In [2]:

```
import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

In [3]:

```
import numpy as np
import pandas as pd

dataset = pd.read_csv('Sarcasm_Headlines_Dataset.csv')

sentences = dataset['headline'].tolist()
labels = dataset['is_sarcastic'].tolist()

# Separate out the sentences and labels into training and test sets
training_size = int(len(sentences) * 0.8)

training_sentences = sentences[0:training_size]
testing_sentences = sentences[training_size:]

training_labels = labels[0:training_size]
testing_labels = labels[training_size:]

# Make labels into numpy arrays for use with the network later
training_labels_final = np.array(training_labels)
testing_labels_final = np.array(testing_labels)
```

In [4]:

```
vocab_size = 10000
embedding_dim = 16
max_length = 32
trunc_type='post'
padding_type='post'
oov_tok = "<OOV>"


from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

tokenizer = Tokenizer(num_words = vocab_size, oov_token=oov_tok)
tokenizer.fit_on_texts(training_sentences)
word_index = tokenizer.word_index
sequences = tokenizer.texts_to_sequences(training_sentences)
padded = pad_sequences(sequences,maxlen=max_length, padding=padding_type,
                       truncating=trunc_type)

testing_sequences = tokenizer.texts_to_sequences(testing_sentences)
testing_padded = pad_sequences(testing_sequences,maxlen=max_length,
                               padding=padding_type, truncating=trunc_type)
```

In [5]:

```
padded
```

Out[5]:

```
array([[ 320,    1,  681, ...,    0,    0,    0],
       [   4, 7191, 2989, ...,    0,    0,    0],
       [ 156,  924,    2, ...,    0,    0,    0],
       ...,
       [1020, 3614,    5, ...,    0,    0,    0],
       [3702,    1,   12, ...,    0,    0,    0],
       [1247, 1017, 1087, ...,    0,    0,    0]])
```

In [6]:

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim, input_length=max_length),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(24, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding (Embedding)       (None, 32, 16)            160000    
                                                                 
 flatten (Flatten)           (None, 512)               0         
                                                                 
 dense (Dense)               (None, 24)                12312     
                                                                 
 dense_1 (Dense)             (None, 1)                 25        
                                                                 
=================================================================
Total params: 172337 (673.19 KB)
Trainable params: 172337 (673.19 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [7]:

```
num_epochs = 100
```

```
history=model.fit(padded, training_labels_final, epochs=num_epochs, validation_data=(testing_padded, testing_la
```

```
Epoch 1/100
668/668 [==============================] - 5s 6ms/step - loss: 0.4244 - accuracy: 0.7876 - val_loss: 0.3137 - v
al_accuracy: 0.8605
Epoch 2/100
668/668 [==============================] - 4s 6ms/step - loss: 0.1867 - accuracy: 0.9261 - val_loss: 0.3213 - v
al_accuracy: 0.8643
Epoch 3/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0852 - accuracy: 0.9711 - val_loss: 0.3843 - v
al_accuracy: 0.8592
Epoch 4/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0312 - accuracy: 0.9919 - val_loss: 0.4673 - v
al_accuracy: 0.8551
Epoch 5/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0108 - accuracy: 0.9980 - val_loss: 0.5396 - v
al_accuracy: 0.8525
Epoch 6/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0046 - accuracy: 0.9991 - val_loss: 0.5981 - v
al_accuracy: 0.8506
Epoch 7/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0026 - accuracy: 0.9993 - val_loss: 0.6435 - v
al_accuracy: 0.8501
Epoch 8/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0018 - accuracy: 0.9994 - val_loss: 0.6874 - v
al_accuracy: 0.8486
Epoch 9/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0017 - accuracy: 0.9996 - val_loss: 0.7469 - v
al_accuracy: 0.8478
Epoch 10/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0015 - accuracy: 0.9995 - val_loss: 0.7635 - v
al_accuracy: 0.8484
Epoch 11/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0011 - accuracy: 0.9997 - val_loss: 0.7996 - v
al_accuracy: 0.8487
Epoch 12/100
668/668 [==============================] - 3s 5ms/step - loss: 7.1350e-04 - accuracy: 0.9998 - val_loss: 0.8495
- val_accuracy: 0.8422
Epoch 13/100
668/668 [==============================] - 3s 5ms/step - loss: 8.7255e-04 - accuracy: 0.9997 - val_loss: 0.8878
- val_accuracy: 0.8433
Epoch 14/100
668/668 [==============================] - 3s 5ms/step - loss: 0.0010 - accuracy: 0.9997 - val_loss: 0.8948 - v
al_accuracy: 0.8446
Epoch 15/100
668/668 [==============================] - 3s 5ms/step - loss: 5.0746e-04 - accuracy: 0.9999 - val_loss: 0.9061
- val_accuracy: 0.8448
Epoch 16/100
668/668 [==============================] - 3s 5ms/step - loss: 6.9976e-04 - accuracy: 0.9998 - val_loss: 0.9802
- val_accuracy: 0.8428
Epoch 17/100
668/668 [==============================] - 3s 5ms/step - loss: 6.4700e-04 - accuracy: 0.9999 - val_loss: 0.9740
- val_accuracy: 0.8401
Epoch 18/100
668/668 [==============================] - 3s 5ms/step - loss: 9.7398e-04 - accuracy: 0.9996 - val_loss: 1.0203
- val_accuracy: 0.8392
Epoch 19/100
668/668 [==============================] - 3s 5ms/step - loss: 9.8893e-04 - accuracy: 0.9997 - val_loss: 0.9766
- val_accuracy: 0.8386
Epoch 20/100
668/668 [==============================] - 3s 5ms/step - loss: 6.6580e-04 - accuracy: 0.9998 - val_loss: 1.0297
- val_accuracy: 0.8371
Epoch 21/100
668/668 [==============================] - 4s 6ms/step - loss: 8.4706e-04 - accuracy: 0.9997 - val_loss: 1.0298
- val_accuracy: 0.8368
Epoch 22/100
668/668 [==============================] - 4s 6ms/step - loss: 6.4325e-04 - accuracy: 0.9997 - val_loss: 0.9667
- val_accuracy: 0.8375
Epoch 23/100
668/668 [==============================] - 5s 7ms/step - loss: 7.0361e-04 - accuracy: 0.9997 - val_loss: 1.0392
- val_accuracy: 0.8371
Epoch 24/100
668/668 [==============================] - 3s 5ms/step - loss: 5.7372e-04 - accuracy: 0.9997 - val_loss: 1.1203
- val_accuracy: 0.8355
Epoch 25/100
668/668 [==============================] - 3s 5ms/step - loss: 5.5205e-04 - accuracy: 0.9997 - val_loss: 1.1018
- val_accuracy: 0.8347
Epoch 26/100
668/668 [==============================] - 3s 5ms/step - loss: 6.2936e-04 - accuracy: 0.9997 - val_loss: 1.1457
- val_accuracy: 0.8356
Epoch 27/100
668/668 [==============================] - 4s 6ms/step - loss: 7.9425e-04 - accuracy: 0.9996 - val_loss: 1.1691
- val_accuracy: 0.8319
Epoch 28/100
668/668 [==============================] - 4s 6ms/step - loss: 4.6386e-04 - accuracy: 0.9998 - val_loss: 1.1700
- val_accuracy: 0.8306
Epoch 29/100
668/668 [==============================] - 4s 6ms/step - loss: 7.3471e-04 - accuracy: 0.9996 - val_loss: 1.0638
```

```
- val_accuracy: 0.8278
Epoch 30/100
668/668 [==============================] - 3s 5ms/step - loss: 5.7317e-04 - accuracy: 0.9996 - val_loss: 1.1987
- val_accuracy: 0.8300
Epoch 31/100
668/668 [==============================] - 3s 5ms/step - loss: 4.7104e-04 - accuracy: 0.9998 - val_loss: 1.2469
- val_accuracy: 0.8270
Epoch 32/100
668/668 [==============================] - 3s 5ms/step - loss: 6.4549e-04 - accuracy: 0.9997 - val_loss: 1.1591
- val_accuracy: 0.8252
Epoch 33/100
668/668 [==============================] - 4s 6ms/step - loss: 4.2772e-04 - accuracy: 0.9996 - val_loss: 1.4147
- val_accuracy: 0.8240
Epoch 34/100
668/668 [==============================] - 3s 5ms/step - loss: 5.0847e-04 - accuracy: 0.9997 - val_loss: 1.1266
- val_accuracy: 0.8313
Epoch 35/100
668/668 [==============================] - 4s 5ms/step - loss: 3.8286e-04 - accuracy: 0.9998 - val_loss: 1.5796
- val_accuracy: 0.8242
Epoch 36/100
668/668 [==============================] - 3s 5ms/step - loss: 4.8310e-04 - accuracy: 0.9996 - val_loss: 1.3295
- val_accuracy: 0.8280
Epoch 37/100
668/668 [==============================] - 3s 5ms/step - loss: 3.7460e-04 - accuracy: 0.9997 - val_loss: 1.4001
- val_accuracy: 0.8252
Epoch 38/100
668/668 [==============================] - 3s 5ms/step - loss: 3.6458e-04 - accuracy: 0.9997 - val_loss: 1.4449
- val_accuracy: 0.8252
Epoch 39/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5788e-04 - accuracy: 0.9998 - val_loss: 1.4592
- val_accuracy: 0.8246
Epoch 40/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5702e-04 - accuracy: 0.9998 - val_loss: 1.5189
- val_accuracy: 0.8244
Epoch 41/100
668/668 [==============================] - 4s 5ms/step - loss: 3.5639e-04 - accuracy: 0.9998 - val_loss: 1.5627
- val_accuracy: 0.8248
Epoch 42/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5668e-04 - accuracy: 0.9998 - val_loss: 1.5991
- val_accuracy: 0.8248
Epoch 43/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5644e-04 - accuracy: 0.9998 - val_loss: 1.4976
- val_accuracy: 0.8252
Epoch 44/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5638e-04 - accuracy: 0.9998 - val_loss: 1.6245
- val_accuracy: 0.8244
Epoch 45/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5620e-04 - accuracy: 0.9998 - val_loss: 1.6791
- val_accuracy: 0.8248
Epoch 46/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5601e-04 - accuracy: 0.9998 - val_loss: 1.7201
- val_accuracy: 0.8255
Epoch 47/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5654e-04 - accuracy: 0.9998 - val_loss: 1.4840
- val_accuracy: 0.8184
Epoch 48/100
668/668 [==============================] - 3s 5ms/step - loss: 9.4522e-04 - accuracy: 0.9997 - val_loss: 1.6828
- val_accuracy: 0.8199
Epoch 49/100
668/668 [==============================] - 3s 5ms/step - loss: 3.6833e-04 - accuracy: 0.9998 - val_loss: 1.7080
- val_accuracy: 0.8246
Epoch 50/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5589e-04 - accuracy: 0.9998 - val_loss: 1.7394
- val_accuracy: 0.8248
Epoch 51/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5611e-04 - accuracy: 0.9998 - val_loss: 1.7683
- val_accuracy: 0.8248
Epoch 52/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5586e-04 - accuracy: 0.9998 - val_loss: 1.7953
- val_accuracy: 0.8248
Epoch 53/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5578e-04 - accuracy: 0.9998 - val_loss: 1.8204
- val_accuracy: 0.8253
Epoch 54/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5566e-04 - accuracy: 0.9998 - val_loss: 1.8466
- val_accuracy: 0.8255
Epoch 55/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5569e-04 - accuracy: 0.9998 - val_loss: 1.8689
- val_accuracy: 0.8246
Epoch 56/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5555e-04 - accuracy: 0.9998 - val_loss: 1.8948
- val_accuracy: 0.8248
Epoch 57/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5565e-04 - accuracy: 0.9998 - val_loss: 1.9191
- val_accuracy: 0.8253
Epoch 58/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5567e-04 - accuracy: 0.9998 - val_loss: 1.9406
- val_accuracy: 0.8250
Epoch 59/100
```

```
668/668 [==============================] - 3s 5ms/step - loss: 3.5594e-04 - accuracy: 0.9998 - val_loss: 1.9649
- val_accuracy: 0.8253
Epoch 60/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5577e-04 - accuracy: 0.9998 - val_loss: 1.9867
- val_accuracy: 0.8261
Epoch 61/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5577e-04 - accuracy: 0.9998 - val_loss: 2.0075
- val_accuracy: 0.8257
Epoch 62/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5547e-04 - accuracy: 0.9998 - val_loss: 2.0254
- val_accuracy: 0.8255
Epoch 63/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5576e-04 - accuracy: 0.9998 - val_loss: 2.0470
- val_accuracy: 0.8259
Epoch 64/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5559e-04 - accuracy: 0.9998 - val_loss: 2.0667
- val_accuracy: 0.8267
Epoch 65/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5577e-04 - accuracy: 0.9998 - val_loss: 2.0842
- val_accuracy: 0.8267
Epoch 66/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5567e-04 - accuracy: 0.9998 - val_loss: 2.1026
- val_accuracy: 0.8268
Epoch 67/100
668/668 [==============================] - 3s 5ms/step - loss: 5.6970e-04 - accuracy: 0.9997 - val_loss: 1.7994
- val_accuracy: 0.8203
Epoch 68/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5593e-04 - accuracy: 0.9998 - val_loss: 1.8914
- val_accuracy: 0.8233
Epoch 69/100
668/668 [==============================] - 4s 5ms/step - loss: 3.5554e-04 - accuracy: 0.9998 - val_loss: 1.9308
- val_accuracy: 0.8224
Epoch 70/100
668/668 [==============================] - 3s 5ms/step - loss: 3.7697e-04 - accuracy: 0.9996 - val_loss: 1.9329
- val_accuracy: 0.8190
Epoch 71/100
668/668 [==============================] - 4s 6ms/step - loss: 3.7612e-04 - accuracy: 0.9996 - val_loss: 2.0308
- val_accuracy: 0.8207
Epoch 72/100
668/668 [==============================] - 3s 5ms/step - loss: 6.0386e-04 - accuracy: 0.9997 - val_loss: 1.9975
- val_accuracy: 0.8205
Epoch 73/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5559e-04 - accuracy: 0.9998 - val_loss: 2.0352
- val_accuracy: 0.8205
Epoch 74/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5553e-04 - accuracy: 0.9998 - val_loss: 2.0568
- val_accuracy: 0.8203
Epoch 75/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5557e-04 - accuracy: 0.9998 - val_loss: 2.0746
- val_accuracy: 0.8203
Epoch 76/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5560e-04 - accuracy: 0.9998 - val_loss: 2.0876
- val_accuracy: 0.8205
Epoch 77/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5565e-04 - accuracy: 0.9998 - val_loss: 2.1035
- val_accuracy: 0.8205
Epoch 78/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5549e-04 - accuracy: 0.9998 - val_loss: 2.1192
- val_accuracy: 0.8207
Epoch 79/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5568e-04 - accuracy: 0.9998 - val_loss: 2.1355
- val_accuracy: 0.8210
Epoch 80/100
668/668 [==============================] - 3s 5ms/step - loss: 6.2551e-04 - accuracy: 0.9997 - val_loss: 2.2294
- val_accuracy: 0.8182
Epoch 81/100
668/668 [==============================] - 3s 5ms/step - loss: 5.5759e-04 - accuracy: 0.9998 - val_loss: 2.0905
- val_accuracy: 0.8192
Epoch 82/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5571e-04 - accuracy: 0.9998 - val_loss: 2.1096
- val_accuracy: 0.8201
Epoch 83/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5554e-04 - accuracy: 0.9998 - val_loss: 2.1285
- val_accuracy: 0.8197
Epoch 84/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5564e-04 - accuracy: 0.9998 - val_loss: 2.1471
- val_accuracy: 0.8197
Epoch 85/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5574e-04 - accuracy: 0.9998 - val_loss: 2.1641
- val_accuracy: 0.8205
Epoch 86/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5592e-04 - accuracy: 0.9998 - val_loss: 2.1803
- val_accuracy: 0.8207
Epoch 87/100
668/668 [==============================] - 4s 5ms/step - loss: 3.5565e-04 - accuracy: 0.9998 - val_loss: 2.1969
- val_accuracy: 0.8205
Epoch 88/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5568e-04 - accuracy: 0.9998 - val_loss: 2.2128
- val_accuracy: 0.8203
```

```
Epoch 89/100
668/668 [==============================] - 4s 5ms/step - loss: 3.5576e-04 - accuracy: 0.9998 - val_loss: 2.2285
- val_accuracy: 0.8209
Epoch 90/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5557e-04 - accuracy: 0.9998 - val_loss: 2.2451
- val_accuracy: 0.8212
Epoch 91/100
668/668 [==============================] - 4s 5ms/step - loss: 3.5554e-04 - accuracy: 0.9998 - val_loss: 2.2602
- val_accuracy: 0.8212
Epoch 92/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5556e-04 - accuracy: 0.9998 - val_loss: 2.2755
- val_accuracy: 0.8216
Epoch 93/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5591e-04 - accuracy: 0.9998 - val_loss: 2.2907
- val_accuracy: 0.8220
Epoch 94/100
668/668 [==============================] - 5s 7ms/step - loss: 3.5580e-04 - accuracy: 0.9998 - val_loss: 2.3060
- val_accuracy: 0.8222
Epoch 95/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5578e-04 - accuracy: 0.9998 - val_loss: 2.3208
- val_accuracy: 0.8224
Epoch 96/100
668/668 [==============================] - 3s 5ms/step - loss: 3.5561e-04 - accuracy: 0.9998 - val_loss: 2.3353
- val_accuracy: 0.8220
Epoch 97/100
668/668 [==============================] - 4s 7ms/step - loss: 3.5549e-04 - accuracy: 0.9998 - val_loss: 2.3486
- val_accuracy: 0.8218
Epoch 98/100
668/668 [==============================] - 5s 7ms/step - loss: 3.5559e-04 - accuracy: 0.9998 - val_loss: 2.3623
- val_accuracy: 0.8210
Epoch 99/100
668/668 [==============================] - 4s 6ms/step - loss: 3.5553e-04 - accuracy: 0.9998 - val_loss: 2.3753
- val_accuracy: 0.8209
Epoch 100/100
668/668 [==============================] - 5s 7ms/step - loss: 3.5580e-04 - accuracy: 0.9998 - val_loss: 2.3891
- val_accuracy: 0.8214
```

In [8]:

```
e = model.layers[0]
weights = e.get_weights()[0]
print(weights.shape)
```

```
(10000, 16)
```

In [9]:

```
fake_reviews = ['The service was not up to par either',
                'Thus far have only visited twice and the food was absolutely delicious each time',
                'Just as good as when I had it more than a year ago',
                'For a self proclaimed coffee cafe I was wildly disappointed',
                'The Veggitarian platter is out of this world',
                'Life’s good, you should get one',
                'Cancel my subscription because I don’t need your issues',
                'I clapped because it’s finished, not because I like it',
                'If had a dollar for every smart thing you say. I’ll be poor']

print(fake_reviews)

# Create the sequences
padding_type='post'
sample_sequences = tokenizer.texts_to_sequences(fake_reviews)
fakes_padded = pad_sequences(sample_sequences, padding=padding_type, maxlen=max_length)

classes = model.predict(fakes_padded)

# The closer the class is to 1, the more positive the review is deemed to be
for x in range(len(fake_reviews)):
  print(fake_reviews[x])
  print(classes[x])
  print('\n')
```

```
['The service was not up to par either', 'Thus far have only visited twice and the food was absolutely deliciou
s each time', 'Just as good as when I had it more than a year ago', 'For a self proclaimed coffee cafe I was wi
ldly disappointed', 'The Veggitarian platter is out of this world', 'Life’s good, you should get one', 'Cancel
my subscription because I don’t need your issues', 'I clapped because it’s finished, not because I like it', 'I
f had a dollar for every smart thing you say. I’ll be poor']
1/1 [==============================] - 1s 623ms/step
The service was not up to par either
[4.852899e-08]


Thus far have only visited twice and the food was absolutely delicious each time
[0.00140244]


Just as good as when I had it more than a year ago
[8.1498386e-14]


For a self proclaimed coffee cafe I was wildly disappointed
[1.5213802e-05]


The Veggitarian platter is out of this world
[8.411127e-21]


Life’s good, you should get one
[0.99996555]


Cancel my subscription because I don’t need your issues
[4.093752e-15]


I clapped because it’s finished, not because I like it
[0.9932086]


If had a dollar for every smart thing you say. I’ll be poor
[0.9608609]
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js