Bidrectional LSTM with pre-trained Twitter Word Embeddings

Cliche used an ensemble of bidirectional LSTMs along with CNNs to produce state of the art results in Twitter sentiment analysis. He trains initial word embeddings on a large, unlabled corpus of Twitter data using a neural language model. We will instead be using Stanford's pre-trained Glove word embeddings that were specifically trained on Twitter data. Since our training data is not very large, we anticipate that using these pre-trained word embeddings will result in an increase in performance.

In [2]:
```
import pandas as pd
import re
import nltk
import string
import os
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from keras.preprocessing.text import Tokenizer
from sklearn.model_selection import train_test_split
import glob, os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Embedding, Dense
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer, TfidfTran
sformer
from sklearn.metrics import confusion_matrix, classification_report, a
ccuracy_score, f1_score
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.text import Tokenizer
```

```
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:519: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:520: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorflow/python/framework/dtypes.py:525: FutureWarni
ng: 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)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:541: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:544: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
```

```
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:545: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/Users/shivaomrani/opt/anaconda3/envs/neural_networks/lib/python3.7/
site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550: Futu
reWarning: Passing (type, 1) or '1type' as a synonym of type is depr
ecated; in a future version of numpy, it will be understood as (type
, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
Using TensorFlow backend.
```

In [3]:

```
os.chdir("data/")
```

Helper methods for reading tweets and cleaning them.

In [43]:

```
def read_tsv(file_path):
    df = pd.read_table(file_path)
    return df

import string
import re

# code inspired from https://www.kaggle.com/rahulvv/bidirectional-lstm
-glove200d


def remove_urls(text):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'',text)

def remove_html(text):
    html=re.compile(r'<.*?>')
    return html.sub(r'',text)

def split_text(text):
    text = text.split()
    return text

def lower(text):
    text = [word.lower() for word in text]
    return str(text)

def remove_punct(text):
    text = ''.join([char for char in text if char not in string.punctu
ation])
    text = re.sub('[0-9]+', '', str(text))
    return text

def remove_stopwords(text):
    pattern = re.compile(r'\b('+r'|'.join(stopwords.words('english'))
+ r')\b\s*')
    text = pattern.sub(' ', text)
    return text

lemmatizer = WordNetLemmatizer()
def lemmatize_words(text):
    text = lemmatizer.lemmatize(text)
    return text

def clean_tweet(text):
    t0 = remove_urls(text)
    t1 = remove_html(t0)
    t2 = split_text(t1)
    t3 = lower(t2)
    t4 = remove_punct(t3)
    t5 = remove_stopwords(t4)
    t6 = lemmatize_words(t5)
    return t6
```

In [44]:

```
tweet_df = pd.DataFrame(columns=['tweet', 'sentiment','NA'])
df_test = pd.DataFrame(columns=['tweet', 'sentiment','NA'])

for file in glob.glob("*.tsv"):
        if 'final_test' in file:
            df_test_cur = read_tsv(file)
            df_test = pd.concat([df_test, df_test_cur])
        else:
            df_train_cur = read_tsv(file)
            tweet_df = pd.concat([tweet_df, df_train_cur])
```

In [45]:
```
print(tweet_df[['tweet', 'sentiment']] )
```

```
                                                   tweet sentiment
0     05 Beat it - Michael Jackson - Thriller (25th ...   neutral
1     Jay Z joins Instagram with nostalgic tribute t...  positive
2     Michael Jackson: Bad 25th Anniversary Edition ...   neutral
3     I liked a @YouTube video http://t.co/AaR3pjp2P...  positive
4     18th anniv of Princess Diana's death. I still ...  positive
...                                                 ...       ...
1137                   Maybe it was - his - fantasy ?  positive
1138  It was ok , but they always just seem so nervo...  negative
1139  It is streamable from YepRoc -- matter of fact...  positive
1140  comment telling me who you are , or how you fo...  positive
1141  im on myspace ... ill try and find you and add...   neutral

[53368 rows x 2 columns]
```

