In [1]:

```
#@title Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

In [2]:

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import regularizers
import tensorflow.keras.utils as ku
import numpy as np
```

In [3]:

```
tokenizer = Tokenizer()
!wget --no-check-certificate \
    https://storage.googleapis.com/laurencemoroney-blog.appspot.com/sonnets.txt \
    -O /tmp/sonnets.txt
data = open('/tmp/sonnets.txt').read()

corpus = data.lower().split("\n")


tokenizer.fit_on_texts(corpus)
total_words = len(tokenizer.word_index) + 1

# create input sequences using list of tokens
input_sequences = []
for line in corpus:
 token_list = tokenizer.texts_to_sequences([line])[0]
 for i in range(1, len(token_list)):
  n_gram_sequence = token_list[:i+1]
  input_sequences.append(n_gram_sequence)


# pad sequences
max_sequence_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_sequence_len, padding='pre'))

# create predictors and label
predictors, label = input_sequences[:,:-1],input_sequences[:,-1]

label = ku.to_categorical(label, num_classes=total_words)
```

```
--2020-10-06 13:00:09--  https://storage.googleapis.com/laurencemoroney-
blog.appspot.com/sonnets.txt
Resolving storage.googleapis.com (storage.googleapis.com)... 64.233.167.128, 74.125.133.128,
74.125.140.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|64.233.167.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 93578 (91K) [text/plain]
Saving to: '/tmp/sonnets.txt'

/tmp/sonnets.txt    100%[===================>]  91.38K  --.-KB/s    in 0.002s

2020-10-06 13:00:10 (56.1 MB/s) - '/tmp/sonnets.txt' saved [93578/93578]
```

In [4]:

```
model = Sequential()
model.add(Embedding(total_words, 100, input_length=max_sequence_len-1))
model.add(Bidirectional(LSTM(150, return_sequences = True)))
model.add(Dropout(0.2))
model.add(LSTM(100))
model.add(Dense(total_words/2, activation='relu', kernel_regularizer=regularizers.l2(0.01)))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding (Embedding)        (None, 10, 100)           321100    
_________________________________________________________________
bidirectional (Bidirectional (None, 10, 300)           301200    
_________________________________________________________________
dropout (Dropout)            (None, 10, 300)           0         
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               160400    
_________________________________________________________________
dense (Dense)                (None, 1605)              162105    
_________________________________________________________________
dense_1 (Dense)              (None, 3211)              5156866   
=================================================================
Total params: 6,101,671
Trainable params: 6,101,671
Non-trainable params: 0
_________________________________________________________________
None
```

In [5]:

```
history = model.fit(predictors, label, epochs=100, verbose=1)
```

```
Epoch 1/100
484/484 [==============================] - 13s 26ms/step - loss: 6.9131 - accuracy: 0.0213
Epoch 2/100
484/484 [==============================] - 13s 26ms/step - loss: 6.5012 - accuracy: 0.0237
Epoch 3/100
484/484 [==============================] - 13s 26ms/step - loss: 6.4064 - accuracy: 0.0254
Epoch 4/100
484/484 [==============================] - 13s 26ms/step - loss: 6.2829 - accuracy: 0.0298
Epoch 5/100
484/484 [==============================] - 13s 26ms/step - loss: 6.1736 - accuracy: 0.0360
Epoch 6/100
484/484 [==============================] - 13s 26ms/step - loss: 6.0823 - accuracy: 0.0385
Epoch 7/100
484/484 [==============================] - 13s 26ms/step - loss: 5.9952 - accuracy: 0.0423
Epoch 8/100
484/484 [==============================] - 13s 26ms/step - loss: 5.9050 - accuracy: 0.0451
Epoch 9/100
484/484 [==============================] - 13s 26ms/step - loss: 5.7981 - accuracy: 0.0519
Epoch 10/100
484/484 [==============================] - 13s 26ms/step - loss: 5.6816 - accuracy: 0.0614
Epoch 11/100
484/484 [==============================] - 13s 26ms/step - loss: 5.5676 - accuracy: 0.0666
Epoch 12/100
484/484 [==============================] - 13s 26ms/step - loss: 5.4530 - accuracy: 0.0733
Epoch 13/100
484/484 [==============================] - 13s 26ms/step - loss: 5.3455 - accuracy: 0.0820
Epoch 14/100
484/484 [==============================] - 13s 27ms/step - loss: 5.2316 - accuracy: 0.0908
Epoch 15/100
484/484 [==============================] - 13s 27ms/step - loss: 5.1231 - accuracy: 0.0955
Epoch 16/100
484/484 [==============================] - 13s 26ms/step - loss: 5.0147 - accuracy: 0.1035
Epoch 17/100
484/484 [==============================] - 13s 26ms/step - loss: 4.9118 - accuracy: 0.1126
Epoch 18/100
484/484 [==============================] - 13s 26ms/step - loss: 4.8062 - accuracy: 0.1229
```

