In [1]:

```
#import gensim library
import gensim
from gensim.models.doc2vec import LabeledSentence

import numpy as np
import os
import time
import codecs

#parameters
data_dir = 'episodes'# data directory containing input.txt
save_dir = 'episodes' # directory to store models
file_list=["HP1"]
```

In [2]:

```
#import spacy, and french model
import en_core_web_sm


# In[12]:


#import spacy, and french model
import spacy
nlp = en_core_web_sm.load()

#initiate sentences and labels lists
sentences = []
sentences_label = []

#create sentences function:
def create_sentences(doc):
    ponctuation = [".","?","!",":","…"]
    sentences = []
    sent = []
    for word in doc:
        if word.text not in ponctuation:
            if word.text not in ("\n","\n\n",'\u2009','\xa0'):
                sent.append(word.text.lower())
        else:
            sent.append(word.text.lower())
            if len(sent) > 1:
                sentences.append(sent)
            sent=[]
    return sentences

#create sentences from files
for file_name in file_list:
    input_file = os.path.join(data_dir, file_name + ".txt")
    #read data
    with codecs.open(input_file, "r") as f:
        data = f.read()
    #create sentences
    doc = nlp(data)
    sents = create_sentences(doc)
    sentences = sentences + sents

#create labels
for i in range(np.array(sentences).shape[0]):
    sentences_label.append("ID" + str(i))
```

In [3]:

```
class LabeledLineSentence(object):
    def __init__(self, doc_list, labels_list):
        self.labels_list = labels_list
        self.doc_list = doc_list
    def __iter__(self):
        for idx, doc in enumerate(self.doc_list):
            yield gensim.models.doc2vec.LabeledSentence(doc,[self.labels_list[idx]])
```

In [12]:

```
def train_doc2vec_model(data, docLabels, size=300, sample=0.000001, dm=0, hs=1, window=10,
    startime = time.time()

    print("{0} articles loaded for model".format(len(data)))

    it = LabeledLineSentence(data, docLabels)

    model = gensim.models.Doc2Vec(size=size, sample=sample, dm=dm, window=window, min_count
    model.build_vocab(it)
    for epoch in range(epoch):
        print("Training epoch {}".format(epoch + 1))
        model.train(it,total_examples=model.corpus_count,epochs=model.iter)
        # model.alpha -= 0.002 # decrease the learning rate
        # model.min_alpha = model.alpha # fix the learning rate, no decay

    #saving the created model
    model.save(os.path.join(save_file))
    print('model saved')
```

In [14]:

```
train_doc2vec_model(sentences, sentences_label, size=500,sample=0.0,alpha=0.025, min_alpha=
```

```
2600 articles loaded for model
```

```
c:\users\mayan\appdata\local\programs\python\python37\lib\site-packages\gens
im\models\doc2vec.py:574: UserWarning: The parameter `size` is deprecated, w
ill be removed in 4.0.0, use `vector_size` instead.
  warnings.warn("The parameter `size` is deprecated, will be removed in 4.0.
0, use `vector_size` instead.")
c:\users\mayan\appdata\local\programs\python\python37\lib\site-packages\ipyk
ernel_launcher.py:7: DeprecationWarning: Call to deprecated `LabeledSentence
` (Class will be removed in 4.0.0, use TaggedDocument instead).
  import sys
```

```
Training epoch 1
```

```
c:\users\mayan\appdata\local\programs\python\python37\lib\site-packages\ipyk
ernel_launcher.py:12: DeprecationWarning: Call to deprecated `iter` (Attribu
te will be removed in 4.0.0, use self.epochs instead).
  if sys.path[0] == '':
```

```
Training epoch 2
Training epoch 3
Training epoch 4
Training epoch 5
Training epoch 6
Training epoch 7
Training epoch 8
Training epoch 9
Training epoch 10
Training epoch 11
Training epoch 12
Training epoch 13
Training epoch 14
Training epoch 15
Training epoch 16
Training epoch 17
Training epoch 18
Training epoch 19
Training epoch 20
model saved
```

