In [1]:
```
from __future__ import print_function

from sklearn.preprocessing import OneHotEncoder
from keras.layers.core import Dense, Activation, Dropout
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM, Merge
from keras.datasets import imdb
import pandas as pd
import numpy as np
import os
```

```
Using TensorFlow backend.
/Users/DuaaTashkandi/anaconda/lib/python3.6/importlib/_bootstrap.py:
205: RuntimeWarning: compiletime version 3.5 of module 'tensorflow.p
ython.framework.fast_tensor_util' does not match runtime version 3.6
  return f(*args, **kwds)
```

In [2]:
```
#parameters
maxlen = 30
labels = 2
```

In [3]:
```
input = pd.read_csv("data_new/merged.csv",header=None)
input.columns = ['first', 'last','b_or_n']

# remove encode
input['first'] = input['first'].str[2:-1]
input['last'] = input['last'].str[2:-1]

input['firstlen']= [len(str(i)) for i in input['first']]
input['lastlen'] = [len(str(i)) for i in input['last']]
input1 = input[(input['firstlen'] >= 2) & (input['lastlen'] >= 2)]
```

In [4]:
```
firsts = input['first']
lasts = input['last']
labels = input['b_or_n']

vocab = set(' '.join([str(i) for i in firsts]))
vocab = set(' '.join([str(i) for i in lasts]))

vocab.add('END')
len_vocab = len(vocab)
```

In [5]:
```
char_index = dict((c, i) for i, c in enumerate(vocab))
```

In [6]:
```
#train test split
msk = np.random.rand(len(input1)) < 0.8

train = input1[msk]
test = input1[~msk]
```

In [7]:
```
def set_flag(i):
    tmp = np.zeros(39);
    tmp[i] = 1
    return(tmp)
```

In [8]:
```
# Truncating and padding training data
train_X = []
train_Y = []
train_Z = []

trunc_train_first = [str(i)[0:maxlen] for i in train['first']]
trunc_train_last = [str(i)[0:maxlen] for i in train['last']]

for i in trunc_train_first:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    train_X.append(tmp)

for i in trunc_train_last:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    train_Y.append(tmp)

for i in train['b_or_n']:
    if i == 1:
        train_Z.append([1,0])
    else:
        train_Z.append([0,1])
```

In [9]:
```
np.asarray(train_X).shape
```

Out[9]:
```
(48013, 30, 39)
```

In [10]:
```
np.asarray(train_Y).shape
```

Out[10]:
```
(48013, 30, 39)
```

```
In [11]: np.asarray(train_Z).shape

Out[11]: (48013, 2)
```

```
In [12]: #build the model: 2 stacked LSTM
         print('Building LSTM model')

         left_branch = Sequential()
         left_branch.add(LSTM(512, return_sequences=True, input_shape=(maxlen,l
         en_vocab)))
         right_branch = Sequential()
         right_branch.add(LSTM(512, return_sequences=True, input_shape=(maxlen,
         len_vocab)))

         model = Sequential()
         model.add(Merge([left_branch, right_branch], mode='concat'))
         model.add(Dropout(0.2))
         model.add(LSTM(512, return_sequences=False))
         model.add(Dropout(0.2))
         model.add(Dense(2))

         # Softmax activation function
         model.add(Activation('softmax'))

         # Cross-entropy loss, metric is accuracy
         model.compile(loss='categorical_crossentropy', optimizer='adam',metric
         s=['accuracy'])
```

```
Building LSTM model

/Users/DuaaTashkandi/anaconda/lib/python3.6/site-packages/ipykernel_
launcher.py:10: UserWarning: The `Merge` layer is deprecated and wil
l be removed after 08/2017. Use instead layers from `keras.layers.me
rge`, e.g. `add`, `concatenate`, etc.
  # Remove the CWD from sys.path while we load stuff.
```

