EMOTION DETECTION IN TEXT.

Srividhya Pasam
Raghuvamshi batini
Srinaga Vishnu devineni

INTRODUCTION

- The goal of the proposed project is to create an advanced natural language processing (NLP) system that can reliably recognize and categorize emotions in textual data, with potential applications in a wide range of fields.
- The practical applications are numerous: in mental health, it provides tools for tracking emotional well-being through textual analysis; in customer service, it can improve chatbots and support systems to react more sympathetically to user moods; and in social media analytics, it
- Through enhancing the precision and profundity of emotion identification, this study offers broad public sentiment insights.

DATA PREPARATION & PROCESSING

Data Preparation

- The dataset **tweet_emotions** is loaded into a pandas DataFrame, containing numerous tweets with associated emotions
- The dataset is split into training (80%), validation (16%), and test (4%) sets. This is done to ensure that the model is trained on a large portion of the data, validated on a smaller portion, and finally tested on a separate set to evaluate its performanc

Preprocessing

- The content from the training data is tokenized. This process involves converting the text into a list, removing punctuation, and then splitting it into individual words or tokens. The **Tokenizer** from Keras preprocessing is used for this purpose. It helps in converting the text data into sequences of integers, where each integer represents a specific word in a dictionary

MODEL ANALYSIS

Architecture and Training

- A custom TensorFlow Keras layer, **T_encoder**, is defined, which includes multiple layers such as MultiHeadAttention, Dense, Dropout, and LayerNormalization. This architecture is indicative of a Transformer model, which is effective for handling sequence data like text.

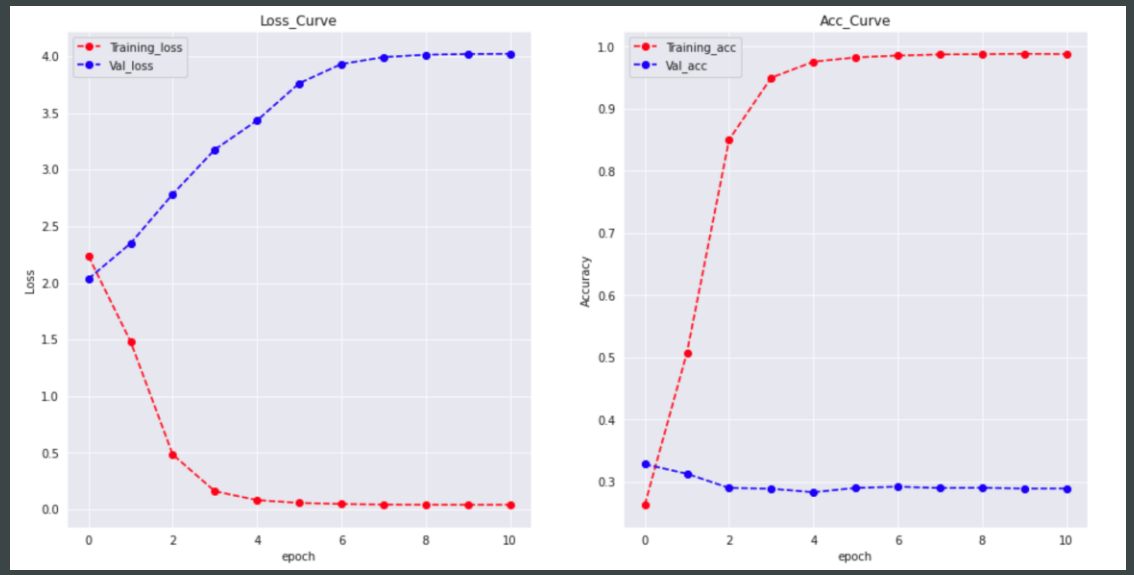
- Learning rate schedulers and early stopping callbacks are used during training. The learning rate scheduler adjusts the learning rate throughout training, potentially improving model performance by fine-tuning the learning process. Early stopping is used to halt the training if the model's performance on the validation data doesn't improve, preventing overfitting

MODEL ANALYSIS

Fitting and Evaluation

- The model, named **model_transformer**, is trained on the tokenized training data (**x_train** and **y_train**) for 25 epochs, with validation data provided for performance monitoring. This process involves feeding the input data into the model, which learns to predict the emotion associated with each tweet
- After training, the model is evaluated on the test set (**x_test** and **y_test**). Performance metrics like the confusion matrix, F1 score, precision, and recall are calculated. These metrics provide insights into the model's ability to correctly classify emotions in tweets, with a focus on both the accuracy and balance of predictions across different emotion classe

RESULTS



- The accuracy and loss curves for both the losses from training and validation are plotted as shown above.
- And the respective calculated metrics are:

f1_score: 0.3053817271589487

Precsion: 0.3053817271589487

Recall: 0.3053817271589487

- And with the inference test the respective results from the model are shown here.

```
--------------------------------------------------
Input_Sentence:   happy mothers day to all moms
--------------------------------------------------
Actual_emotion:  neutral
--------------------------------------------------
Predicted_emotion:  neutral
--------------------------------------------------

--------------------------------------------------
Input_Sentence:   somehow i cant reply to your message lol and yes i know thank youuu
--------------------------------------------------
Actual_emotion:  happiness
--------------------------------------------------
Predicted_emotion:  love
--------------------------------------------------

--------------------------------------------------
Input_Sentence:   hi fellow
--------------------------------------------------
Actual_emotion:  neutral
--------------------------------------------------
Predicted_emotion:  neutral
--------------------------------------------------
```

THANK YOU