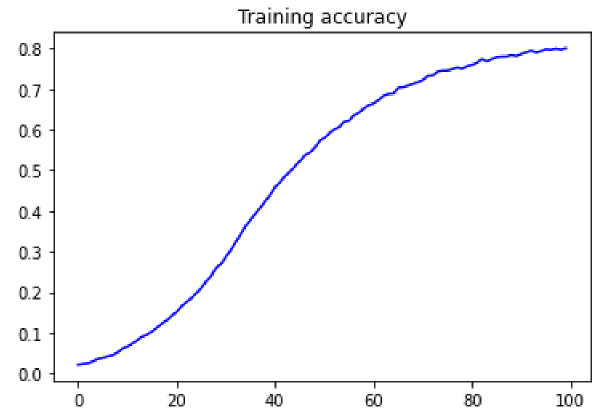
```
Epoch 19/100
484/484 [==============================] - 13s 27ms/step - loss: 4.6997 - accuracy: 0.1308
Epoch 20/100
484/484 [==============================] - 13s 27ms/step - loss: 4.5999 - accuracy: 0.1418
Epoch 21/100
484/484 [==============================] - 13s 27ms/step - loss: 4.4983 - accuracy: 0.1513
Epoch 22/100
484/484 [==============================] - 13s 27ms/step - loss: 4.3984 - accuracy: 0.1651
Epoch 23/100
484/484 [==============================] - 13s 27ms/step - loss: 4.3028 - accuracy: 0.1757
Epoch 24/100
484/484 [==============================] - 13s 27ms/step - loss: 4.2044 - accuracy: 0.1852
Epoch 25/100
484/484 [==============================] - 13s 27ms/step - loss: 4.1068 - accuracy: 0.1980
Epoch 26/100
484/484 [==============================] - 13s 27ms/step - loss: 4.0084 - accuracy: 0.2101
Epoch 27/100
484/484 [==============================] - 13s 27ms/step - loss: 3.9090 - accuracy: 0.2267
Epoch 28/100
484/484 [==============================] - 13s 27ms/step - loss: 3.8215 - accuracy: 0.2395
Epoch 29/100
484/484 [==============================] - 13s 27ms/step - loss: 3.7222 - accuracy: 0.2601
Epoch 30/100
484/484 [==============================] - 13s 27ms/step - loss: 3.6341 - accuracy: 0.2691
Epoch 31/100
484/484 [==============================] - 13s 27ms/step - loss: 3.5549 - accuracy: 0.2868
Epoch 32/100
484/484 [==============================] - 13s 27ms/step - loss: 3.4714 - accuracy: 0.3036
Epoch 33/100
484/484 [==============================] - 13s 27ms/step - loss: 3.3785 - accuracy: 0.3229
Epoch 34/100
484/484 [==============================] - 13s 27ms/step - loss: 3.2973 - accuracy: 0.3421
Epoch 35/100
484/484 [==============================] - 13s 27ms/step - loss: 3.2175 - accuracy: 0.3624
Epoch 36/100
484/484 [==============================] - 13s 27ms/step - loss: 3.1437 - accuracy: 0.3774
Epoch 37/100
484/484 [==============================] - 13s 27ms/step - loss: 3.0644 - accuracy: 0.3934
Epoch 38/100
484/484 [==============================] - 13s 27ms/step - loss: 3.0016 - accuracy: 0.4075
Epoch 39/100
484/484 [==============================] - 13s 27ms/step - loss: 2.9272 - accuracy: 0.4240
Epoch 40/100
484/484 [==============================] - 13s 27ms/step - loss: 2.8601 - accuracy: 0.4396
Epoch 41/100
484/484 [==============================] - 13s 27ms/step - loss: 2.7910 - accuracy: 0.4589
Epoch 42/100
484/484 [==============================] - 13s 27ms/step - loss: 2.7319 - accuracy: 0.4705
Epoch 43/100
484/484 [==============================] - 13s 26ms/step - loss: 2.6667 - accuracy: 0.4858
Epoch 44/100
484/484 [==============================] - 13s 26ms/step - loss: 2.6084 - accuracy: 0.4959
Epoch 45/100
484/484 [==============================] - 13s 27ms/step - loss: 2.5656 - accuracy: 0.5096
Epoch 46/100
484/484 [==============================] - 13s 27ms/step - loss: 2.5026 - accuracy: 0.5222
Epoch 47/100
484/484 [==============================] - 13s 27ms/step - loss: 2.4462 - accuracy: 0.5361
Epoch 48/100
484/484 [==============================] - 13s 27ms/step - loss: 2.4026 - accuracy: 0.5433
Epoch 49/100
484/484 [==============================] - 13s 27ms/step - loss: 2.3620 - accuracy: 0.5541
Epoch 50/100
484/484 [==============================] - 13s 27ms/step - loss: 2.2968 - accuracy: 0.5717
Epoch 51/100
484/484 [==============================] - 13s 27ms/step - loss: 2.2534 - accuracy: 0.5796
Epoch 52/100
484/484 [==============================] - 13s 27ms/step - loss: 2.2069 - accuracy: 0.5903
Epoch 53/100
484/484 [==============================] - 13s 27ms/step - loss: 2.1611 - accuracy: 0.6008
Epoch 54/100
484/484 [==============================] - 13s 27ms/step - loss: 2.1316 - accuracy: 0.6059
Epoch 55/100
484/484 [==============================] - 13s 26ms/step - loss: 2.0797 - accuracy: 0.6191
Epoch 56/100
484/484 [==============================] - 13s 27ms/step - loss: 2.0554 - accuracy: 0.6220
Epoch 57/100
```