In [13]:
```
# Truncating and padding test data

test_X = []
test_Y = []
test_Z = []

trunc_test_first = [str(i)[0:maxlen] for i in test['first']]
trunc_test_last = [str(i)[0:maxlen] for i in test['last']]

for i in trunc_test_first:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    test_X.append(tmp)

for i in trunc_test_last:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    test_Y.append(tmp)

for i in test['b_or_n']:
    if i == 1:
        test_Z.append([1,0])
    else:
        test_Z.append([0,1])
```

In [14]:
```
print(np.asarray(test_X).shape)
print(np.asarray(test_Y).shape)
print(np.asarray(test_Z).shape)
```

```
(11941, 30, 39)
(11941, 30, 39)
(11941, 2)
```

In [15]:
```
batch_size = 1000
model.fit([np.asarray(train_X), np.asarray(train_Y)], train_Z, batch_s
ize=batch_size, nb_epoch=50, validation_data=([np.asarray(test_X), np.
asarray(test_Y)], test_Z))
```

```
/Users/DuaaTashkandi/anaconda/lib/python3.6/site-packages/keras/mode
ls.py:939: UserWarning: The `nb_epoch` argument in `fit` has been re
named `epochs`.
  warnings.warn('The `nb_epoch` argument in `fit` '

Train on 48013 samples, validate on 11941 samples
Epoch 1/50
48013/48013 [==============================] - 1474s 31ms/step - los
s: 0.6589 - acc: 0.5925 - val_loss: 0.5259 - val_acc: 0.7562
```

```
Epoch 2/50
48013/48013 [==============================] - 1407s 29ms/step - los
s: 0.4934 - acc: 0.7720 - val_loss: 0.5144 - val_acc: 0.7621
Epoch 3/50
48013/48013 [==============================] - 1473s 31ms/step - los
s: 0.4585 - acc: 0.7941 - val_loss: 0.5232 - val_acc: 0.7586
Epoch 4/50
48013/48013 [==============================] - 1442s 30ms/step - los
s: 0.4429 - acc: 0.8017 - val_loss: 0.4991 - val_acc: 0.7864
Epoch 5/50
48013/48013 [==============================] - 1495s 31ms/step - los
s: 0.4213 - acc: 0.8171 - val_loss: 0.4070 - val_acc: 0.8232
Epoch 6/50
48013/48013 [==============================] - 1493s 31ms/step - los
s: 0.3961 - acc: 0.8288 - val_loss: 0.4094 - val_acc: 0.8248
Epoch 7/50
48013/48013 [==============================] - 1508s 31ms/step - los
s: 0.3776 - acc: 0.8404 - val_loss: 0.5977 - val_acc: 0.7334
Epoch 8/50
48013/48013 [==============================] - 1379s 29ms/step - los
s: 0.3828 - acc: 0.8398 - val_loss: 0.4222 - val_acc: 0.8276
Epoch 9/50
48013/48013 [==============================] - 1381s 29ms/step - los
s: 0.3641 - acc: 0.8478 - val_loss: 0.3531 - val_acc: 0.8554
Epoch 10/50
48013/48013 [==============================] - 1366s 28ms/step - los
s: 0.3330 - acc: 0.8620 - val_loss: 0.3713 - val_acc: 0.8446
Epoch 11/50
48013/48013 [==============================] - 1579s 33ms/step - los
s: 0.3370 - acc: 0.8623 - val_loss: 0.3468 - val_acc: 0.8591
Epoch 12/50
48013/48013 [==============================] - 1666s 35ms/step - los
s: 0.3242 - acc: 0.8682 - val_loss: 0.3330 - val_acc: 0.8610
Epoch 13/50
48013/48013 [==============================] - 1506s 31ms/step - los
s: 0.3162 - acc: 0.8724 - val_loss: 0.3369 - val_acc: 0.8642
Epoch 14/50
48013/48013 [==============================] - 1412s 29ms/step - los
s: 0.3041 - acc: 0.8775 - val_loss: 0.3177 - val_acc: 0.8733
Epoch 15/50
48013/48013 [==============================] - 1393s 29ms/step - los
s: 0.3059 - acc: 0.8768 - val_loss: 0.3424 - val_acc: 0.8572
Epoch 16/50
48013/48013 [==============================] - 1383s 29ms/step - los
s: 0.3067 - acc: 0.8746 - val_loss: 0.3177 - val_acc: 0.8715
Epoch 17/50
48013/48013 [==============================] - 1376s 29ms/step - los
s: 0.2874 - acc: 0.8864 - val_loss: 0.3299 - val_acc: 0.8621
Epoch 18/50
48013/48013 [==============================] - 1450s 30ms/step - los
```

```
s: 0.3203 - acc: 0.8711 - val_loss: 0.3100 - val_acc: 0.8757
Epoch 19/50
48013/48013 [==============================] - 1507s 31ms/step - los
s: 0.2869 - acc: 0.8865 - val_loss: 0.3051 - val_acc: 0.8822
Epoch 20/50
48013/48013 [==============================] - 1385s 29ms/step - los
s: 0.2834 - acc: 0.8887 - val_loss: 0.3066 - val_acc: 0.8776
Epoch 21/50