In [46]:
```
print(df_test[['tweet', 'sentiment']] )
```

```
                                                   tweet sentiment
0      #ArianaGrande Ari By Ariana Grande 80% Full ht...   neutral
1      Ariana Grande KIIS FM Yours Truly CD listening...  positive
2      Ariana Grande White House Easter Egg Roll in W...  positive
3      #CD #Musics Ariana Grande Sweet Like Candy 3.4...  positive
4      SIDE TO SIDE 😘 @arianagrande #sidetoside #aria...   neutral
...                                                  ...       ...
11901  @dansen17 update: Zac Efron kissing a puppy ht...  positive
11902  #zac efron sex pic skins michelle sex https://...   neutral
11903  First Look at Neighbors 2 with Zac Efron Shirt...   neutral
11904  zac efron poses nude #lovely libra porn https:...   neutral
11905  #Fashion #Style The Paperboy (NEW Blu-ray Disc...   neutral

[11906 rows x 2 columns]
```

Reading Glove word embeddings into a dictionary.

In [47]:
```
#preparing train lables
tweet_df.loc[tweet_df.sentiment == "positive", "sentiment"] = 2
tweet_df.loc[tweet_df.sentiment == "neutral", "sentiment"] = 1
tweet_df.loc[tweet_df.sentiment == "negative", "sentiment"] = 0

labels = tweet_df["sentiment"].tolist()
labels = [ int(x) for x in labels ]

#preparing test labels
df_test.loc[df_test.sentiment == "positive", "sentiment"] = 2
df_test.loc[df_test.sentiment == "neutral", "sentiment"] = 1
df_test.loc[df_test.sentiment == "negative", "sentiment"] = 0

labels_test = df_test["sentiment"].tolist()
labels_test = [ int(x) for x in labels_test ]
```

Converting tweets and labels into lists.

In [48]:
```
train_tweets = tweet_df.tweet.values
y_train_orig = tweet_df.sentiment.values
test_tweets = df_test.tweet.values
```

In [49]:
```
from keras.utils import to_categorical

train_labels = to_categorical(y_train_orig)

clean_training_tweets = []
for i in range(len(train_tweets)):
    data = clean_tweet(train_tweets[i])
    clean_training_tweets.append(data)

clean_testing_tweets = []
for i in range(len(test_tweets)):
    data = clean_tweet(test_tweets[i])
    clean_testing_tweets.append(data)
```

Checking the tweets after cleaning them.

In [50]:
```
print(clean_training_tweets[:10])
print(clean_testing_tweets[:10])
```

```
[' beat  michael jackson  thriller th anniversary edition hd', 'jay
z joins instagram  nostalgic tribute  michael jackson jay z apparent
ly joined instagram  saturday  ', 'michael jackson bad th anniversar
y edition picture vinyl  unique picture disc vinyl includes  origina
l ', ' liked  youtube video one direction singing man   mirror  mich
ael jackson  atlanta ga june ', 'th anniv  princess dianas death  st
ill want  believe   living   private island away   public  michael j
ackson', 'oridaganjazz  st time  heard michael jackson sing   honolu
lu hawaii   restaurant  radio   abc    loved  ', 'michael jackson ap
peared  saturday    th place   top  miamis trends trndnl', '  old en
ough  remember michael jackson attending  grammys  brooke shields  w
ebster sat   lap   show', 'etbowser  u enjoy  nd rate michael jackso
n bit honest ques like  cant feel face song  god   obvious  want mj
', ' weeknd   closest thing may get  michael jackson   long timeesp
ecially since  damn near mimics everything']
['arianagrande ari  ariana grande  full singer actress', 'ariana gra
nde kiis fm  truly cd listening party  burbank arianagrande', 'arian
a grande white house easter egg roll  washington arianagrande', 'cd
musics ariana grande sweet like candy  oz  ml sealed  box  authenic
new', 'side  side 😘 arianagrande sidetoside arianagrande musically
comunidadgay lgbt🌈 lotb…', 'hairspray live previews   macys thanksg
iving day parade arianagrande televisionnbc', 'lindsaylohan ‘feelin
g thankful’  blasting arianagrande  wearing ‘toomuch…', ' hate   lo
ve  songs dammit arianagrande', 'ariana grande 【right  ft big sean】
アリアナ arianagrande', ' one would  prefer  listen   whole day 😍🤘🏼
could never choose arianagrande intoyou sidetoside songs poll']
```

