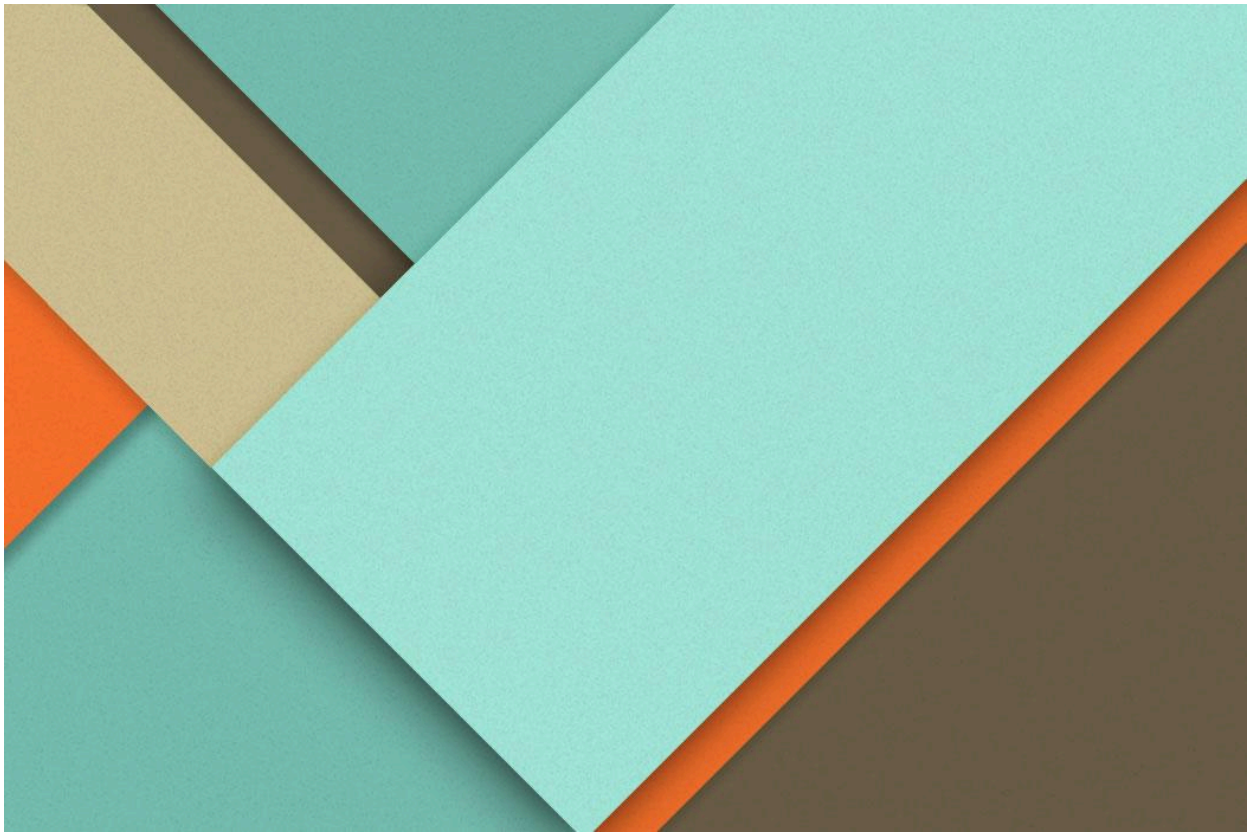


Emotion Intensity Detection Task Paper

15.03.2024

Setlem Nikhith
Mechanical Engineering
National Institute of Technology Karnataka,Surathkal

Overview

This task addresses the limitations of existing emotion datasets by focusing on determining the intensity or degree of a specific emotion expressed in tweets. Unlike traditional datasets which annotate emotions categorically, this task introduces real-valued scores between 0 and 1 to represent the degree of emotion felt by the speaker. The goal is to provide systems with the ability to automatically gauge the intensity of emotions in text, which is crucial for various applications. The task involves assessing the intensity of a given emotion X in a tweet, with a score of 1 indicating maximum emotion and 0 indicating minimum emotion. These scores serve as a measure to compare the degree of emotion expressed across different instances, rather than having inherent meaning on their own.

Objectives

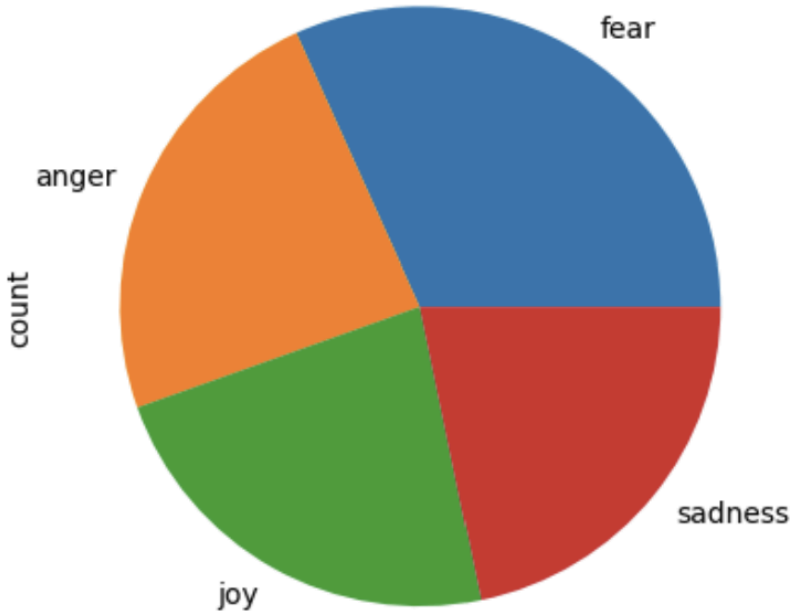
1. Develop a purely statistical model using various machine learning methods to predict emotion intensity in tweets.
2. Implement a deep learning model, experimenting with pre-trained embeddings to accurately determine emotion intensity in tweets.

Dataset

Dataset contains 3 attributes.

- 1st attribute as index
- 2nd attribute contains sentences expressing emotions and feelings.
- 3rd attribute contains the labels from range 0 to 1 which indicates the intensity or category of the emotion expressed in the corresponding text.

Dataset Distribution is like below shown



Methodology

I. Statistical model

Initially, the training dataset was read from each individual CSV file of emotions into a pandas DataFrame and later in runtime I concatenated the data of each emotion in training as shown in Fig 1 . Similarly, the testing validation dataset was also read. Next, the training data was shuffled to prevent any implicit order or patterns from affecting the training process, ensuring that the model learns more generalized patterns. The textual data was then converted into numerical features using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique.

A Multinomial Naive Bayes classifier object was instantiated for further analysis. Multinomial Naive Bayes is a variant of the Naive Bayes algorithm suitable for classification tasks with discrete features, such as word counts or TF-IDF values. With the dataset prepared, experiments were conducted on different classification models to identify the most suitable one, aiming for the highest accuracy on testing and validation.

The shuffled dataset was then trained with the Multinomial Naive Bayes classifier model. Similarly, the shuffled dataset was trained with the Random Forest Model, known for its power and flexibility, especially suitable for classification tasks, and typically providing high accuracy while handling high-dimensional data well. Additionally, training was conducted with the KNN classifier, a simple yet effective classification algorithm that classifies data points based on the majority class among their k nearest neighbors. This final best model achieved an accuracy of 89% when validated and tested with the testing and validation datasets, with Random Forest respectively.

Based on the analysis, it became evident that the Random Forest outperformed other models by a significant margin. Thus, efforts were directed towards tuning the hyperparameters to enhance its accuracy, aiming for an improved version beyond the current performance level

.

II. Deep learning model

The methodology used employs a deep learning approach, specifically Long Short-Term Memory (LSTM) neural networks for text categorization tasks. Initially, text is preprocessed to ensure compliance with the neural network design. To aid efficient calculation, the text is tokenized, which involves converting words into numerical indices and padding or truncating sequences to a uniform length. Furthermore, the categorical labels are converted into one-hot encoded vectors, allowing classification. The dataset is then divided into training and testing subsets to assess the model's performance. The LSTM model architecture is built, with layers such as embedding, LSTM, spatial dropout, and dense layers that use appropriate activation functions.

I have implemented a Long Short-Term Memory (LSTM) neural network model for text classification. The model starts with an embedding layer to convert input text into dense vectors of fixed size. It then applies spatial dropout to reduce overfitting, followed by an LSTM layer with 196 units and dropout to capture long-term dependencies in the sequence data. Next, a dense layer with 64 units and ReLU activation function is added for feature extraction, followed by another dense layer with softmax activation for multi-class classification. The model is compiled with categorical cross-entropy loss function, Adam optimizer, and accuracy metric for training.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 50, 128)           128000

 spatial_dropout1d_2 (Spati  (None, 50, 128)           0
 alDropout1D)

 lstm_2 (LSTM)               (None, 196)               254800

 dense_4 (Dense)             (None, 64)                12608

 dense_5 (Dense)             (None, 4)                 260

=================================================================
Total params: 395668 (1.51 MB)
Trainable params: 395668 (1.51 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Results

I. Naives Bayes model

This model gives an accuracy of 76% after testing and validating with testing and validation dataset respectively.

```
Validation Accuracy: 0.80
Validation Classification Report:
              precision    recall  f1-score   support

       anger       0.80      0.73      0.76        84
        fear       0.72      0.89      0.79       110
         joy       0.89      0.80      0.84        79
     sadness       0.86      0.73      0.79        74

    accuracy                           0.80       347
   macro avg       0.82      0.79      0.80       347
weighted avg       0.81      0.80      0.80       347

Test Accuracy: 0.76
Test Classification Report:
              precision    recall  f1-score   support

       anger       0.83      0.71      0.76       760
        fear       0.66      0.88      0.76       995
         joy       0.85      0.76      0.80       714
     sadness       0.80      0.62      0.70       673

    accuracy                           0.76      3142
   macro avg       0.78      0.74      0.76      3142
weighted avg       0.77      0.76      0.76      3142
```

II. Random Forest model

This model increased the accuracy to 82% when validated and tested with testing and validation dataset respectively.

```
Validation Accuracy: 0.84
Validation Classification Report:
              precision    recall  f1-score   support

       anger       0.90      0.76      0.83        84
        fear       0.78      0.89      0.83       110
         joy       0.94      0.84      0.89        79
     sadness       0.77      0.84      0.80        74

    accuracy                           0.84       347
   macro avg       0.85      0.83      0.84       347
weighted avg       0.84      0.84      0.84       347

Test Accuracy: 0.82
Test Classification Report:
              precision    recall  f1-score   support

       anger       0.90      0.76      0.83       760
        fear       0.76      0.86      0.81       995
         joy       0.91      0.85      0.88       714
     sadness       0.77      0.79      0.78       673

    accuracy                           0.82      3142
   macro avg       0.83      0.82      0.82      3142
weighted avg       0.83      0.82      0.82      3142
```

III. KNN model

This model gave the accuracy of 64% when validated and tested with testing and validation dataset respectively.

```
Validation Accuracy: 0.67
Validation Classification Report:
              precision    recall  f1-score   support

       anger       0.79      0.65      0.71        84
        fear       0.61      0.82      0.70       110
         joy       0.80      0.51      0.62        79
     sadness       0.58      0.62      0.60        74

    accuracy                           0.67       347
   macro avg       0.69      0.65      0.66       347
weighted avg       0.69      0.67      0.66       347

Test Accuracy: 0.64
Test Classification Report:
              precision    recall  f1-score   support

       anger       0.69      0.61      0.65       760
        fear       0.59      0.76      0.66       995
         joy       0.79      0.50      0.62       714
     sadness       0.60      0.64      0.62       673

    accuracy                           0.64      3142
   macro avg       0.67      0.63      0.64      3142
weighted avg       0.66      0.64      0.64      3142
```

IV. Deep Learning model

This model gave the accuracy of 25% when validated and tested with testing and validation dataset respectively.

```
99/99 [==============================] - 3s 29ms/step
Accuracy: 0.25
              precision    recall  f1-score   support

           0       0.22      0.19      0.21       760
           1       0.32      0.29      0.30       995
           2       0.20      0.26      0.23       714
           3       0.24      0.23      0.23       673

    accuracy                           0.25      3142
   macro avg       0.25      0.24      0.24      3142
weighted avg       0.25      0.25      0.25      3142

99/99 [==============================] - 3s 31ms/step
Accuracy: 0.25
              precision    recall  f1-score   support

           0       0.22      0.19      0.21       760
           1       0.32      0.29      0.30       995
           2       0.20      0.26      0.23       714
           3       0.24      0.23      0.23       673

    accuracy                           0.25      3142
   macro avg       0.25      0.24      0.24      3142
weighted avg       0.25      0.25      0.25      3142

Accuracy: 0.25
```