48013/48013 [==============================] - 1374s 29ms/step - los
s: 0.2846 - acc: 0.8869 - val_loss: 0.2994 - val_acc: 0.8844
Epoch 22/50
48013/48013 [==============================] - 1378s 29ms/step - los
s: 0.2682 - acc: 0.8943 - val_loss: 0.2956 - val_acc: 0.8861
Epoch 23/50
48013/48013 [==============================] - 1366s 28ms/step - los
s: 0.2622 - acc: 0.8983 - val_loss: 0.2951 - val_acc: 0.8843
Epoch 24/50
48013/48013 [==============================] - 1377s 29ms/step - los
s: 0.2549 - acc: 0.9009 - val_loss: 0.2740 - val_acc: 0.8946
Epoch 25/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.2451 - acc: 0.9050 - val_loss: 0.2854 - val_acc: 0.8920
Epoch 26/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.2544 - acc: 0.9025 - val_loss: 0.2812 - val_acc: 0.8916
Epoch 27/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.2365 - acc: 0.9096 - val_loss: 0.2833 - val_acc: 0.8916
Epoch 28/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.2315 - acc: 0.9114 - val_loss: 0.2719 - val_acc: 0.8978
Epoch 29/50
48013/48013 [==============================] - 1378s 29ms/step - los
s: 0.2280 - acc: 0.9130 - val_loss: 0.2908 - val_acc: 0.8881
Epoch 30/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.2575 - acc: 0.8997 - val_loss: 0.2714 - val_acc: 0.8956
Epoch 31/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.2325 - acc: 0.9121 - val_loss: 0.2734 - val_acc: 0.8952
Epoch 32/50
48013/48013 [==============================] - 1369s 29ms/step - los
s: 0.2406 - acc: 0.9098 - val_loss: 0.2907 - val_acc: 0.8893
Epoch 33/50
48013/48013 [==============================] - 1369s 29ms/step - los
s: 0.2200 - acc: 0.9174 - val_loss: 0.2711 - val_acc: 0.8967
Epoch 34/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.2174 - acc: 0.9175 - val_loss: 0.2534 - val_acc: 0.9032
Epoch 35/50
```

```
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.2082 - acc: 0.9226 - val_loss: 0.2710 - val_acc: 0.9029
Epoch 36/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.2007 - acc: 0.9251 - val_loss: 0.2933 - val_acc: 0.8962
Epoch 37/50
48013/48013 [==============================] - 1376s 29ms/step - los
s: 0.2033 - acc: 0.9256 - val_loss: 0.2641 - val_acc: 0.9003
Epoch 38/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.1980 - acc: 0.9271 - val_loss: 0.2706 - val_acc: 0.9018
Epoch 39/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.1824 - acc: 0.9331 - val_loss: 0.2745 - val_acc: 0.8991
Epoch 40/50
48013/48013 [==============================] - 1378s 29ms/step - los
s: 0.1763 - acc: 0.9358 - val_loss: 0.2661 - val_acc: 0.9056
Epoch 41/50
48013/48013 [==============================] - 1380s 29ms/step - los
s: 0.1752 - acc: 0.9357 - val_loss: 0.2678 - val_acc: 0.9034
Epoch 42/50
48013/48013 [==============================] - 1373s 29ms/step - los
s: 0.1765 - acc: 0.9355 - val_loss: 0.2645 - val_acc: 0.9046
Epoch 43/50
48013/48013 [==============================] - 1378s 29ms/step - los
s: 0.1676 - acc: 0.9381 - val_loss: 0.2742 - val_acc: 0.9008
Epoch 44/50
48013/48013 [==============================] - 1406s 29ms/step - los
s: 0.1660 - acc: 0.9396 - val_loss: 0.2794 - val_acc: 0.8977
Epoch 45/50
48013/48013 [==============================] - 1380s 29ms/step - los
s: 0.1781 - acc: 0.9335 - val_loss: 0.3036 - val_acc: 0.8916
Epoch 46/50
48013/48013 [==============================] - 1375s 29ms/step - los
s: 0.2114 - acc: 0.9201 - val_loss: 0.3214 - val_acc: 0.8620
Epoch 47/50
48013/48013 [==============================] - 1383s 29ms/step - los
s: 0.1970 - acc: 0.9245 - val_loss: 0.2628 - val_acc: 0.9011
Epoch 48/50
48013/48013 [==============================] - 1380s 29ms/step - los
s: 0.1753 - acc: 0.9349 - val_loss: 0.2637 - val_acc: 0.9067
Epoch 49/50
48013/48013 [==============================] - 1377s 29ms/step - los
s: 0.1546 - acc: 0.9436 - val_loss: 0.2890 - val_acc: 0.9005
Epoch 50/50
48013/48013 [==============================] - 1381s 29ms/step - los
s: 0.1543 - acc: 0.9425 - val_loss: 0.2924 - val_acc: 0.9005
```

```
Out[15]: <keras.callbacks.History at 0x182ea6588>
```

In [17]:
```
score, acc = model.evaluate([np.asarray(test_X), np.asarray(test_Y)],
test_Z)
print('Test score:', score)
print('Test accuracy:', acc)
```

```
11941/11941 [==============================] - 141s 12ms/step
Test score: 0.292373455935
Test accuracy: 0.900510844993
```

In [18]:
```
#save our model and data
model.save_weights('model', overwrite=True)
train.to_csv("train_split.csv")
test.to_csv("test_split.csv")
```

In [19]:
```
evals = model.predict([np.asarray(test_X), np.asarray(test_Y)])
prob_m = [i[0] for i in evals]
```

In [20]:
```
out = pd.DataFrame(prob_m)
out['first'] = test['first'].reset_index()['first']
out['last'] = test['last'].reset_index()['last']
out['b_or_n'] = test['b_or_n'].reset_index()['b_or_n']
```

In [21]:
```
out.head(10)
out.columns = ['prob_b','first', 'last', 'actual']
out.head(10)
out.to_csv("pred_out.csv")
```

In [44]:
```
# small test
first = ["tequila","wjdan","alvaro","duaa","michel","luiz","baraa"]
last = ["amigo","alharthi","lima","tashkandi","caetano","guilherme","k
oshak"]
X = []
Y = []
trunc_first = [i[0:maxlen] for i in first]
trunc_last = [i[0:maxlen] for i in last]
for i in trunc_first:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    X.append(tmp)
for i in trunc_last:
    tmp = [set_flag(char_index[j]) for j in str(i)]
    for k in range(0,maxlen - len(str(i))):
        tmp.append(set_flag(char_index["END"]))
    Y.append(tmp)

pred = model.predict([np.asarray(X), np.asarray(Y)])
pred
```

Out[44]:
```
array([[ 0.79103982,  0.20896016],
       [ 0.12539442,  0.8746056 ],
       [ 0.99756986,  0.0024301 ],
       [ 0.22257635,  0.77742368],
       [ 0.85553694,  0.14446311],
       [ 0.99840802,  0.00159197],
       [ 0.01068358,  0.98931646]], dtype=float32)
```