Model Development Phase Template

Date	5 November 2024
Team ID	SWTID1726832093
Project Title	Analysis of Amazon Cell Phone Reviews Using NLP Technique
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

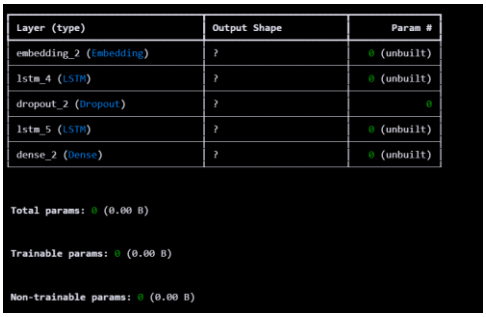
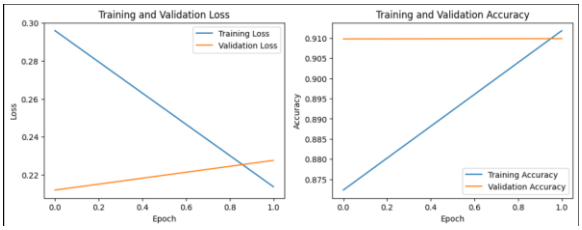
Initial Model Training Code (5 marks):

```python
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import plot_model
import matplotlib.pyplot as plt

# Load your CSV file
data = pd.read_csv('20191226-reviews.csv')

# Preprocess text and labels, handling NaN values in 'body' column
data['body'] = data['body'].fillna('')  # Replace NaNs with empty strings
reviews = data['body'].astype(str).values  # Convert all entries to strings

# Convert ratings to binary labels (1 if rating >= 3, else 0)
labels = data['rating'].apply(lambda x: 1 if x >= 3 else 0).values

# Text tokenization
tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(reviews)
sequences = tokenizer.texts_to_sequences(reviews)
X = pad_sequences(sequences, maxlen=200)
y = np.array(labels)

# Define the neural network model
model = Sequential([
    Embedding(input_dim=10000, output_dim=128, input_length=200),
    LSTM(64, return_sequences=True),
    Dropout(0.5),
    LSTM(32),
    Dense(1, activation='sigmoid')
])
```

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

# Display the model summary and save it as a screenshot
model.summary()

# Train the model and capture the training history
history = model.fit(X, y, validation_split=0.2, epochs=2, batch_size=32, verbose=1)

# Save model summary as an image
plot_model(model, to_file='model_summary.png', show_shapes=True, show_layer_names=True)

# Plotting Training and Validation Metrics
plt.figure(figsize=(12, 4))

# Plot for Loss
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

# Plot for Accuracy
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

# Save and show plots
plt.savefig('training_validation_metrics.png')
plt.show()
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics

sequential	 Layer (type) Output Shape Param # embedding_2 (Embedding) ? 0 (unbuilt) lstm_4 (LSTM) ? 0 (unbuilt) dropout_2 (Dropout) ? 0 lstm_5 (LSTM) ? 0 (unbuilt) dense_2 (Dense) ? 0 (unbuilt) Total params: 0 (0.00 B) Trainable params: 0 (0.00 B) Non-trainable params: 0 (0.00 B)	 Training and Validation Loss Training Loss Validation Loss Loss Epoch Training and Validation Accuracy Training Accuracy Validation Accuracy Accuracy Epoch