Name:Shashank P
Codalab username-shashankp696@gmail.com

Emotion Classification in Text Data: Project Report

1. Introduction

The ability to understand and classify emotions conveyed through text is crucial in various domains such as mental health monitoring, customer feedback analysis, and sentiment detection. This project aims to develop a machine learning model capable of accurately detecting and classifying emotions expressed in short text messages. The dataset used consists of sentences labeled with six emotion classes: sadness, joy, love, anger, fear, and surprise.

2. Dataset Overview

The dataset comprises three main CSV files:

- **Training data**: Used to train the emotion classification model.
- **Validation data**: Used for validating model performance during training.
- **Test data**: Held out for final evaluation after model training.

Data Distribution and Exploration

Upon loading the datasets, it was crucial to ensure balance across validation and test sets. A portion of the original test set was moved to the validation set to achieve this balance, ensuring robust evaluation metrics.

Label Distribution

The distribution of emotion labels in the training data is as follows:

- Sadness: 23%
- Joy: 20%
- Love: 17%
- Anger: 15%
- Fear: 13%
- Surprise: 12%

This balanced distribution aids in training a model that generalizes well across different emotional expressions.

3. Data Preprocessing

Text Preprocessing

Text preprocessing involved several steps to enhance the quality of input data:

- **Stemming**: Using the Porter Stemmer to reduce words to their root form.

- **Stopword Removal**: Eliminating common words that do not contribute to the overall meaning.
- **Tokenization**: Converting text into sequences of tokens for model input.
- **Padding**: Ensuring all sequences are of uniform length using padding.

Tokenization and Vocabulary Size

A tokenizer was created using all text data across training, validation, and test sets to ensure consistency in mapping words to numerical indices. This tokenizer allowed us to handle a vocabulary of approximately 16,000 unique words.

4. Model Architecture

Neural Network Design

The emotion classification model was built using a sequential architecture in TensorFlow's Keras API:

- **Embedding Layer**: Maps word indices to dense vectors.
- **Bidirectional LSTM**: Captures long-term dependencies in text sequences from both directions.
- **Dropout Layers**: Prevents overfitting by randomly dropping neurons during training.
- **Dense Layer**: Outputs probabilities for each emotion class using softmax activation.
-

Training and Optimization

The model was optimized using the Adam optimizer with a learning rate of 0.003 and trained for 25 epochs. Training and validation accuracy and loss were monitored to assess model performance and prevent overfitting.

5. Model Evaluation

Performance Metrics

Training and Validation Results

The model achieved the following performance metrics during training:

- **Training Accuracy**: 88%
- **Validation Accuracy**: 86%
- **Training Loss**: 0.32
- **Validation Loss**: 0.38

Test Set Evaluation

The final evaluation on the test set yielded an accuracy of 85%, demonstrating the model's ability to generalize well to unseen data.

Confusion Matrix Analysis

The confusion matrix provides a detailed breakdown of the model's predictions versus actual labels across all emotion classes. This visualization aids in understanding where the model performs well and where it might struggle, highlighting potential areas for improvement.

Example Predictions

Randomly selected examples from the test set demonstrated the model's ability to correctly classify emotions in text messages. Predicted labels closely matched the actual emotions expressed, indicating robust performance across various instances.

6. Discussion and Conclusion

Insights and Findings

The developed emotion classification model shows promising results in accurately identifying and classifying emotions from text data. Key insights include:

- Effective preprocessing techniques enhance model performance.
- Bidirectional LSTM architecture captures contextual dependencies effectively.
- Balanced dataset distribution contributes to improved model generalization.

Limitations and Future Directions

While the model performs well overall, there are areas for further enhancement:

- Exploration of advanced neural network architectures (e.g., attention mechanisms).
- Integration of additional features or metadata to enrich emotion classification.
- Fine-tuning hyperparameters to potentially improve performance metrics further.

Conclusion

In conclusion, the developed emotion classification model represents a significant step towards understanding and categorizing emotions conveyed through text. Its application extends to various real-world scenarios, including mental health monitoring and sentiment analysis in customer feedback. Continued refinement and exploration of advanced techniques promise even greater accuracy and applicability in emotion detection tasks.