In [1]:

```
# Import necessary libraries
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
```

```
C:\Users\chtan\anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required
for this version of SciPy (detected version 1.26.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:

```
# Load the IMDB dataset
(train_data, train_labels), (test_data, test_labels) = tf.keras.datasets.imdb.load_data(num_words=10000)
```

In [3]:

```
# Convert the integers back to words (decoding)
word_index = tf.keras.datasets.imdb.get_word_index()
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])

def decode_review(text):
    return ' '.join([reverse_word_index.get(i, '?') for i in text])

# the decode_review function was defined, but never called. which caused the preprocessing to fail, since the data remain unreve
# hence the error: 'int' has no attribute lower()

train_data = decode_review(reverse_word_index)
test_data = decode_review(reverse_word_index)
```

In [4]:

```
# Preprocess the data
tokenizer = Tokenizer(num_words=10000, oov_token="<OOV>")
tokenizer.fit_on_texts(train_data)

train_sequences = tokenizer.texts_to_sequences(train_data)
train_padded = pad_sequences(train_sequences, maxlen=100, padding='post', truncating='post')

test_sequences = tokenizer.texts_to_sequences(test_data)
test_padded = pad_sequences(test_sequences, maxlen=100, padding='post', truncating='post')
```

In [5]:

```
# Define the LSTM model
model = Sequential([
    Embedding(10000, 16, input_length=100),
    LSTM(32, return_sequences=True),
    LSTM(32),
    Dense(24, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

In [6]:

```
# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [7]:

```
# Train the model
model.fit(train_padded, train_labels, epochs=5, validation_data=(test_padded, test_labels))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_9516\36782277.py in <module>
      1 # Train the model
----> 2 model.fit(train_padded, train_labels, epochs=5, validation_data=(test_padded, test_labels))

~\anaconda3\lib\site-packages\keras\src\utils\traceback_utils.py in error_handler(*args, **kwargs)
     68             # To get the full stack trace, call:
     69             # `tf.debugging.disable_traceback_filtering()`
---> 70             raise e.with_traceback(filtered_tb) from None
     71         finally:
     72             del filtered_tb

~\anaconda3\lib\site-packages\keras\src\engine\data_adapter.py in _check_data_cardinality(data)
   1958             )
   1959         msg += "Make sure all arrays contain the same number of samples."
-> 1960         raise ValueError(msg)
   1961 
   1962 

ValueError: Data cardinality is ambiguous:
  x sizes: 756597
  y sizes: 25000
Make sure all arrays contain the same number of samples.
```

In [ ]: