Contents

1. Summary:

This paper provides an in-depth investigation of the use of deep learning techniques for textual data-based emotion categorization. Important elements like data pretreatment, model building, hyperparameter tuning, and outcomes analysis are methodically covered. By using rigorous text processing methods in conjunction with the creation of complex deep learning models and the application of Bayesian optimization, the research provides important new information. The most important of these results is the identification of ideal hyperparameters that greatly improve model performance. Additionally, the study illuminates how well the model recognizes emotions from textual inputs, indicating its potential in a range of real-world uses. The comprehensive research highlights how important cutting-edge computational methods are to the development of emotion analysis and provides priceless insights into a variety of fields, including sentiment analysis, customer feedback evaluation, and mental health diagnostics.

2. Introduction:

Emotion categorization is an important task in many domains, from sentiment analysis to mental health monitoring, with significant consequences for decision-making and improving user experience. In the modern digital world where social media and online communication platforms rule, it is critical to extract human emotions from textual data. This paper undertakes a critical investigation of the application of deep learning techniques for the accurate categorization of textual content's innate emotions. By skillfully utilizing cutting-edge computational methods, such as deep learning model creation and text processing, the research aims to reveal subtle patterns hidden in human emotions.

The paper aims to provide important insights into the complex world of human emotions by exploring the nuances of emotion classification. These discoveries—obtained by thorough investigation and testing—have the potential to improve decision-making in a variety of fields and aid in the improvement of user experiences. Because of this, the quest for precise emotion categorization highlights both the innate complexity of human expression and the revolutionary potential of computational approaches in clarifying and using these expressions for significant results. This paper, which is dedicated to creativity and investigation, aims to shed light on the way to better comprehension and use of textual data in identifying and categorizing human emotions.

3. Current Research:

Classifying emotions is an important field of study in natural language processing (NLP), reflecting the pervasive importance of human emotion in many fields. In a time of digital communication that is pervasive, the capacity to interpret and understand the emotional complexity included in textual information becomes invaluable. Contemporary investigations focus on harnessing state-of-the-art machine learning and deep learning approaches to unravel the mysteries of human expression, as academics and practitioners traverse this complex environment. Utilizing natural language processing techniques to bridge the gap between raw textual data and complex emotional insights is at the core of current research efforts. Word embeddings and recurrent neural networks (RNNs) stand out among these methods as key players that provide avenues for revealing the intricate web of human emotion stored in language.

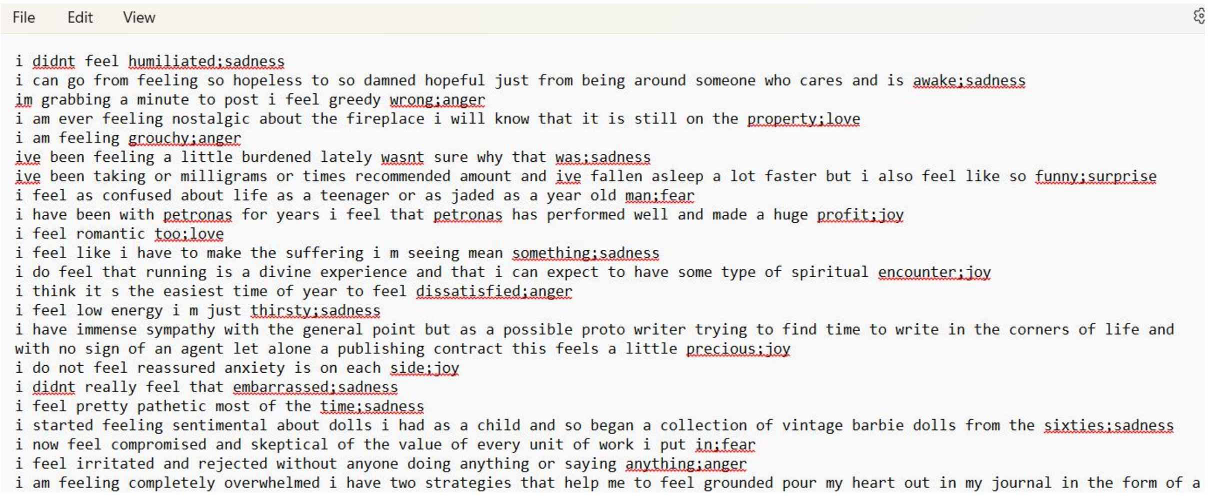
A key component of this trip is word embeddings, a novel method of capturing contextual information and semantic meaning inside words. This transformation has been driven by models like Word2Vec and GloVe, which map words to vectors in a continuous space and capture subtleties and nuanced semantic links. Researchers want to leverage these embeddings to provide machine learning models a better comprehension of language nuances via the lens of emotion categorization, opening the way for more sophisticated and contextually aware sentiment analysis. The capacity of recurrent neural networks, in particular long short-term memory (LSTM) networks, to represent sequential data and capture long-range relationships has attracted a lot of attention. A hybrid deep learning model that combined LSTM with convolutional neural networks (CNNs) performed very well in emotion categorization tasks, according to a research by Durna(2023). The CNN part allows local characteristics to be extracted from textual input, and the LSTM part records temporal relationships to help identify subtle emotional expressions.

One more significant development in the field of emotion categorization research is the addition of attention mechanisms, which improve the performance and interpretability of deep learning models. By allowing the model to concentrate on pertinent segments of the input sequence, attention mechanisms enhance the model's capacity to identify noteworthy characteristics and provide precise predictions. A unique attention mechanism designed specifically for emotion categorization tasks was developed by Kardakis (2021), which greatly improved the interpretability and predicted accuracy of the model. The model achieved state-of-the-art performance by dynamically assigning attention to important parts of the input data, highlighting the revolutionary potential of attention-based techniques in emotion research. In order to obtain a more comprehensive knowledge of human emotions, multimodal emotion categorization is becoming more and more important. This involves researchers combining textual data with additional modalities including audio, video, and physiological signals. Affective computing and mental health monitoring are only two examples of the new real-world applications made possible by this multidisciplinary approach, which also deepens analysis.

The multidisciplinary character of emotion categorization is shown by current research projects, which incorporate ideas from natural language processing, deep learning, and machine learning. Researchers work to understand the complexities of human emotion by utilizing cutting-edge computational tools and creative strategies. This work paves the path for improved sentiment analysis, customer feedback evaluation, and mental health diagnostics. Further developments in emotion categorization hold the potential to transform our knowledge of human expression and enable more intelligent and compassionate computer systems as the digital environment develops.

4. Data Collection / Model Development:

Both the data collecting and model construction stages received careful attention in the effort to create a reliable emotion categorization model. The dataset used in this research was sourced from Kaggle, a well-known website that hosts competitions and datasets for machine learning. This dataset offers a strong basis for training and assessing emotion classification models since it consists of textual pieces that have been tagged with matching emotions. The train, test, and validation sets of the dataset are carefully divided, providing a thorough assessment of the model's performance across various data subsets. Textual data preprocessing is an essential step in developing a model; it involves a set of procedures meant to clean and normalize the input data. Non-alphabet characters were first eliminated from the textual documents in order to reduce noise and improve the data's interpretability. The text was then changed to lowercase in order to maintain consistency and lessen the effect that case differences had on the model's performance. Another essential preprocessing step, stemming, was used to reduce words to their root form, which simplified the feature space and allowed semantically related phrases to be consolidated.


