In [1]:
```python
#importing necessary libraries
from tensorflow.keras.layers import Embedding,LSTM,Dropout,Dense
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras .preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
import numpy as np

#Loading the IMDB file

if __name__ == "__main__":
    import urllib.request as req
    import tarfile
    import os

    imdb_url = "http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz"

    save_filename = "aclImdb_v1.tar.gz"
    if not os.path.exists(save_filename):
        req.urlretrieve(imdb_url, save_filename)

    imdb_folder = "aclImdb"
    if not os.path.exists(imdb_folder):
        with tarfile.open(save_filename) as tar:
            tar.extractall()
    import numpy as np
    import re
#loading the train set into a train file alongside its labels

    def get_train_file(data_folder="/train"):
        reviews = []
        labels = []
        for index,sentiment in enumerate(["/neg/", "/pos/"]):
            path = imdb_folder + data_folder + sentiment
            for filename in sorted(os.listdir(path)):
                with open(path + filename, 'r') as f:
                    review = f.read()
                    review = review.lower()
                    review = review.replace("<br />", " ")
                    review = re.sub(r"[^a-z ]", " ", review)
                    review = re.sub(r" +", " ", review)
                    reviews.append(review)

                    label = [0,0]
                    label[index] = 1
                    labels.append(label)
        return reviews, np.array(labels)
    train_reviews, train_labels = get_train_file()
    voc_size=10000#vocabulary size
    onehot_repr=[one_hot(words, voc_size) for words in train_reviews]#Performing vectorization
    sent_len=100#Sentence length
    train_reviews=pad_sequences(onehot_repr , maxlen=sent_len)#Padding the vecorized word to fit into t
he defined sentence length
    embed_size=128
    X_train=train_reviews[:15000]
    X_val=train_reviews[15000:]
    y_train=train_labels[:15000]
    y_val=train_labels[15000:]
        # 2. Train your network
    model=Sequential()
    model.add(Embedding(voc_size, embed_size,input_shape=(X_train.shape [1],)))
    model.add(LSTM(units=60, activation='tanh',return_sequences=True))
    model.add(Dropout(0.4))
    model.add(LSTM(units=60,activation='tanh',return_sequences=True))
    model.add(Dropout(0.4))
    model.add(LSTM(units=60,activation='tanh',return_sequences=True))
    model.add(Dropout(0.4))
    model.add(LSTM(units=60))
    model.add(Dropout(0.4))
    model.add(Dense(units=2, activation='softmax'))
    model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=20, batch_size=128, validation_data=(X_val, y_val))
    model.save('20868189_NLP_model')
```

```
Epoch 1/20
118/118 [==============================] - 40s 340ms/step - loss: 0.3917 - accuracy: 0.8483 - val_los
s: 1.1580 - val_accuracy: 0.4810
Epoch 2/20
118/118 [==============================] - 39s 328ms/step - loss: 0.2038 - accuracy: 0.9261 - val_los
s: 1.4106 - val_accuracy: 0.4915
Epoch 3/20
118/118 [==============================] - 37s 313ms/step - loss: 0.1243 - accuracy: 0.9574 - val_los
s: 1.9498 - val_accuracy: 0.4674
Epoch 4/20
118/118 [==============================] - 36s 307ms/step - loss: 0.0796 - accuracy: 0.9739 - val_los
s: 2.7682 - val_accuracy: 0.2990
Epoch 5/20
118/118 [==============================] - 36s 308ms/step - loss: 0.0496 - accuracy: 0.9856 - val_los
s: 3.0055 - val_accuracy: 0.3140
Epoch 6/20
118/118 [==============================] - 38s 319ms/step - loss: 0.0318 - accuracy: 0.9912 - val_los
s: 3.2559 - val_accuracy: 0.4104
Epoch 7/20
118/118 [==============================] - 37s 313ms/step - loss: 0.0199 - accuracy: 0.9945 - val_los
s: 3.2023 - val_accuracy: 0.4091
Epoch 8/20
118/118 [==============================] - 58s 495ms/step - loss: 0.0130 - accuracy: 0.9970 - val_los
s: 4.2566 - val_accuracy: 0.3859
Epoch 9/20
118/118 [==============================] - 63s 530ms/step - loss: 0.0195 - accuracy: 0.9937 - val_los
s: 3.7365 - val_accuracy: 0.4214
Epoch 10/20
118/118 [==============================] - 38s 321ms/step - loss: 0.0122 - accuracy: 0.9967 - val_los
s: 3.2959 - val_accuracy: 0.4752
Epoch 11/20
118/118 [==============================] - 41s 348ms/step - loss: 0.0097 - accuracy: 0.9976 - val_los
s: 4.1530 - val_accuracy: 0.3605
Epoch 12/20
118/118 [==============================] - 39s 332ms/step - loss: 0.0090 - accuracy: 0.9975 - val_los
s: 3.1669 - val_accuracy: 0.5000
Epoch 13/20
118/118 [==============================] - 38s 323ms/step - loss: 0.0180 - accuracy: 0.9949 - val_los
s: 4.8119 - val_accuracy: 0.3732
Epoch 14/20
118/118 [==============================] - 36s 306ms/step - loss: 0.0101 - accuracy: 0.9967 - val_los
s: 4.7915 - val_accuracy: 0.3936
Epoch 15/20
118/118 [==============================] - 37s 311ms/step - loss: 0.0077 - accuracy: 0.9979 - val_los
s: 3.9205 - val_accuracy: 0.4591
Epoch 16/20
118/118 [==============================] - 36s 307ms/step - loss: 0.0164 - accuracy: 0.9949 - val_los
s: 2.9553 - val_accuracy: 0.4774
Epoch 17/20
118/118 [==============================] - 39s 329ms/step - loss: 0.0095 - accuracy: 0.9970 - val_los
s: 3.1556 - val_accuracy: 0.4987
Epoch 18/20
118/118 [==============================] - 38s 320ms/step - loss: 0.0090 - accuracy: 0.9970 - val_los
s: 4.0217 - val_accuracy: 0.4538
Epoch 19/20
118/118 [==============================] - 37s 318ms/step - loss: 0.0091 - accuracy: 0.9971 - val_los
s: 4.9740 - val_accuracy: 0.3598
Epoch 20/20
118/118 [==============================] - 37s 310ms/step - loss: 0.0021 - accuracy: 0.9995 - val_los
s: 5.3723 - val_accuracy: 0.4147
WARNING:tensorflow:From /Users/eberechukwukathomas/opt/anaconda3/lib/python3.7/site-packages/tensorfl
ow/python/ops/resource_variable_ops.py:1817: calling BaseResourceVariable.__init__ (from tensorflow.p
ython.ops.resource_variable_ops) with constraint is deprecated and will be removed in a future versio
n.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
INFO:tensorflow:Assets written to: 20868189_NLP_model/assets
```

In [ ]:

In [ ]: