ML HACKATHON

Sentiment Analysis with Text and Video Fusion Model

Team Name: PESU_RR_10_499_513_543_528

Team Members:

SRN	NAME
PES1UG22CS499	Sadhana Hegde
PES1UG22CS513	Samiksha M
PES1UG22CS528	Sanjana Rathish
PES1UG22CS543	Sharanya G S

1. Project Overview

This project focuses on performing sentiment analysis using multimodal data, combining text and video features to classify sentiments. Using early and late fusion techniques, it integrates text-based and video-based models to enhance classification accuracy.

2. Data Preprocessing Methods

Text Preprocessing:

- **Stopwords Removal**: A minimal list of stopwords is used to filter out non-informative words.
- **Contraction Expansion**: A predefined dictionary is used to expand common contractions (e.g., "don't" to "do not").
- **Tokenization & Lowercasing**: Text is tokenized, lowercased, and stripped of punctuation.
- **Feature Extraction (TF-IDF)**: The preprocessed text is transformed into a TF-IDF feature matrix with a maximum of 5000 features.

Video Preprocessing:

- **Frame Extraction**: Frames are extracted from each video using OpenCV.
- **Resizing and Grayscaling**: Each frame is resized to 48x48 and converted to grayscale.
- **Padding**: For videos with fewer than 30 frames, zero-padding is used to maintain a consistent input shape.

3. Model Architectures

Text Model:

- **Random Forest Classifier**: A random forest with 100 estimators is used to classify sentiments from TF-IDF features. Probabilities from this model serve as input features for the fusion model.

Video Model (CNN):

- **CNN Layers**: The video model uses a TimeDistributed CNN architecture with layers to extract spatial features from each frame, followed by:
  - **Convolution and Pooling Layers**: 3x3 Conv2D layers with ReLU activation, followed by MaxPooling layers.
  - **Flattening and Pooling**: TimeDistributed Flatten and Global Average Pooling layers.
  - **Dense Layer**: A final dense layer of 128 units with ReLU activation.

4. Model Fusion and Training Strategy

Fusion Model:

- **Early Fusion (Concatenation)**: Video and text features are concatenated after independent processing, with:
  - **Dense Layers**: Combined dense layers with ReLU activation and dropout for regularization.
  - **Output Layer**: Softmax output with three units (positive, neutral, negative).

Training:

- **Optimizer and Loss Function**: The fusion model is compiled using the Adam optimizer with categorical cross-entropy loss.
- **Training Process**: Both video and text features are trained together with a batch size of 8 over 10 epochs.

5. Output and Results

- **Model Evaluation**: The trained model's performance on the test data was evaluated, providing accuracy and loss metrics.
- **Predictions**: Predicted sentiments are saved in a CSV file (fusion_predictions.csv), providing each utterance with its predicted sentiment.

6. Code Snippets

Text Preprocessing Example:

```python
def preprocess_text(text):
    text = expand_contractions(text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.lower().strip()
    tokens = text.split()
    tokens = [word for word in tokens if word not in stop_words]
    return ' '.join(tokens)
```

Video Model Creation:

```python
def create_video_model(input_shape=(30, 48, 48, 1)):
    video_input = Input(shape=input_shape)
    x = TimeDistributed(Conv2D(32, (3, 3), activation='relu'))(video_input)
    x = TimeDistributed(MaxPooling2D((2, 2)))(x)
    x = TimeDistributed(Conv2D(64, (3, 3), activation='relu'))(x)
    x = TimeDistributed(MaxPooling2D((2, 2)))(x)
    x = TimeDistributed(Flatten())(x)
    x = GlobalAveragePooling1D()(x)
    x = Dense(128, activation='relu')(x)
    x = Dropout(0.5)(x)
    return Model(inputs=video_input, outputs=x)
```

7. Interfaces and Deliverables

- **Model Outputs**: The final trained fusion model provides sentiment predictions as probabilities, formatted into a CSV output.
- **File Naming Convention**:
    - **Text and Video Data**: Consistently organized using Dialogue_ID and Utterance_ID (e.g., dia1_utt1.mp4).
    - **Output File**: Sentiment predictions stored in submission.csv.