Approach

Text Preprocessing

Performed text preprocessing by

1. Removing entity tags
2. Removing newline occurrences
3. Removing punctuations
4. Demojizing emoticons
5. Converting all text to lowercase

Visualizations

We observed the distribution of each emotion with the help of pie charts.

Count of Train Data :

['anger','anticipation','disgust','fear','joy','love','optimism','pessimism','sadness','surprise','trust',’neutral’]
[2859, 1102, 2921, 1363, 2877, 832, 2291, 895, 2273, 396, 400, 218]

Count of Predicted Data :

['anger','anticipation','disgust','fear','joy','love','optimism','pessimism','sadness','surprise','trust','neutral']
[156, 56, 120, 33, 199, 32, 157, 8, 59, 8, 20, 3]

Word Embedding used

After this, we employed a normalized word embedding of 128 dimensions that was pretrained on the polarity classification task available on the TensorFlow hub. Initially, we also tried to apply 512-dimensional word embedding. However, this resulted in a decreasing accuracy over epochs accompanied by longer training time.

Embedding used: https://tfhub.dev/google/nnlm-en-dim128-with-normalization/2

Training

We initially deployed a Bidirectional LSTM model on the training data. However, the overall model performance was poor. Hence, we decided to employ a 3-layer deep neural network with 12, 12, and 10 neurons respectively (all with "RELU" activation functions) and an additional output layer of 12 neurons with a "sigmoid" activation layer for outputting the final result vector.

Training arguments

3-layer deep neural network with 12, 12, and 10 neurons respectively (all with "RELU" activation functions)
Optimizer: Adam
Learning Rate: 0.01
Loss: Binary Crossentropy

Metrics: Accuracy
Epochs: 20
Batch size: 32

The best accuracy obtained on the training set was 67.10%

Testing

We then moved on to check the performance of our model on the test set. The model predicts the outputs in terms of logits which needs to be converted into human interpretable form. Therefore, we set a threshold of 0.50 in order to classify a sample to a particular emotion. Samples that have a score of more than 0.50 for emotion are classified into that emotion with the help of the map function.

Tools Used

Language: Python

Libraries:
TensorFlow
Keras
Numpy
Pandas
Matplotlib
Regex