.8335 - val\_loss: 1.4525 - val\_acc: 0.6269

Epoch 34/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3513 - acc: 0.8365 - val\_loss: 1.4711 - val\_acc: 0.6249

Epoch 35/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3460 - acc: 0.8386 - val\_loss: 1.5060 - val\_acc: 0.6242

Epoch 36/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3402 - acc: 0.8420 - val\_loss: 1.5252 - val\_acc: 0.6246

Epoch 37/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3344 - acc: 0.8446 - val\_loss: 1.5562 - val\_acc: 0.6243

Epoch 38/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3287 - acc: 0.8468 - val\_loss: 1.6023 - val\_acc: 0.6283

Epoch 39/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3228 - acc: 0.8497 - val\_loss: 1.6070 - val\_acc: 0.6252

Epoch 40/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3185 - acc: 0.8519 - val\_loss: 1.6321 - val\_acc: 0.6246

Epoch 41/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3123 - acc: 0.8549 - val\_loss: 1.6629 - val\_acc: 0.6214

Epoch 42/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3064 - acc: 0.8582 - val\_loss: 1.7207 - val\_acc: 0.6188

Epoch 43/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.3000 - acc: 0.8612 - val\_loss: 1.7255 - val\_acc: 0.6243

Epoch 44/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2939 - acc: 0.8637 - val\_loss: 1.7552 - val\_acc: 0.6183

Epoch 45/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2894 - acc: 0.8672 - val\_loss: 1.7896 - val\_acc: 0.6211

Epoch 46/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2829 - acc: 0.8698 - val\_loss: 1.8336 - val\_acc: 0.6216

Epoch 47/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2772 - acc: 0.8728 - val\_loss: 1.8612 - val\_acc: 0.6184

Epoch 48/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2725 - acc: 0.8755 - val\_loss: 1.8939 - val\_acc: 0.6140

Epoch 49/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2665 - acc: 0.8778 - val\_loss: 1.8899 - val\_acc: 0.6186

Epoch 50/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2603 - acc: 0.8821 - val\_loss: 1.9506 - val\_acc: 0.6163

Epoch 51/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2544 - acc: 0.8844 - val\_loss: 1.9620 - val\_acc: 0.6186

Epoch 52/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2493 - acc: 0.8871 - val\_loss: 2.0095 - val\_acc: 0.6172

Epoch 53/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2457 - acc: 0.8884 - val\_loss: 2.0228 - val\_acc: 0.6159

Epoch 54/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2400 - acc: 0.8916 - val\_loss: 2.0603 - val\_acc: 0.6150

Epoch 55/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2337 - acc: 0.8936 - val\_loss: 2.1149 - val\_acc: 0.6133

Epoch 56/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2280 - acc: 0.8960 - val\_loss: 2.1472 - val\_acc: 0.6164

Epoch 57/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2244 - acc: 0.8984 - val\_loss: 2.1537 - val\_acc: 0.6154

Epoch 58/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2192 - acc: 0.9000 - val\_loss: 2.2011 - val\_acc: 0.6151

Epoch 59/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2147 - acc: 0.9025 - val\_loss: 2.2371 - val\_acc: 0.6129

Epoch 60/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2100 - acc: 0.9053 - val\_loss: 2.2626 - val\_acc: 0.6155

Epoch 61/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2061 - acc: 0.9064 - val\_loss: 2.2751 - val\_acc: 0.6137

Epoch 62/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.2022 - acc: 0.9083 - val\_loss: 2.3141 - val\_acc: 0.6115

Epoch 63/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1985 - acc: 0.9094 - val\_loss: 2.3502 - val\_acc: 0.6122

Epoch 64/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1951 - acc: 0.9106 - val\_loss: 2.3863 - val\_acc: 0.6132

Epoch 65/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1910 - acc: 0.9125 - val\_loss: 2.4010 - val\_acc: 0.6118

Epoch 66/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1866 - acc: 0.9145 - val\_loss: 2.4213 - val\_acc: 0.6118

Epoch 67/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1842 - acc: 0.9159 - val\_loss: 2.4621 - val\_acc: 0.6117

Epoch 68/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1820 - acc: 0.9159 - val\_loss: 2.4772 - val\_acc: 0.6110

Epoch 69/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1807 - acc: 0.9171 - val\_loss: 2.4701 - val\_acc: 0.6110

Epoch 70/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1764 - acc: 0.9183 - val\_loss: 2.5149 - val\_acc: 0.6096

Epoch 71/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1729 - acc: 0.9194 - val\_loss: 2.5272 - val\_acc: 0.6133

Epoch 72/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1701 - acc: 0.9209 - val\_loss: 2.5551 - val\_acc: 0.6094

Epoch 73/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1678 - acc: 0.9215 - val\_loss: 2.5875 - val\_acc: 0.6126

Epoch 74/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1643 - acc: 0.9227 - val\_loss: 2.6285 - val\_acc: 0.6082

Epoch 75/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1640 - acc: 0.9232 - val\_loss: 2.5960 - val\_acc: 0.6150

Epoch 76/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1622 - acc: 0.9233 - val\_loss: 2.6503 - val\_acc: 0.6114

Epoch 77/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1588 - acc: 0.9243 - val\_loss: 2.6656 - val\_acc: 0.6111

Epoch 78/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1572 - acc: 0.9249 - val\_loss: 2.6588 - val\_acc: 0.6106

Epoch 79/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1566 - acc: 0.9255 - val\_loss: 2.6932 - val\_acc: 0.6069

Epoch 80/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1553 - acc: 0.9256 - val\_loss: 2.7335 - val\_acc: 0.6091

Epoch 81/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1540 - acc: 0.9264 - val\_loss: 2.7166 - val\_acc: 0.6067

Epoch 82/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1532 - acc: 0.9262 - val\_loss: 2.7212 - val\_acc: 0.6111

Epoch 83/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1515 - acc: 0.9274 - val\_loss: 2.7517 - val\_acc: 0.6112

Epoch 84/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1524 - acc: 0.9266 - val\_loss: 2.7612 - val\_acc: 0.6085

Epoch 85/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1499 - acc: 0.9277 - val\_loss: 2.7705 - val\_acc: 0.6120

Epoch 86/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1488 - acc: 0.9281 - val\_loss: 2.7639 - val\_acc: 0.6100

Epoch 87/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1475 - acc: 0.9279 - val\_loss: 2.8092 - val\_acc: 0.6066

Epoch 88/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1459 - acc: 0.9286 - val\_loss: 2.7996 - val\_acc: 0.6073

Epoch 89/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1437 - acc: 0.9296 - val\_loss: 2.7961 - val\_acc: 0.6089

Epoch 90/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1436 - acc: 0.9293 - val\_loss: 2.8070 - val\_acc: 0.6103

Epoch 91/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1422 - acc: 0.9297 - val\_loss: 2.8439 - val\_acc: 0.6092

Epoch 92/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1412 - acc: 0.9302 - val\_loss: 2.8522 - val\_acc: 0.6065

Epoch 93/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1393 - acc: 0.9303 - val\_loss: 2.8594 - val\_acc: 0.6097

Epoch 94/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1391 - acc: 0.9307 - val\_loss: 2.8803 - val\_acc: 0.6083

Epoch 95/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1388 - acc: 0.9313 - val\_loss: 2.8842 - val\_acc: 0.6071

Epoch 96/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1395 - acc: 0.9300 - val\_loss: 2.8871 - val\_acc: 0.6069

Epoch 97/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1381 - acc: 0.9306 - val\_loss: 2.8858 - val\_acc: 0.6086

Epoch 98/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1373 - acc: 0.9308 - val\_loss: 2.9244 - val\_acc: 0.6075

Epoch 99/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1379 - acc: 0.9312 - val\_loss: 2.9013 - val\_acc: 0.6078

Epoch 100/100

124848/124848 [==============================] - 3s 23us/step - loss: 0.1387 - acc: 0.9308 - val\_loss: 2.8881 - val\_acc: 0.6080

In [31]:

y\_pred = ann\_model.predict\_classes(X\_test)

In [32]:

test["Sentiment"] = y\_pred

test[['PhraseId', 'Sentiment']].to\_csv('submission\_ann.csv', index = False)