In [15]:

```
#import library
from six.moves import cPickle

#load the model
d2v_model = gensim.models.doc2vec.Doc2Vec.load('models\\doc2vec.w2v')

sentences_vector=[]

t = 500

for i in range(len(sentences)):
    if i % t == 0:
        print("sentence", i, ":", sentences[i])
        print("***")
    sent = sentences[i]
    sentences_vector.append(d2v_model.infer_vector(sent, alpha=0.001, min_alpha=0.001, step

#save the sentences_vector
sentences_vector_file = os.path.join("models", "sentences_vector_500_a001_ma001_s10000.pkl"
with open(os.path.join(sentences_vector_file), 'wb') as f:
    cPickle.dump((sentences_vector), f)
```

```
sentence 0 : ['the', 'sorting', 'hat', 'the', 'door', 'swung', 'open', 'at',
'once', '.']
***
sentence 500 : ['"', 'this', 'was', 'so', 'unfair', 'that', 'harry', 'opene
d', 'his', 'mouth', 'to', 'argue', ',', 'but', 'ron', 'kicked', 'him', 'behi
nd', 'their', 'cauldron', '.']
***
sentence 1000 : ['"', 'ron', 'grinned', 'at', 'harry', '.']
***
sentence 1500 : ['"', 'dunno', 'what', 'harry', 'thinks', 'he', "'s", 'doin
g', ',', '"', 'hagrid', 'mumbled', '.']
***
sentence 2000 : ['the', 'happiest', 'man', 'on', 'earth', 'would', 'be', 'ab
le', 'to', 'use', 'the', 'mirror', 'of', 'erised', 'like', 'a', 'normal', 'm
irror', ',', 'that', 'is', ',', 'he', 'would', 'look', 'into', 'it', 'and',
'see', 'himself', 'exactly', 'as', 'he', 'is', '.']
***
sentence 2500 : ['the', 'clock', 'on', 'the', 'wall', 'had', 'just', 'chime
d', 'midnight', 'when', 'the', 'portrait', 'hole', 'burst', 'open', '.']
***
```

In [4]:

```
from six.moves import cPickle
with open("models\\sentences_vector_500_a001_ma001_s10000.pkl",'rb') as f:
    sentences_vector = cPickle.load(f)
```

In [5]:

```
nb_sequenced_sentences = 15
vector_dim = 500

X_train = np.zeros((len(sentences), nb_sequenced_sentences, vector_dim), dtype=np.float)
y_train = np.zeros((len(sentences), vector_dim), dtype=np.float)

t = 1000
for i in range(len(sentences_label)-nb_sequenced_sentences-1):
    if i % t == 0: print("new sequence: ", i)

    for k in range(nb_sequenced_sentences):
        sent = sentences_label[i+k]
        vect = sentences_vector[i+k]

        if i % t == 0:
            print("  ", k + 1 ,"th vector for this sequence. Sentence ", sent, "(vector dim

        for j in range(len(vect)):
            X_train[i, k, j] = vect[j]

    senty = sentences_label[i+nb_sequenced_sentences]
    vecty = sentences_vector[i+nb_sequenced_sentences]
    if i % t == 0: print("  y vector for this sequence ", senty, ": (vector dim = ", len(ve
    for j in range(len(vecty)):
        y_train[i, j] = vecty[j]

print(X_train.shape, y_train.shape)
```

```
new sequence:  0
   1 th vector for this sequence. Sentence  ID0 (vector dim =  500 )
   2 th vector for this sequence. Sentence  ID1 (vector dim =  500 )
   3 th vector for this sequence. Sentence  ID2 (vector dim =  500 )
   4 th vector for this sequence. Sentence  ID3 (vector dim =  500 )
   5 th vector for this sequence. Sentence  ID4 (vector dim =  500 )
   6 th vector for this sequence. Sentence  ID5 (vector dim =  500 )
   7 th vector for this sequence. Sentence  ID6 (vector dim =  500 )
   8 th vector for this sequence. Sentence  ID7 (vector dim =  500 )
   9 th vector for this sequence. Sentence  ID8 (vector dim =  500 )
   10 th vector for this sequence. Sentence  ID9 (vector dim =  500 )
   11 th vector for this sequence. Sentence  ID10 (vector dim =  500 )
   12 th vector for this sequence. Sentence  ID11 (vector dim =  500 )
   13 th vector for this sequence. Sentence  ID12 (vector dim =  500 )
   14 th vector for this sequence. Sentence  ID13 (vector dim =  500 )
   15 th vector for this sequence. Sentence  ID14 (vector dim =  500 )
  y vector for this sequence  ID15 : (vector dim =  500 )
new sequence:  1000
   1 th vector for this sequence. Sentence  ID1000 (vector dim =  500 )
   2 th vector for this sequence. Sentence  ID1001 (vector dim =  500 )
   3 th vector for this sequence. Sentence  ID1002 (vector dim =  500 )
   4 th vector for this sequence. Sentence  ID1003 (vector dim =  500 )
   5 th vector for this sequence. Sentence  ID1004 (vector dim =  500 )
   6 th vector for this sequence. Sentence  ID1005 (vector dim =  500 )
   7 th vector for this sequence. Sentence  ID1006 (vector dim =  500 )
   8 th vector for this sequence. Sentence  ID1007 (vector dim =  500 )
   9 th vector for this sequence. Sentence  ID1008 (vector dim =  500 )
   10 th vector for this sequence. Sentence  ID1009 (vector dim =  500 )
   11 th vector for this sequence. Sentence  ID1010 (vector dim =  500 )
   12 th vector for this sequence. Sentence  ID1011 (vector dim =  500 )
   13 th vector for this sequence. Sentence  ID1012 (vector dim =  500 )
   14 th vector for this sequence. Sentence  ID1013 (vector dim =  500 )
   15 th vector for this sequence. Sentence  ID1014 (vector dim =  500 )
  y vector for this sequence  ID1015 : (vector dim =  500 )
new sequence:  2000
   1 th vector for this sequence. Sentence  ID2000 (vector dim =  500 )
   2 th vector for this sequence. Sentence  ID2001 (vector dim =  500 )
   3 th vector for this sequence. Sentence  ID2002 (vector dim =  500 )
   4 th vector for this sequence. Sentence  ID2003 (vector dim =  500 )
   5 th vector for this sequence. Sentence  ID2004 (vector dim =  500 )
   6 th vector for this sequence. Sentence  ID2005 (vector dim =  500 )
   7 th vector for this sequence. Sentence  ID2006 (vector dim =  500 )
   8 th vector for this sequence. Sentence  ID2007 (vector dim =  500 )
   9 th vector for this sequence. Sentence  ID2008 (vector dim =  500 )
   10 th vector for this sequence. Sentence  ID2009 (vector dim =  500 )
   11 th vector for this sequence. Sentence  ID2010 (vector dim =  500 )
   12 th vector for this sequence. Sentence  ID2011 (vector dim =  500 )
   13 th vector for this sequence. Sentence  ID2012 (vector dim =  500 )
   14 th vector for this sequence. Sentence  ID2013 (vector dim =  500 )
   15 th vector for this sequence. Sentence  ID2014 (vector dim =  500 )
  y vector for this sequence  ID2015 : (vector dim =  500 )
(2600, 15, 500) (2600, 500)
```