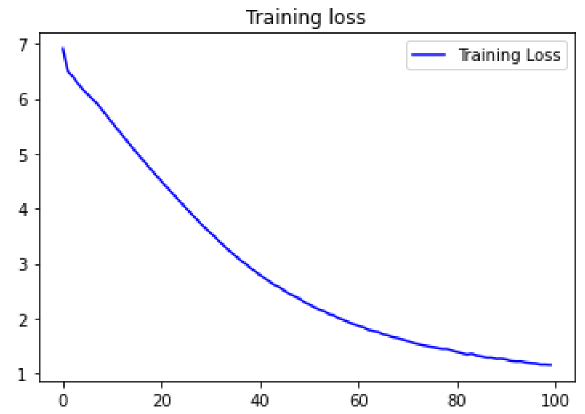
```
484/484 [==============================] - 13s 27ms/step - loss: 2.0009 - accuracy: 0.6351
Epoch 58/100
484/484 [==============================] - 13s 27ms/step - loss: 1.9713 - accuracy: 0.6418
Epoch 59/100
484/484 [==============================] - 13s 27ms/step - loss: 1.9305 - accuracy: 0.6519
Epoch 60/100
484/484 [==============================] - 13s 27ms/step - loss: 1.8968 - accuracy: 0.6599
Epoch 61/100
484/484 [==============================] - 13s 27ms/step - loss: 1.8655 - accuracy: 0.6641
Epoch 62/100
484/484 [==============================] - 13s 27ms/step - loss: 1.8384 - accuracy: 0.6729
Epoch 63/100
484/484 [==============================] - 13s 27ms/step - loss: 1.7925 - accuracy: 0.6827
Epoch 64/100
484/484 [==============================] - 13s 27ms/step - loss: 1.7706 - accuracy: 0.6878
Epoch 65/100
484/484 [==============================] - 13s 27ms/step - loss: 1.7536 - accuracy: 0.6885
Epoch 66/100
484/484 [==============================] - 13s 27ms/step - loss: 1.7097 - accuracy: 0.7029
Epoch 67/100
484/484 [==============================] - 13s 27ms/step - loss: 1.6906 - accuracy: 0.7042
Epoch 68/100
484/484 [==============================] - 13s 27ms/step - loss: 1.6586 - accuracy: 0.7083
Epoch 69/100
484/484 [==============================] - 13s 27ms/step - loss: 1.6408 - accuracy: 0.7127
Epoch 70/100
484/484 [==============================] - 13s 27ms/step - loss: 1.6154 - accuracy: 0.7169
Epoch 71/100
484/484 [==============================] - 13s 26ms/step - loss: 1.5878 - accuracy: 0.7218
Epoch 72/100
484/484 [==============================] - 13s 27ms/step - loss: 1.5584 - accuracy: 0.7326
Epoch 73/100
484/484 [==============================] - 13s 27ms/step - loss: 1.5333 - accuracy: 0.7333
Epoch 74/100
484/484 [==============================] - 13s 27ms/step - loss: 1.5134 - accuracy: 0.7427
Epoch 75/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4937 - accuracy: 0.7452
Epoch 76/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4794 - accuracy: 0.7453
Epoch 77/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4625 - accuracy: 0.7489
Epoch 78/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4438 - accuracy: 0.7529
Epoch 79/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4435 - accuracy: 0.7497
Epoch 80/100
484/484 [==============================] - 13s 27ms/step - loss: 1.4149 - accuracy: 0.7562
Epoch 81/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3886 - accuracy: 0.7595
Epoch 82/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3675 - accuracy: 0.7643
Epoch 83/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3405 - accuracy: 0.7738
Epoch 84/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3561 - accuracy: 0.7678
Epoch 85/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3228 - accuracy: 0.7736
Epoch 86/100
484/484 [==============================] - 13s 27ms/step - loss: 1.3082 - accuracy: 0.7785
Epoch 87/100
484/484 [==============================] - 13s 28ms/step - loss: 1.2856 - accuracy: 0.7798
Epoch 88/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2848 - accuracy: 0.7800
Epoch 89/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2638 - accuracy: 0.7833
Epoch 90/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2663 - accuracy: 0.7809
Epoch 91/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2512 - accuracy: 0.7864
Epoch 92/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2249 - accuracy: 0.7902
Epoch 93/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2158 - accuracy: 0.7948
Epoch 94/100
484/484 [==============================] - 13s 27ms/step - loss: 1.2166 - accuracy: 0.7899
Epoch 95/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1963 - accuracy: 0.7932
```

```
Epoch 96/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1870 - accuracy: 0.7978
Epoch 97/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1786 - accuracy: 0.7966
Epoch 98/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1597 - accuracy: 0.7996
Epoch 99/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1600 - accuracy: 0.7969
Epoch 100/100
484/484 [==============================] - 13s 27ms/step - loss: 1.1523 - accuracy: 0.8003
```

