# Assignment 11

Using section 8.1 in Deep Learning with Python as a guide, implement an LSTM text generator. Train the model on the Enron corpus or a text source of your choice. Save the model and generate 20 examples to the results directory of dsc650/assignments/assignment11/

In [1]:
```
# Load required libraries
import os
from pathlib import Path
import numpy as np
import keras
from keras import layers
import random
import sys
from tqdm.notebook import tqdm
```

In [2]:
```
# Load 'Adventures of Sherlock Holmes' data
# data is located in assignment folder or can be optained from https://www.gu

current_dir = Path(os.getcwd()).absolute()
path = os.path.join(current_dir, 'sherlock_holmes.txt')

# Read data as lowercase

text = open(path).read().lower()

print('Corpus length: ', len(text))
```

```
Corpus length:  581889
```

In [3]:
```
# Create results directory

results_dir = current_dir.joinpath('results')
results_dir.mkdir(parents=True, exist_ok=True)
```

In [4]:

```
# Vectorize the data

maxlen = 60
step = 3

sentences = []
next_chars = []

for i in tqdm(range(0, len(text) - maxlen, step)):
    sentences.append(text[i: i + maxlen])
    next_chars.append(text[i + maxlen])

print('Number of sequences: ', len(sentences))

# Create list of unique characters

chars = sorted(list(set(text)))

print('Unique characters: ', len(chars))

# Create dictionary to map unique characters to index

char_indices = dict((char, chars.index(char)) for char in chars)

# One-hhoht encode characters into binary arrays

print('Vectorization...')
x = np.zeros((len(sentences), maxlen, len(chars)), dtype = np.bool)
y = np.zeros((len(sentences), len(chars)), dtype = np.bool)

for i, sentence in tqdm(enumerate(sentences)):
    for t, char in enumerate(sentence):
        x[i, t, char_indices[char]] = 1
    y[i, char_indices[next_chars[i]]] = 1
print('Vectorization Complete.')
```

```
Number of sequences:  193943
Unique characters:  73
Vectorization...

Vectorization Complete.
```

In [10]:

```
# Build the model

model = keras.models.Sequential()
model.add(layers.LSTM(128, input_shape = (maxlen, len(chars))))
model.add(layers.Dense(len(chars), activation = 'softmax'))
```

In [11]:
```
# Compile the model

model.compile(loss = 'categorical_crossentropy',
              optimizer = 'Adam')

# Save the model
model.save('results/text_generation_model.h5')
```

In [12]:
```
# Sample next character based on predictions

def sample(preds, temperature = 1.0):
    preds = np.asarray(preds).astype('float64')
    preds = np.log(preds) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    probas = np.random.multinomial(1, preds, 1)
    return np.argmax(probas)
```

In [13]:
```
# Fit model and text gerenation

for epoch in tqdm(range(1, 20)):
    print('epoch ', epoch)
    model.fit(x, y, batch_size = 128, epochs = 1)
    start_index = random.randint(0, len(text) - maxlen - 1)
    generated_text = text[start_index: start_index + maxlen]
    print('--- Generating with seed: "' + generated_text + '"')

    f = open("results/text_example{}.txt".format(epoch), "w+")
    f.write('--- Generating with seed: "' + generated_text + '"\n')

    for temperature in [0.2, 0.5, 1.0, 1.2]:
        f.write('------ temperature: ' + str(temperature) + '\n')
        f.write(generated_text + '\n')
        for i in range(400):
            sampled = np.zeros((1, maxlen, len(chars)))
            for t, char in enumerate(generated_text):
                sampled[0, t, char_indices[char]] = 1

            preds = model.predict(sampled, verbose = 0)[0]
            next_index = sample(preds, temperature)
            next_char = chars[next_index]

            generated_text += next_char
            generated_text = generated_text[1:]

            f.write(next_char)
        f.write('\n')
    f.close()
```

```
epoch  1
1516/1516 [==============================] - 249s 165ms/step - loss: 2.5463
--- Generating with seed: "to him.

“well, have you solved it?” i asked as i entered.

"
epoch  2
1516/1516 [==============================] - 231s 152ms/step - loss: 2.1971
--- Generating with seed: "hey had locked.

“i am naturally observant, as you may have "
epoch  3
1516/1516 [==============================] - 228s 150ms/step - loss: 2.0840
--- Generating with seed: " the front door,” cried holmes, and we all rushed d
own the
s"
epoch  4
1516/1516 [==============================] - 229s 151ms/step - loss: 2.0086
--- Generating with seed: "lem of the grosvenor square furniture van.
that is quite cle"
epoch  5
1516/1516 [==============================] - 231s 153ms/step - loss: 1.9487
--- Generating with seed: "d
there are many noble families to whom we have advanced lar"
epoch  6
1516/1516 [==============================] - 230s 152ms/step - loss: 1.8983
--- Generating with seed: "s to their nature,” he answered.

“then what are they? who i"
epoch  7
1516/1516 [==============================] - 231s 152ms/step - loss: 1.8545
--- Generating with seed: "e before the altar.”

“perhaps, mrs. moulton, you would like"
epoch  8
1516/1516 [==============================] - 230s 152ms/step - loss: 1.8149
--- Generating with seed: "rom the dead man’s lap, and
throwing the noose round the rep"
epoch  9
1516/1516 [==============================] - 231s 152ms/step - loss: 1.7819
--- Generating with seed: "ned to his desk and, unlocking it, drew out a small
case-boo"
epoch  10
1516/1516 [==============================] - 235s 155ms/step - loss: 1.7521
--- Generating with seed: "e paced up and down in front of briony lodge, waiti
ng for th"
epoch  11
1516/1516 [==============================] - 232s 153ms/step - loss: 1.7259
--- Generating with seed: " frightened and ran
out again. oh, it is so dreadfully still"
epoch  12
1516/1516 [==============================] - 233s 153ms/step - loss: 1.7015
--- Generating with seed: "atever
it was i gave him without question, land, money, hous"
epoch  13
```

```
1516/1516 [==============================] - 239s 158ms/step - loss: 1.6802
--- Generating with seed: "not have a farthing from me,’ i cried, on which he
bowed and"
epoch  14
1516/1516 [==============================] - 233s 154ms/step - loss: 1.6608
--- Generating with seed: "in saying that the flooring and walls are
sound, and that th"
epoch  15
1516/1516 [==============================] - 234s 155ms/step - loss: 1.6421
--- Generating with seed: "the dregs of the docks, breathing in the poison
or sleeping "
epoch  16
1516/1516 [==============================] - 235s 155ms/step - loss: 1.6249
--- Generating with seed: " know, and my poor
little reputation, such as it is, will su"
epoch  17
1516/1516 [==============================] - 234s 154ms/step - loss: 1.6095
--- Generating with seed: " course, one can’t refuse a lady, and such a very p
ositive o"
epoch  18
1516/1516 [==============================] - 245s 161ms/step - loss: 1.5943
--- Generating with seed: "per, but i think, watson, that we shall be able to
strike
de"
epoch  19
1516/1516 [==============================] - 247s 163ms/step - loss: 1.5811
--- Generating with seed: "lowed them from
the cellar, “i do not know how the bank can "
```

In [ ]: