Title: Emotion Intensity Analysis in Text: A Comprehensive Approach

Abstract: This technical paper outlines a robust method for identifying the intensity of emotions in text using natural language processing (NLP) techniques. We explore various models and approaches to capture the nuanced expressions of emotions in textual data. Our methodology encompasses pre-processing steps, feature extraction, and the implementation of machine learning models.

1. Introduction: Understanding the intensity of emotions in text is crucial for applications such as sentiment analysis, customer feedback analysis, and mental health monitoring. In this paper, we present an in-depth analysis of different models and techniques for emotion intensity identification.

2. Models Directory: The models directory contains the following subdirectories:

- preprocessing: Code for cleaning and preprocessing text data.
- feature_extraction: Implementation of methods to extract features from text.
- models: Machine learning models for emotion intensity analysis.
- evaluation: Code for evaluating the performance of the models.

3. Requirements:

- Python 3.x
- Pandas
- NumPy
- NLTK
- Scikit-learn
- TensorFlow

4. Model Directory Structure:

models/ |-- preprocessing/ | |-- clean_text.py | |-- preprocess_data.py |-- feature_extraction/ | |-- tfidf_extraction.py | |-- word_embedding_extraction.py |-- models/ | |-- emotion_intensity_model.py |-- evaluation/ | |-- evaluate_model.py |-- README.md

5. Instructions:

Statistical Model:

Run feature_extraction.py to extract statistical features.

Execute statistical_model.py to train the statistical model.

Use evaluate_statistical_model.py for evaluating the statistical model.

Deep Learning Model:

Execute preprocessing.py for text preprocessing.

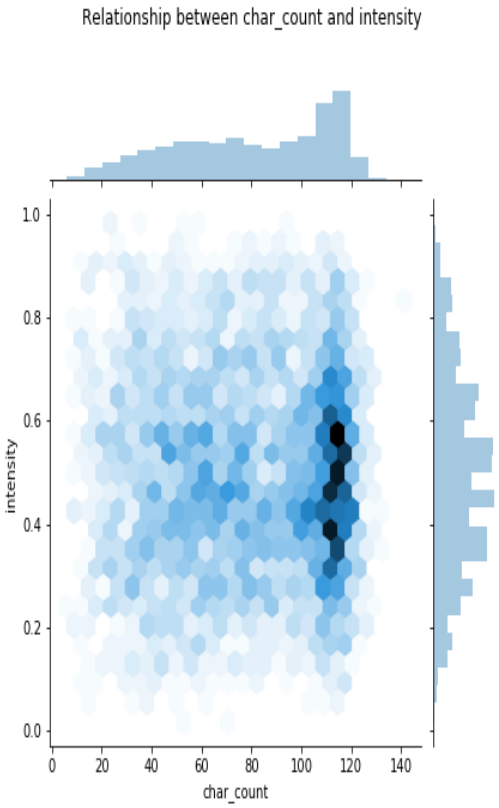
Run deep_learning_model.py to train the deep learning model.

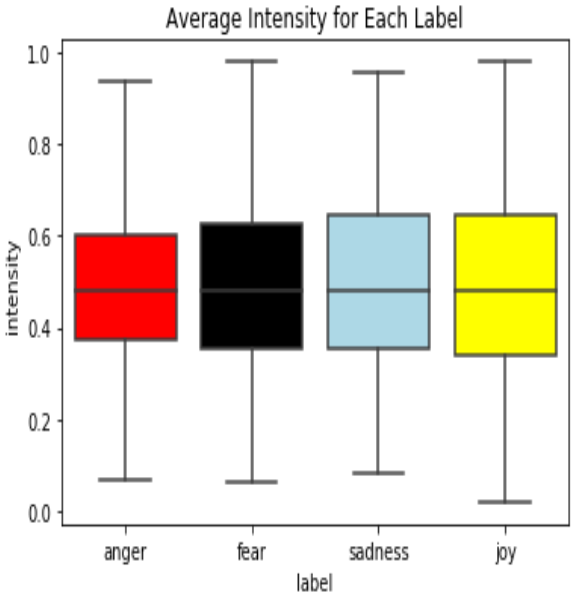
Use evaluate_deep_learning_model.py for evaluating the deep learning model.

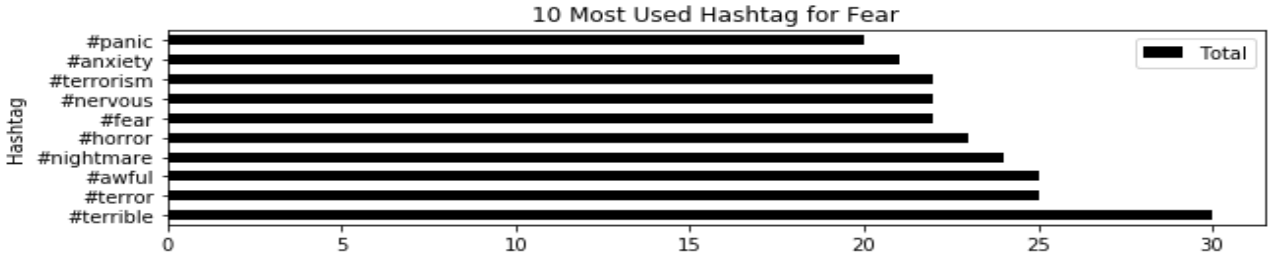
6. Report: The report provides a detailed account of the preprocessing steps, feature extraction methods, and the architecture of the emotion intensity model. It includes a comparative analysis of different approaches and their effectiveness in capturing emotion intensity.