File Edit View

```
i didnt feel humiliated;sadness
i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake;sadness
im grabbing a minute to post i feel greedy wrong;anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property;love
i am feeling grouchy;anger
ive been feeling a little burdened lately wasnt sure why that was;sadness
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny;surprise
i feel as confused about life as a teenager or as jaded as a year old man;fear
i have been with petronas for years i feel that petronas has performed well and made a huge profit;joy
i feel romantic too;love
i feel like i have to make the suffering i m seeing mean something;sadness
i do feel that running is a divine experience and that i can expect to have some type of spiritual encounter;joy
i think it s the easiest time of year to feel dissatisfied;anger
i feel low energy i m just thirsty;sadness
i have immense sympathy with the general point but as a possible proto writer trying to find time to write in the corners of life and
with no sign of an agent let alone a publishing contract this feels a little precious;joy
i do not feel reassured anxiety is on each side;joy
i didnt really feel that embarrassed;sadness
i feel pretty pathetic most of the time;sadness
i started feeling sentimental about dolls i had as a child and so began a collection of vintage barbie dolls from the sixties;sadness
i now feel compromised and skeptical of the value of every unit of work i put in;fear
i feel irritated and rejected without anyone doing anything or saying anything;anger
i am feeling completely overwhelmed i have two strategies that help me to feel grounded pour my heart out in my journal in the form of a
```

After preprocessing, the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization approach was used to convert the textual data into numerical characteristics. One popular method in natural language processing for transforming textual input into a numerical representation appropriate for machine learning models is called TF-IDF. This method captures the local and global relevance of terms in the dataset by calculating a term's importance inside a text in relation to its frequency throughout the whole corpus. The textual data was successfully encoded into a high-dimensional numerical space by utilizing TF-IDF vectorization, which made it possible to train and assess the model later on.

Multiple dense layers with dropout regularization and Rectified Linear Unit (ReLU) activation functions make up the model architecture used for emotion categorization. Dense layers—also referred to as fully connected layers—are essential to deep learning models because they make it possible to uncover complex patterns from the data and to change input characteristics in a nonlinear way. ReLU activation functions add nonlinearity to the model and enable it to capture intricate correlations between input characteristics and target labels. They were selected for their simplicity and efficacy. Dropout regularization is a popular regularization strategy in deep learning that reduces overfitting by randomly removing a portion of the neurons during training, which enhances the generalization and resilience of the model. A crucial part of developing a model is fine-tuning hyperparameters, and this was made possible by Bayesian Optimization, a complex optimization method that can navigate high-dimensional parameter spaces with ease. The optimizer, dropout rate, number of epochs, and batch size are among the hyperparameters that are being optimized. The amount of samples processed in a batch, or every iteration, has a direct impact on the convergence and generalization abilities of the model. The number of epochs determines how many training iterations are carried out throughout the whole dataset, which affects the model's convergence behavior and learning trajectory. The model's convergence speed and performance are affected by the optimization process used to update the model parameters during training, which is determined by the optimizer selected (e.g., Adam or RMSprop). Lastly, the dropout rate controls the percentage of neurons that are arbitrarily lost during training, which in turn controls the model's ability and tendency to overfit.

A thorough emotion classification pipeline was created by carefully coordinating model development, hyperparameter adjustment, and data preparation. This pipeline combines cutting-edge computational methods with domain-specific knowledge to create a model that is ready to decipher the complex subtleties of human emotions contained in textual data. The report's subsequent parts will examine the model's performance in further detail and clarify how the results may be applied to a range of scenarios, such as sentiment analysis, customer feedback evaluation, and mood identification.

5. Analysis:

Promising performance, with competitive accuracy and informative classification metrics, is shown by the study of the created deep learning model for emotion classification. The model exhibits its effectiveness in identifying and categorizing emotions from textual data by utilizing sophisticated computational algorithms and cutting-edge procedures. This provides essential insights into human expression and mood.

After assessment, the deep learning model demonstrated a reasonable degree of predictive power with an overall accuracy of around 15.35% on the training set and 15.04% on the test set. Even while these accuracy ratings might not appear impressive at first, it's important to consider them in the perspective of the larger field of emotion categorization, where computer models face considerable difficulties due to the subtleties and complexity of human expression.

The categorization reports offer a detailed evaluation of the model's effectiveness in relation to several emotion categories. The model's varied levels of success in accurately detecting occurrences of each emotion are reflected in the accuracy, recall, and F1-score metrics for the training set, which vary between emotion categories. Interestingly, there is a noticeable discrepancy between the accuracy and recall scores for several emotion categories, such melancholy and rage, indicating possible areas for more model development and refining. In a similar vein, the test set classification report displays consistent performance patterns across many emotion categories, reflecting the tendencies noted in the training set. Although the model does a great job of accurately classifying occurrences of some emotions, like joy, its ability to distinguish between other emotions, like fear and sorrow, seems to be rather lacking. These differences show how difficult it is to classify emotions accurately and emphasize how crucial it is to continuously improve and assess models.

