Text Semantic Analysis using Neural Networks (RNN)

Project Overview:

This project focused on performing text semantic analysis using Recurrent Neural Networks (RNNs). The objective was to classify text data into various semantic categories, including "positive," "negative," and "neutral." The report provides an overview of the project, data processing, model architecture, training, and results.

Data Collection and Preparation:

Total Data: The dataset comprised 27,480 samples.

Data Cleaning and Preprocessing:

- Data Balancing
- Conversion of text data into string format.
- Removal of special characters.
- Conversion of text to lowercase.
- Elimination of stopwords.
- Removal of duplicate entries.
- Cleaning of HTML tags.

Data Splitting: The dataset was divided into a training set and a testing set to facilitate model training and evaluation.

Tokenization: The text data was tokenized to convert it into a format suitable for input into the RNN.

Padding: Sequence padding was applied to ensure all input sequences had the same length.

Label Encoding: Sentiment labels (positive, negative, neutral) were encoded to numeric values for model training.

Model Architecture:

```
model = Sequential()

model.add(Embedding(input_dim=18672, output_dim=100, input_length=100))

model.add(SimpleRNN(units=128))

model.add(Dense(units=3, activation='softmax'))
```

Embedding Layer: This layer was used to transform the input data into dense vectors.

Simple RNN Layer: The RNN layer was used for sequence modeling.

Dense Layer: The output layer with a softmax activation function to classify text into one of the three semantic categories.

Model Training:

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

model.fit(X_train, y_train, epochs=10, batch_size=64, validation_split=0.2)
```

Epochs: The model was trained for 10 epochs.

Batch Size: A batch size of 64 was used for training.

Results:

Epoch 9/10:

- Training Loss: 0.2405
- Training Accuracy: 91.44%
- Validation Loss: 0.7637
- Validation Accuracy: 75.42%

Epoch 10/10:

- Training Loss: 0.2573
- Training Accuracy: 91.14%

- Validation Loss: 0.7794
- Validation Accuracy: 75.21%

These results indicate that the RNN model performed reasonably well in classifying text into semantic categories. The model achieved a validation accuracy of approximately **75%,** demonstrating its effectiveness for the task.

Challenges and Limitations:

Data Volume: The project started with a significant amount of data (27,480 samples) and was reduced to 23,340 samples. This data reduction may have affected the model's generalization ability.

Model Complexity: RNNs may struggle with capturing long-range dependencies in text data. More advanced models like LSTM or GRU could be explored for potentially improved performance.

Conclusion:

In conclusion, this project successfully demonstrated the application of RNNs for text semantic analysis. The trained model achieved a validation accuracy of approximately **75%**, indicating its effectiveness in classifying text into semantic categories. Challenges related to data volume and model complexity were encountered, and future work could involve further experimentation with advanced recurrent neural network architectures and optimization techniques to improve accuracy.

Text Semantic Analysis using Neural Networks (LSTM)

Same Preprocessing Steps as RNN

Model Architecture:

```
model = Sequential()

model.add(Embedding(input_dim=23340, output_dim=100, input_length=100))

model.add(LSTM(128,return_sequences=True))

model.add(LSTM(8))
```

model.add(Dense(3, activation='softmax'))

Model Training:

model.compile(optimizer='rmsprop',loss='sparse_categorical_crossentropy',metrics=['acc'])

history = model.fit(X_train, y_train,epochs=10,batch_size=64,validation_split=0.2)

Epochs: The model was trained for 10 epochs.

Batch Size: A batch size of 64 was used for training.

Results:

Epoch 9/10

234/234 [==============================] - 73s 311ms/step - loss: 0.2869 - acc: 0.9088 - val_loss: 0.5724 - val_acc: 0.7898

Epoch 10/10

234/234 [==============================] - 75s 321ms/step - loss: 0.2584 - acc: 0.9168 - val_loss: 0.5802 - val_acc: 0.7952

These results indicate that the LSTM model performed reasonably well in classifying text into semantic categories. The model achieved a validation accuracy of approximately **79.5%,** demonstrating its effectiveness for the task.