In [1]:
```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

In [2]:
```
# Number of words to consider as features
vocab_size = 10000
# Cut texts after this number of words (among top max_features most common words)
maxlen = 500

# Load data
(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=vocab_size)

# Pad sequences with max length
train_data = pad_sequences(train_data, maxlen=maxlen)
test_data = pad_sequences(test_data, maxlen=maxlen)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb.npz
17464789/17464789 ━━━━━━━━━━━━━━━━━━━━ 3s 0us/step
```

In [3]:
```
model = models.Sequential()
model.add(layers.Embedding(vocab_size, 32, input_length=maxlen))
model.add(layers.Flatten())
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))  # Sigmoid output for binary classification

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

```
C:\Users\admin\AppData\Roaming\Python\Python311\site-packages\keras\src\layers\core\embedding.py:86: UserWarning
: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

In [4]:
```
history = model.fit(train_data, train_labels,
                    epochs=10,
                    batch_size=512,
                    validation_split=0.2)
```

```
Epoch 1/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 2s 32ms/step - accuracy: 0.5335 - loss: 0.6847 - val_accuracy: 0.7600 - val_loss: 0.5
142
Epoch 2/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.8450 - loss: 0.3982 - val_accuracy: 0.8102 - val_loss: 0.4
461
Epoch 3/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.9059 - loss: 0.2404 - val_accuracy: 0.8668 - val_loss: 0.3
169
Epoch 4/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - accuracy: 0.9541 - loss: 0.1430 - val_accuracy: 0.8690 - val_loss: 0.3
063
Epoch 5/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.9748 - loss: 0.0903 - val_accuracy: 0.8662 - val_loss: 0.3
235
Epoch 6/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.9930 - loss: 0.0475 - val_accuracy: 0.8702 - val_loss: 0.3
363
Epoch 7/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.9976 - loss: 0.0272 - val_accuracy: 0.8690 - val_loss: 0.3
549
Epoch 8/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - accuracy: 0.9991 - loss: 0.0160 - val_accuracy: 0.8684 - val_loss: 0.3
735
Epoch 9/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.9995 - loss: 0.0110 - val_accuracy: 0.8700 - val_loss: 0.3
880
Epoch 10/10
40/40 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - accuracy: 0.9999 - loss: 0.0070 - val_accuracy: 0.8642 - val_loss: 0.4
164
```

In [5]:
```
results = model.evaluate(test_data, test_labels)
print(f"Test Loss: {results[0]}, Test Accuracy: {results[1]}")
```

```
782/782 ━━━━━━━━━━━━━━━━━━━━ 1s 860us/step - accuracy: 0.8623 - loss: 0.4152
Test Loss: 0.4208759367465973, Test Accuracy: 0.8629599809646606
```

In [9]:
```
model.add(layers.Dense(1, activation = "sigmoid"))
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 500, 32)	320,000
flatten (Flatten)	(None, 16000)	0
dense (Dense)	(None, 32)	512,032
dense_1 (Dense)	(None, 1)	33
dense_2 (Dense)	(None, 1)	2

Total params: 2,496,199 (9.52 MB)

Trainable params: 832,067 (3.17 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 1,664,132 (6.35 MB)

In [10]:
```
pip install scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (
1.26.4)
Requirement already satisfied: scipy>=1.3.2 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (1
.11.4)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn) (
1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-l
earn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import numpy as np
```

In [12]:
```
# Predicting the probabilities
probabilities = model.predict(test_data)
predictions = (probabilities > 0.5).astype("int32")
```

```
782/782 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
```

Generating the classification report which includes precision, recall, and F1-score print(classification_report(test_labels, predictions, target_names=['Negative', 'Positive'])) # Computing the accuracy accuracy = accuracy_score(test_labels, predictions) print(f"Accuracy: {accuracy:.4f}") # Generate the confusion matrix conf_matrix = confusion_matrix(test_labels, predictions) print("Confusion Matrix:") print(conf_matrix)

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.heatmap(conf_matrix, annot=True, fmt="d", cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=[
plt.ylabel('Actual Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix')
plt.show()
```