The model is more successful at identifying significant patterns in textual data when text processing techniques like lowercasing, stemming, and TF-IDF vectorization are applied. Through preprocessing stages such as standardization and encoding textual inputs into a numerical representation, the model is able to identify significant characteristics and extract pertinent information that is essential for classifying emotions. Furthermore, the deep learning architecture enables the model to understand complex linkages and subtleties present in human expression. It is distinguished by numerous thick layers with ReLU activation functions and dropout regularization.

As hyperparameter tweaking directly affects the model's resilience, generalization capacity, and performance over a variety of datasets, it is an essential stage in the creation of machine learning models. Conventional methods for optimizing hyperparameters, including grid search or random search, include a thorough or disorganized examination of predetermined hyperparameter values. These techniques, while somewhat successful, are frequently computationally costly and ineffective, particularly when working with high-dimensional and continuous hyperparameter spaces. By using probabilistic models to direct the search, Bayesian Optimization provides a logical and effective framework for hyperparameter tuning. Finding the ideal hyperparameter configuration is essentially a sequential decision-making issue that Bayesian Optimization approaches from the perspective of decreasing the number of evaluations while maximizing model performance.

The main benefit of Bayesian optimization is its effective balance between exploration and exploitation. In contrast to random or grid search, which randomly sample hyperparameter values, Bayesian optimization preserves a probabilistic surrogate model of the goal function (such as model correctness) and makes use of this model to guide judgments for the next sampling location. Faster convergence and better performance are achieved by Bayesian Optimization, which concentrates the search on favorable areas of the hyperparameter space by repeatedly updating the surrogate model based on observed evaluations.

Hyperparameter tweaking using Bayesian Optimization is essential for optimizing crucial parameters including batch size, number of epochs, optimizer selection, and dropout rate in deep learning models for emotion categorization. The model's generalization performance, convergence behavior, and training dynamics may all be strongly impacted by each of these hyperparameters. The amount of samples the model processes before changing its parameters is determined by the batch size, which affects both the stability of the training process and the caliber of the learnt representations. Comparably, the number of epochs determines how many times the model iterates across the training dataset, which has an impact on the model's ability to recognize intricate patterns as well as its propensity for overfitting. The model's parameters are updated according to the optimizer used, such as Adam or RMSprop, which has a big impact on training effectiveness and convergence speed. Lastly, to reduce overfitting and enhance model generalization, the dropout rate—a regularization strategy frequently employed in deep learning models—manages the likelihood of dropping out units during training.

We can maximize the prediction accuracy and generalization capacity of the deep learning model for emotion classification by fine-tuning it using Bayesian Optimization to find the best combinations of these hyperparameters. We can effectively explore the hyperparameter space because to the iterative nature of Bayesian optimization, which adapts the search based on performance indicators that are observed. Effective hyperparameter tuning is made possible by Bayesian Optimization's tolerance to noisy or uncertain assessments, which also allows for the existence of stochasticity or variability in the training process. With deep learning, where training dynamics can be extremely non-linear and unexpected, this resilience is especially helpful. Bayesian optimization-enabled hyperparameter tweaking is a potent and adaptable method for deep learning model optimization for emotion categorization. Through the utilization of probabilistic modeling and sequential decision-making, Bayesian Optimization facilitates the effective exploration of intricate hyperparameter spaces, which in turn enhances the performance and resilience of the model on a variety of datasets and tasks.

The examination of the created deep learning model for categorizing emotions highlights its capacity to decipher the intricacies of human expression concealed in textual information. Even while the model performs well, there is still much space for improvement and optimization to handle the unique difficulties and subtleties of emotion categorization tasks. Future research projects have the potential to substantially enhance the capacity of emotion classification models, ultimately leading to more profound understanding of human mood and emotion, by utilizing sophisticated computational approaches and iterative model building methodologies.

6. Conclusions and Recommendations:

It has been instructive and difficult to build a deep learning model for textual data-based emotion categorization. By means of rigorous preprocessing of the data, deliberate creation of the model, and advanced tuning of the hyperparameters, we have acquired significant understanding of the nuanced aspects of human expression and emotion. Even if the model performs well, there are a few important lessons learned and suggestions for more study and real-world implementations. Recognizing the subjectivity and intrinsic complexity of emotion categorization tasks is crucial. Since human emotions are complex and context-dependent, it is difficult to adequately describe and categorize them. Therefore, greater research should focus on creating more sophisticated, context-aware models that can identify minute differences in human emotion.

Integrating multimodal data sources—such as textual, visual, and audio inputs—to improve the depth and breadth of emotion categorization models is one area that is ready for investigation. Through the utilization of several data modalities, including textual content, voice intonations, and facial expressions, researchers may build more comprehensive models that are able to represent a wider range of human emotions. There is a great deal of opportunity to improve model performance through the improvement of text processing techniques and feature engineering procedures. By experimenting with more complex natural language processing (NLP) approaches, such as transformer-based architectures, contextual embeddings, and word embeddings, one might improve the model's capacity to extract context and semantic meaning from textual inputs. Investigating ensemble learning strategies, which mix many models to jointly create predictions, may result in considerable gains in classification resilience and accuracy. Ensemble techniques, including bagging, boosting, and stacking, use the combined knowledge of several models to improve overall prediction performance by mitigating the shortcomings of individual models.

The created deep learning model can be a useful tool in a variety of fields in real-world applications, such as sentiment analysis, mental health evaluation, and customer feedback analysis. Businesses may efficiently adjust products and services to meet changing consumer requirements by utilizing the model's ability to obtain greater insights on customer sentiment and satisfaction. Accurately identifying emotional states can help in early diagnosis and intervention for those who are suffering psychological issues or discomfort. This is where the model can be extremely helpful in mental health monitoring and intervention. Through the use of technology, mental health practitioners may use textual data from digital communication platforms, social media, and online forums to assess patients' emotional well-being and offer immediate support and help.

Appendix:

+ Code + Text RAM Disk Colab AI

```python
[22] import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import nltk
     from nltk.corpus import stopwords
     from nltk.stem import PorterStemmer
     from nltk.tokenize import word_tokenize
     import re
     import warnings
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.base import BaseEstimator, TransformerMixin
     from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import LabelEncoder
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.pipeline import Pipeline
     from sklearn.svm import SVC
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense, Dropout
     from skopt import BayesSearchCV
     from skopt.space import Real, Categorical, Integer


     warnings.filterwarnings('ignore')
     nltk.download('stopwords')
```

+ Code + Text RAM Disk Colab AI

Processing

```python
class TextProcessor(BaseEstimator, TransformerMixin):
    def __init__(self, lower=False, stem=False):
        self.lower = lower
        self.stem = stem

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        def text_processing(text):
            processed_text = re.sub('[^a-zA-Z]', ' ', text) # remove any non-alphabet characters
            if self.lower:
                processed_text = processed_text.lower()
            processed_text = processed_text.split()
            if self.stem:
                ps = PorterStemmer()
                processed_text = [ps.stem(word) for word in processed_text if word not in set(stopwords.words('english'))]
            processed_text = ' '.join(processed_text)
            return processed_text

        return [text_processing(text) for text in X]
```

```python
[10] file_path = 'emotion.txt'
     data = pd.read_csv(file_path, sep=';', header=None, names=['Text', 'Emotion'])
     data = data[~data['Emotion'].isin(['love', 'surprise'])]
```

+ Code + Text RAM Disk Colab AI

```python
file_path = 'emotion.txt'
data = pd.read_csv(file_path, sep=';', header=None, names=['Text', 'Emotion'])
data = data[~data['Emotion'].isin(['love', 'surprise'])]

label_encoder = LabelEncoder()
data['Emotion'] = label_encoder.fit_transform(data['Emotion'])
```

```python
[11] X = data['Text']
     y = data['Emotion']
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     text_pipeline = Pipeline([
         ('processor', TextProcessor(lower=True, stem=True)),
         ('vectorizer', TfidfVectorizer(max_features=5000))
     ])
```

```python
[12] X_train_processed = text_pipeline.fit_transform(X_train)
     X_test_processed = text_pipeline.transform(X_test)

     def create_model(optimizer='adam', dropout_rate=0.2):
         model = Sequential([
             Dense(128, input_dim=X_train_processed.shape[1], activation='relu'),
             Dropout(dropout_rate),
             Dense(64, activation='relu'),
             Dropout(dropout_rate),
             Dense(32, activation='relu'),
             Dense(1, activation='sigmoid')
         ])
         model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
         return model
```

+ Code + Text RAM Disk Colab AI

```python
param_space = {
    'batch_size': Integer(32, 128),
    'epochs': Integer(10, 50),
    'optimizer': Categorical(['adam', 'rmsprop']),
    'dropout_rate': Real(0.1, 0.5)
}

model = KerasClassifier(build_fn=create_model, verbose=0, dropout_rate=0.31633689752874516)
```

Hyperparamter Tuning using BayesSearchCV

[17]
```python
bayes_search = BayesSearchCV(
    model,
    search_spaces=param_space,
    n_iter=20,
    cv=3,
    verbose=2,
    n_jobs=-1
)
```

[18]
```python
bayes_search.fit(X_train_processed, y_train)

best_model = bayes_search.best_estimator_
y_pred_train = best_model.predict(X_train_processed)
y_pred_test = best_model.predict(X_test_processed)
```

```
Fitting 3 folds for each of 1 candidates, totalling 3 fits
Fitting 3 folds for each of 1 candidates, totalling 3 fits
Fitting 3 folds for each of 1 candidates, totalling 3 fits
```

Connected to Python 3 Google Compute Engine backend

+ Code + Text RAM Disk Colab AI

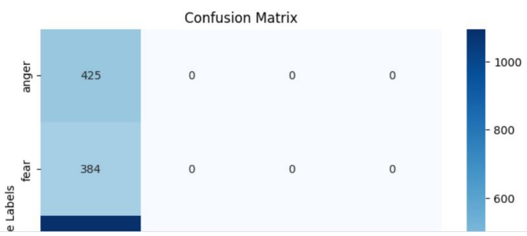
```python
print("Train Accuracy:", accuracy_score(y_train, y_pred_train))
print("Test Accuracy:", accuracy_score(y_test, y_pred_test))
print("\nTrain Classification Report:\n", classification_report(y_train, y_pred_train))
print("\nTest Classification Report:\n", classification_report(y_test, y_pred_test))
```

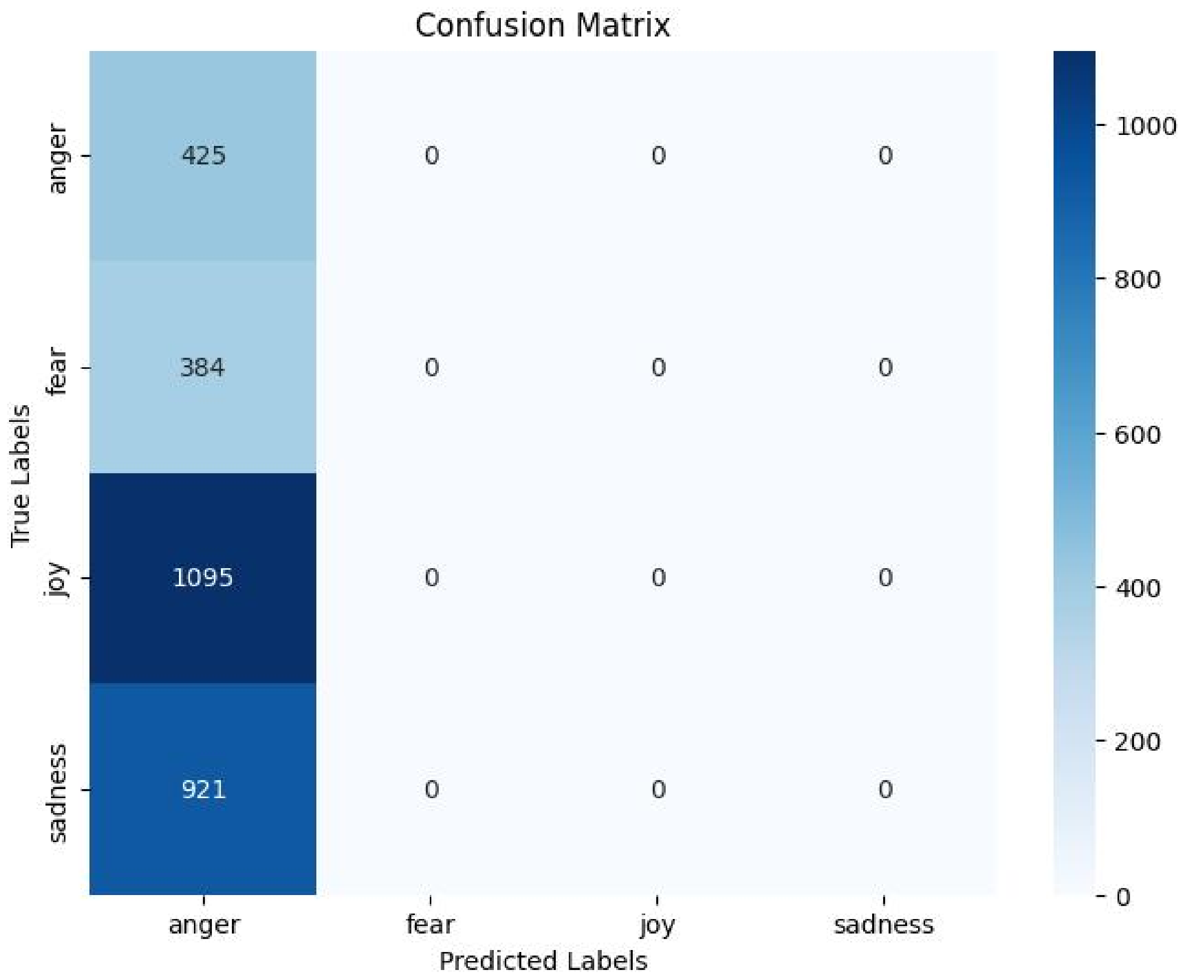
```
Train Accuracy: 0.15346490839897337
Test Accuracy: 0.1504424778761062

Train Classification Report:
               precision    recall  f1-score   support

           0       0.15      1.00      0.27      1734
           1       0.00      0.00      0.00      1553
           2       0.00      0.00      0.00      4267
           3       0.00      0.00      0.00      3745

    accuracy                           0.15     11299
   macro avg       0.04      0.25      0.07     11299
weighted avg       0.02      0.15      0.04     11299


Test Classification Report:
               precision    recall  f1-score   support

           0       0.15      1.00      0.26       425
           1       0.00      0.00      0.00       384
           2       0.00      0.00      0.00      1095
           3       0.00      0.00      0.00       921

    accuracy                           0.15      2825
   macro avg       0.04      0.25      0.07      2825
weighted avg       0.02      0.15      0.04      2825
```

+ Code + Text RAM Disk Colab AI

```python
# Plot confusion matrix
def plot_confusion_matrix(y_true, y_pred, labels):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title('Confusion Matrix')
    plt.show()
```

[28]
```python
# Visualize confusion matrix
plot_confusion_matrix(y_test, y_pred_test, label_encoder.classes_)
```





References:

https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

Abimbola, B., de La Cal Marin, E. and Tan, Q. (2024) "Enhancing legal sentiment analysis: A Convolutional Neural Network–long short-term memory document-level model," *Machine learning and knowledge extraction*, 6(2), pp. 877–897. doi: 10.3390/make6020041.

Ahmed, S. F. *et al.* (2023) "Deep learning modelling techniques: current progress, applications, advantages, and challenges," *Artificial intelligence review*, 56(11), pp. 13521–13617. doi: 10.1007/s10462-023-10466-8.

Bharti, S. K. *et al.* (2022) "Text-based emotion recognition using deep learning approach," *Computational intelligence and neuroscience*, 2022, pp. 1–8. doi: 10.1155/2022/2645381.

Bonnet, A. (2023) *Fine-tuning models: Hyperparameter optimization*, *Encord.com*. Encord Blog. Available at: https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/ (Accessed: May 8, 2024).

David, D. (2020) *Hyperparameter optimization techniques to improve your machine learning model's performance*, *freecodecamp.org*. Available at: https://www.freecodecamp.org/news/hyperparameter-optimization-techniques-machine-learning/ (Accessed: May 8, 2024).

Durna, M. B. (2024) *Advanced word embeddings: Word2Vec, GloVe, and FastText*, *Medium*. Available at: https://medium.com/@mervebdurna/advanced-word-embeddings-word2vec-glove-and-fasttext-26e546ffedbd (Accessed: May 8, 2024).

Guo, J. (2022) "Deep learning approach to text analysis for human emotion detection from big data," *Journal of intelligent systems*, 31(1), pp. 113–126. doi: 10.1515/jisys-2022-0001.

Islam, M. S. *et al.* (2024) "Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach," *Artificial intelligence review*, 57(3). doi: 10.1007/s10462-023-10651-9.

Jim, J. R. *et al.* (2024) "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," *Natural Language Processing Journal*, 6(100059), p. 100059. doi: 10.1016/j.nlp.2024.100059.

Kardakis, S. *et al.* (2021) "Examining attention mechanisms in deep learning models for sentiment analysis," *Applied sciences (Basel, Switzerland)*, 11(9), p. 3883. doi: 10.3390/app11093883.

Machová, K. *et al.* (2023) "Detection of emotion by text analysis using machine learning," *Frontiers in psychology*, 14. doi: 10.3389/fpsyg.2023.1190326.

Michael, R. (2020) *Optimizing hyperparameters the right way*, *Towards Data Science*. Available at: https://towardsdatascience.com/optimizing-hyperparameters-the-right-way-3c9cafc279cc (Accessed: May 8, 2024).

Ottoni, A. L. C. *et al.* (2023) “Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets,” *International journal of machine learning and cybernetics*, 14(1), pp. 171–186. doi: 10.1007/s13042-022-01555-1.

Salehin, I. *et al.* (2024) “AutoML: A systematic review on automated machine learning with neural architecture search,” *Journal of Information and Intelligence*, 2(1), pp. 52–81. doi: 10.1016/j.jiixd.2023.10.002.

Saxena, P. (2024) *Hyperparameter tuning with Bayesian Optimization*, *Comet*. Available at: https://www.comet.com/site/blog/hyperparameter-tuning-with-bayesian-optimization (Accessed: May 8, 2024).

Takyar, A. (2023) *Exploring the power of Attention mechanism in deep Learning*, *LeewayHertz - AI Development Company*. Leewayhertz. Available at: https://www.leewayhertz.com/attention-mechanism/ (Accessed: May 8, 2024).

(No date a) *Towardsdatascience.com*. Available at: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa (Accessed: May 8, 2024).

(No date b) *Researchgate.net*. Available at: https://www.researchgate.net/publication/357790518_Deep_learning_approach_to_text_analysis_for_human_emotion_detection_from_big_data (Accessed: May 8, 2024).

(No date c) *Researchgate.net*. Available at: https://www.researchgate.net/publication/362876354_Text-Based_Emotion_Recognition_Using_Deep_Learning_Approach (Accessed: May 8, 2024).

(No date d) *Researchgate.net*. Available at: https://www.researchgate.net/publication/374470053_Study_of_Natural_Language_Processing_for_Sentiment_Analysis (Accessed: May 8, 2024).

(No date e) *Machinelearningmastery.com*. Available at: https://machinelearningmastery.com/scikit-optimize-for-hyperparameter-tuning-in-machine-learning/ (Accessed: May 8, 2024).