In [6]:

```python
import matplotlib.pyplot as plt
acc = history.history['accuracy']
loss = history.history['loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training accuracy')
plt.title('Training accuracy')

plt.figure()

plt.plot(epochs, loss, 'b', label='Training Loss')
plt.title('Training loss')
plt.legend()

plt.show()
```





In [7]:

```python
seed_text = "Help me Obi Wan Kenobi, you're my only hope"
next_words = 100

for _ in range(next_words):
 token_list = tokenizer.texts_to_sequences([seed_text])[0]
 token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
 predicted = model.predict_classes(token_list, verbose=0)
 output_word = ""
 for word, index in tokenizer.word_index.items():
```

```
  if index == predicted:
   output_word = word
   break
 seed_text += " " + output_word
print(seed_text)
```

```
WARNING:tensorflow:From <ipython-input-7-622d307fa19a>:7: Sequential.predict_classes (from
tensorflow.python.keras.engine.sequential) is deprecated and will be removed after 2021-01-01.
Instructions for updating:
Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class
classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).as
type("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-
layer activation).
Help me Obi Wan Kenobi, you're my only hope itself what in other wrong days forsworn me o'er date
substance wide shore state did glance aside done fight take so speechless waste alone cured alone
still make thine old end shines so green thee thence not reckon'd say ill ill ill ill new ' must t
hee best ' forsworn me best sweetness tell 'tis thee back seen high distill'd by both her gate dim
m'd hate words new grow wide old can grow old old night of ill men ' must thee find find each o'er
pride be shows be seen thyself be done chide my heart so rare wrong
```

In [ ]: