Report for Arabic Sentiment Analysis

Overview:

The provided code implements sentiment analysis on Arabic text reviews using two different models: a Transformer model and a Recurrent Neural Network (RNN). The sentiment labels are classified into three categories (ratings -1, 0, and 1) based on the content of the reviews. The code includes data preprocessing, tokenization, and the construction and training of both the Transformer and RNN models.

Components:

1. Data Preprocessing:

- Text cleaning: Standardizes Arabic text by replacing specific characters, removing diacritics, and converting to lowercase.
- Tokenization: Splits text into individual words.
- Stopword removal: Filters out common Arabic and English stopwords.
- Stemming: Applies ISRIStemmer for stemming Arabic words.
- Punctuation removal: Eliminates Arabic and English punctuation.
- Lemmatization: Reduces words to their base form using WordNetLemmatizer.
- Label adjustment: Modifies ratings from -1, 0, 1 to 0, 1, 2 for **to_categorical**.

2. Tokenizer Saving:

- Saves the tokenizer used for text processing to a file named 'tokenizer.pkl'. This tokenizer can be loaded later for consistency in tokenizing new data.

3. Positional Encoding:

- Implements a function for positional encoding, a technique used in the Transformer model.

4. Transformer Model:

- Architecture: Multi-head self-attention mechanism with positional encoding and feed-forward layers.
- Compilation: Uses Adam optimizer and categorical crossentropy loss.
- Training: Fit on the training data for 5 epochs with a batch size of 64.
- Evaluation: Prints and saves accuracy and loss metrics.

5. RNN Model:

- Architecture: Embedding layer, two LSTM layers, dropout layer, and dense layer.
- Compilation: Uses Adam optimizer and categorical crossentropy loss.
- Training: Fit on the training data for 5 epochs with a batch size of 64.
- Evaluation: Prints accuracy and loss metrics.

File Usage:

- **Train Data**: Loaded from an Excel file named 'train.xlsx'.
- **Model Saving:**
    - The trained Transformer model is saved as 'transformer_model.h5'.
    - The trained RNN model is saved as 'rnn3.h5'.

Results:

RNN: -

```
Epoch 1/5
401/401 [==============================] - 501s 1s/step - loss: 0.5596 - categorical_accuracy: 0.7825 - val_loss: 0.4739 - val_categorical_accuracy: 0.8263
Epoch 2/5
401/401 [==============================] - 482s 1s/step - loss: 0.4242 - categorical_accuracy: 0.8487 - val_loss: 0.4631 - val_categorical_accuracy: 0.8252
Epoch 3/5
401/401 [==============================] - 470s 1s/step - loss: 0.3678 - categorical_accuracy: 0.8708 - val_loss: 0.4946 - val_categorical_accuracy: 0.8232
Epoch 4/5
401/401 [==============================] - 454s 1s/step - loss: 0.3307 - categorical_accuracy: 0.8823 - val_loss: 0.4990 - val_categorical_accuracy: 0.8194
Epoch 5/5
401/401 [==============================] - 438s 1s/step - loss: 0.3000 - categorical_accuracy: 0.8949 - val_loss: 0.5428 - val_categorical_accuracy: 0.8143
Accuracy: 81.429464
Loss: 0.542774
```

Transformers:

```
Epoch 1/5
320/320 [==============================] - 277s 858ms/step - loss: 0.8425 - accuracy: 0.7175 - val_loss: 0.5001 - val_accuracy: 0.8203
Epoch 2/5
320/320 [==============================] - 258s 807ms/step - loss: 0.4396 - accuracy: 0.8459 - val_loss: 0.5119 - val_accuracy: 0.8230
Epoch 3/5
320/320 [==============================] - 267s 835ms/step - loss: 0.3114 - accuracy: 0.8931 - val_loss: 0.6401 - val_accuracy: 0.7994
Epoch 4/5
320/320 [==============================] - 256s 801ms/step - loss: 0.2262 - accuracy: 0.9226 - val_loss: 0.6782 - val_accuracy: 0.8035
Epoch 5/5
320/320 [==============================] - 257s 804ms/step - loss: 0.1851 - accuracy: 0.9367 - val_loss: 0.7041 - val_accuracy: 0.7980
C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\keras\src\engine\training.py:3079: UserWarning: You are saving your model as
  saving_api.save_model(
Accuracy: 79.369539
Loss: 0.744092
```