In [6]:

```
from keras import regularizers
from keras.models import Sequential, Model
from keras.layers import Dense, Activation, Dropout, Embedding, Flatten, Bidirectional, Inp
from keras.callbacks import EarlyStopping,ModelCheckpoint
from keras.optimizers import Adam
from keras.metrics import categorical_accuracy, mean_squared_error, mean_absolute_error, lo
from keras.layers.normalization import BatchNormalization

def bidirectional_lstm_model(seq_length, vector_dim):
    print('Building LSTM model...')
    model = Sequential()
    model.add(Bidirectional(LSTM(rnn_size, activation="relu"),input_shape=(seq_length, vect
    model.add(Dropout(0.5))
    model.add(Dense(vector_dim))

    optimizer = Adam(lr=learning_rate)
    callbacks=[EarlyStopping(patience=2, monitor='val_loss')]
    model.compile(loss='logcosh', optimizer=optimizer, metrics=['acc'])
    print('LSTM model built.')
    return model
```

```
Using TensorFlow backend.
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorflow\py
thon\framework\dtypes.py:525: FutureWarning: Passing (type, 1) or '1type'
as a synonym of type is deprecated; in a future version of numpy, it will
be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type, 1) or
```

```
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\mayan\AppData\Roaming\Python\Python37\site-packages\tensorboard\c
ompat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy,
it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [7]:

```
rnn_size = 512 # size of RNN
vector_dim = 500
learning_rate = 0.0001 #learning rate

model_sequence = bidirectional_lstm_model(nb_sequenced_sentences, vector_dim)
```

```
Building LSTM model...
LSTM model built.
```

In [8]:

```
batch_size = 30 # minibatch size

callbacks=[EarlyStopping(patience=3, monitor='val_loss'),
           ModelCheckpoint(filepath='models\\my_model_sequence_lstm.{epoch:02d}.hdf5',\
                           monitor='val_loss', verbose=1, mode='auto', period=5)]

history = model_sequence.fit(X_train, y_train,
                 batch_size=batch_size,
                 shuffle=True,
                 epochs=40,
                 callbacks=callbacks,
                 validation_split=0.1)

#save the model
model_sequence.save('models\\my_model_sequence_lstm.final2.hdf5')
```

```
WARNING:tensorflow:From c:\users\mayan\appdata\local\programs\python\python3
7\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.glo
bal_variables is deprecated. Please use tf.compat.v1.global_variables instea
d.

Train on 2340 samples, validate on 260 samples
Epoch 1/40
2340/2340 [==============================] - 5s 2ms/step - loss: 0.0622 - ac
c: 0.0256 - val_loss: 0.0531 - val_acc: 0.0731
Epoch 2/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0573 - ac
c: 0.0603 - val_loss: 0.0527 - val_acc: 0.0731
Epoch 3/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0563 - ac
c: 0.0624 - val_loss: 0.0525 - val_acc: 0.0615
Epoch 4/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0557 - ac
c: 0.0637 - val_loss: 0.0524 - val_acc: 0.0577
Epoch 5/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0553 - ac
c: 0.0607 - val_loss: 0.0523 - val_acc: 0.0538

Epoch 00005: saving model to models\my_model_sequence_lstm.05.hdf5
Epoch 6/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0550 - ac
c: 0.0598 - val_loss: 0.0523 - val_acc: 0.0538
Epoch 7/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0547 - ac
c: 0.0573 - val_loss: 0.0523 - val_acc: 0.0538
Epoch 8/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0544 - ac
c: 0.0624 - val_loss: 0.0522 - val_acc: 0.0538
Epoch 9/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0542 - ac
c: 0.0624 - val_loss: 0.0522 - val_acc: 0.0500
Epoch 10/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0540 - ac
c: 0.0620 - val_loss: 0.0522 - val_acc: 0.0500

Epoch 00010: saving model to models\my_model_sequence_lstm.10.hdf5
Epoch 11/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0537 - ac
c: 0.0607 - val_loss: 0.0522 - val_acc: 0.0577
```

```
Epoch 12/40
2340/2340 [==============================] - 4s 2ms/step - loss: 0.0535 - ac
c: 0.0607 - val_loss: 0.0522 - val_acc: 0.0577
```

```
In [ ]:
```