In [20]:
```
print('Loading word vectors...')
word2vec = {}
with open(os.path.join('../glove/glove.twitter.27B.200d.txt'), encodin
g = "utf-8") as f:
    for line in f:
        values = line.split()
        word = values[0]
        vec = np.asarray(values[1:], dtype='float32')
        word2vec[word] = vec
print('Found %s word vectors.' % len(word2vec))
```

```
Loading word vectors...
Found 1193514 word vectors.
```

In [52]:

```
# converting tweets to integer sequences
tokenizer = Tokenizer(num_words= 20000, oov_token= 'OOV')
tokenizer.fit_on_texts(clean_training_tweets)
train_tweet_sequences = tokenizer.texts_to_sequences(clean_training_tweets)
word_index_train = tokenizer.word_index
print('Found %s unique words in train tweets.' % len(word_index_train))
X_train = pad_sequences(sequences=train_tweet_sequences, maxlen=32, padding= 'post', truncating='post')


test_tweet_sequences = tokenizer.texts_to_sequences(clean_testing_tweets)
X_test = pad_sequences(sequences= test_tweet_sequences, maxlen=32, padding='post', truncating='post')
```

```
Found 67101 unique words in train tweets.
```

In [53]:

```
print('Shape of X train tensor: ', X_train.shape)
print('Shape of X test: ', X_test.shape)
```

```
Shape of X train tensor:  (53368, 32)
Shape of X test:  (11906, 32)
```

In [54]:

```
num_words = min(20000, len(word_index_train)+1)
embedding_matrix = np.zeros((num_words, 200))

embeddings = []
for word, i in word_index_train.items():
    if i<20000:
        embeddings = word2vec.get(word)
        if embeddings is not None:
            embedding_matrix[i] = embeddings
```

In [17]:

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Embedding(input_dim=num_words,output_dim = 200, weights=[embedding_matrix], input_length=32,trainable=False))
model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(100, return_sequences=True)))
model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32, return_sequences=True)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.Adam(lr=0.01), metrics=['accuracy'])
```

```
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/keras/initia
lizers.py:119: calling RandomUniform.__init__ (from tensorflow.pytho
n.ops.init_ops) with dtype is deprecated and will be removed in a fu
ture version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/ops/init_ops
.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.o
ps.init_ops) with dtype is deprecated and will be removed in a futur
e version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/ops/init_ops
.py:97: calling GlorotUniform.__init__ (from tensorflow.python.ops.i
nit_ops) with dtype is deprecated and will be removed in a future ve
rsion.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/ops/init_ops
.py:97: calling Orthogonal.__init__ (from tensorflow.python.ops.init
_ops) with dtype is deprecated and will be removed in a future versi
on.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/ops/init_ops
.py:97: calling Zeros.__init__ (from tensorflow.python.ops.init_ops)
with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
```

In [18]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding (Embedding)        (None, 32, 200)           4000000   
_________________________________________________________________
bidirectional (Bidirectional (None, 32, 200)           240800    
_________________________________________________________________
bidirectional_1 (Bidirection (None, 32, 64)            59648     
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0         
_________________________________________________________________
dense (Dense)                (None, 3)                 6147      
=================================================================
Total params: 4,306,595
Trainable params: 306,595
Non-trainable params: 4,000,000
_________________________________________________________________
```

In [19]:
```
history=model.fit(X_train, train_labels, batch_size=128, epochs=15)
```

```
WARNING:tensorflow:From /Users/shivaomrani/opt/anaconda3/envs/neural
_networks/lib/python3.7/site-packages/tensorflow/python/ops/math_gra
d.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.py
thon.ops.array_ops) is deprecated and will be removed in a future ve
rsion.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Epoch 1/15
53368/53368 [==============================] - 103s 2ms/sample - los
s: 0.7818 - acc: 0.6416
Epoch 2/15
53368/53368 [==============================] - 108s 2ms/sample - los
s: 0.6944 - acc: 0.6876
Epoch 3/15
53368/53368 [==============================] - 100s 2ms/sample - los
s: 0.6321 - acc: 0.7211
Epoch 4/15
53368/53368 [==============================] - 99s 2ms/sample - loss
: 0.5661 - acc: 0.7535
Epoch 5/15
53368/53368 [==============================] - 95s 2ms/sample - loss
: 0.4978 - acc: 0.7854
Epoch 6/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.4374 - acc: 0.8170
Epoch 7/15
```

```
53368/53368 [==============================] - 91s 2ms/sample - loss
: 0.3940 - acc: 0.8358
Epoch 8/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.3537 - acc: 0.8544
Epoch 9/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.3260 - acc: 0.8683
Epoch 10/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.3010 - acc: 0.8809
Epoch 11/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.2785 - acc: 0.8896
Epoch 12/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.2758 - acc: 0.8918
Epoch 13/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.2516 - acc: 0.9013
Epoch 14/15
53368/53368 [==============================] - 92s 2ms/sample - loss
: 0.2503 - acc: 0.9038
Epoch 15/15
53368/53368 [==============================] - 94s 2ms/sample - loss
: 0.2383 - acc: 0.9092
```

In [20]:

```
pred_p = model.predict(X_test)
```

In [21]:

```
pred = (np.round(pred_p)).astype(int)
final_pred = []
for sample in pred:
    pred_label = sample.argmax()
    final_pred.append(pred_label)
```

In [22]:

```
y_binary = to_categorical(labels_test)
model.evaluate(x = X_test, y =y_binary )
```

```
11906/11906 [==============================] - 10s 830us/sample - lo
ss: 1.6071 - acc: 0.5879
```

Out[22]:

```
[1.6070550479709491, 0.58793885]
```

In [23]:
```
from sklearn.metrics import classification_report
print(classification_report(labels_test, final_pred))
```

```
              precision    recall  f1-score   support

           0       0.58      0.61      0.59      3811
           1       0.62      0.58      0.60      5743
           2       0.51      0.56      0.53      2352

    accuracy                           0.58     11906
   macro avg       0.57      0.58      0.58     11906
weighted avg       0.59      0.58      0.58     11906
```

In [24]:
```
# Calling `save('my_model')` creates a SavedModel folder `my_model`.
model.save("bidirectional-lstm")
```

In [56]:
```
# It can be used to reconstruct the model identically.
reconstructed_model = keras.models.load_model("bidirectional-lstm")
y_binary = to_categorical(labels_test)
reconstructed_model.evaluate(x = X_test, y =y_binary)
```

```
11906/11906 [==============================] - 13s 1ms/sample - loss
: 1.6071 - acc: 0.5879
```

Out[56]:
```
[1.6070550479709491, 0.58793885]
```

In [60]:
```
from sklearn.metrics import classification_report
pred_p = reconstructed_model.predict(X_test)

pred = (np.round(pred_p)).astype(int)
final_pred = []
for sample in pred:
    pred_label = sample.argmax()
    final_pred.append(pred_label)

print(classification_report(labels_test, final_pred))
```

```
              precision    recall  f1-score   support

           0       0.58      0.61      0.59      3811
           1       0.62      0.58      0.60      5743
           2       0.51      0.56      0.53      2352

    accuracy                           0.58     11906
   macro avg       0.57      0.58      0.58     11906
weighted avg       0.59      0.58      0.58     11906
```

In [ ]: