

Sentiment Analysis Model Evaluation Report

Data Preprocessing Steps:

1. Reading Data: The dataset is read from a CSV file using Pandas.
2. Cleaning Text: Regular expressions are used to remove non-alphanumeric characters, punctuation, and extra spaces. The text is also converted to lowercase.
3. Tokenization: NLTK's word_tokenize function is used to tokenize the cleaned text into individual words.
4. Removing Stopwords: NLTK's stopwords list is used to remove common English stopwords from the tokenized text.
5. Label Encoding: Labels are encoded using Scikit-learn's LabelEncoder to convert string labels into numerical values.

Model Architecture and Parameters:

- Tokenization and Padding: The text data is tokenized and padded to ensure uniform length using Keras' Tokenizer and pad_sequences.
- Embedding Layer: An embedding layer is used to convert text tokens into dense vectors of fixed size.
- Flatten Layer: The output of the embedding layer is flattened to be fed into a densely connected neural network.
- Dense Layers: Two dense layers with ReLU activation functions are used for classification, followed by a final output layer with softmax activation for multi-class classification.
- Optimizer: Adam optimizer is used.
- Loss Function: Sparse categorical cross entropy is used as the loss function.
- Metrics: Accuracy is used as the evaluation metric.

Training Process and Hyperparameters:

- Splitting Data: The data is split into training and testing sets using Scikit-learn's train_test_split.
- Training: The model is trained on the training data with a batch size of 32 and for 15 epochs.
- Validation Split: 20% of the training data is used for validation during training.

Evaluation Results and Analysis:

- Accuracy: The model achieves an accuracy of 58.42% on the test set.
- Precision, Recall, and F1-score: Precision, recall, and F1-score are reported for each class (negative, neutral, and positive sentiment). The model performs well in identifying negative sentiment but struggles with neutral sentiment due to zero precision, recall, and F1-score. Positive sentiment classification also shows room for improvement, especially in recall.
- Support: The number of instances for each class in the test set is provided.
- Analysis: The strengths and weaknesses of the model are discussed based on the evaluation results. The model shows decent performance in identifying negative sentiment but lacks accuracy in classifying neutral sentiment, possibly due to class imbalance or insufficient representation in the training data.

Overall, the model demonstrates some capability in sentiment analysis but requires further optimization, especially in handling neutral sentiment and improving overall classification performance.