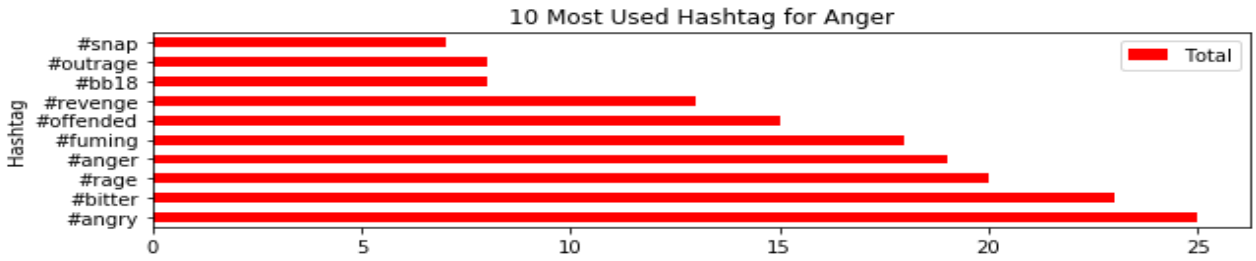
7. Conclusion: Our approach combines state-of-the-art NLP techniques with machine learning models to accurately identify the intensity of emotions in text. The provided code and report offer a comprehensive guide for implementing and understanding the proposed methodology.

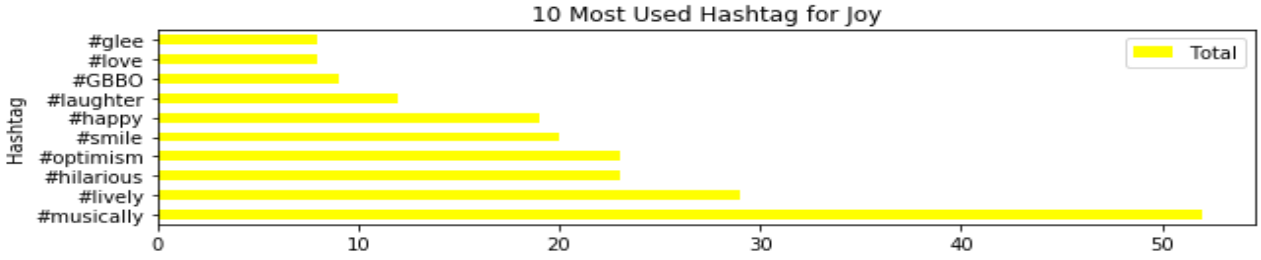
8.Graphs:

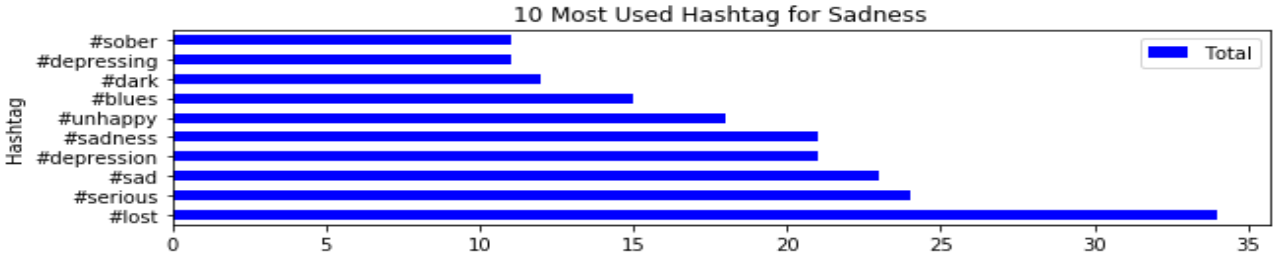


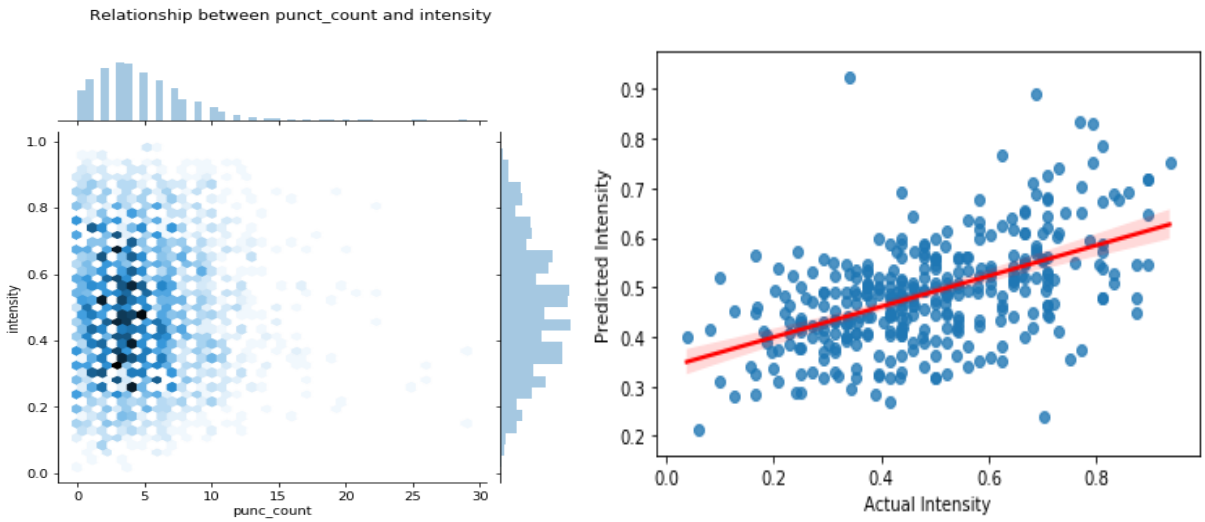


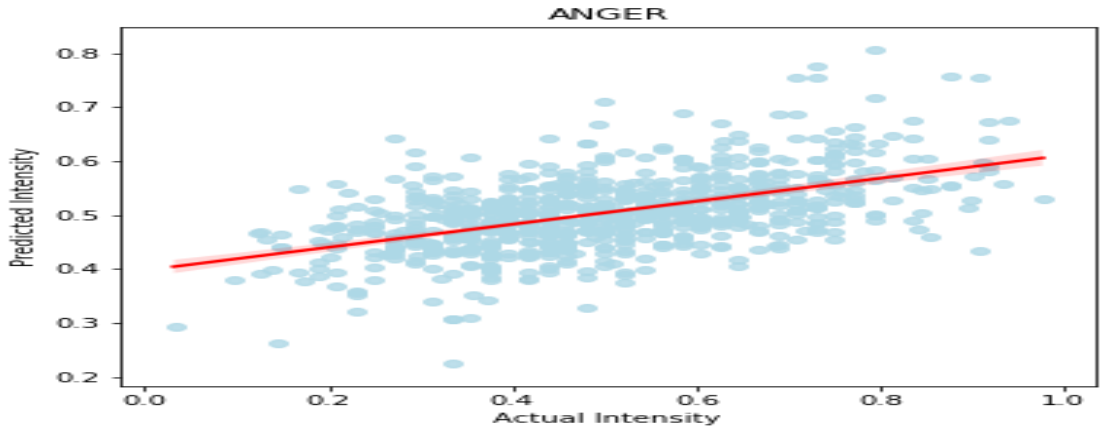

10 Most Used Hashtag for Fear
Hashtag
#panic
#anxiety
#terrorism
#nervous
#fear
#horror
#nightmare
#awful
#terror
#terrible
Total
0
5
10
15
20
25
30

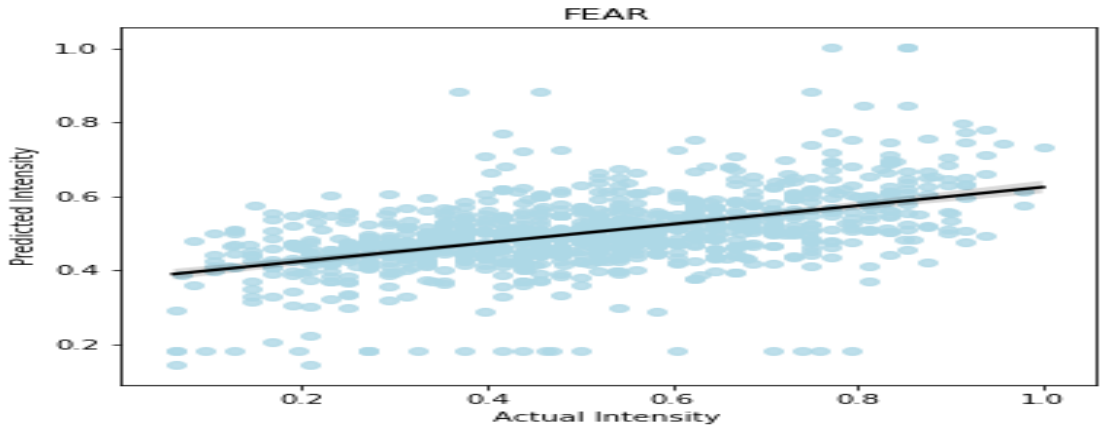

10 Most Used Hashtag for Anger
Hashtag
#snap
#outrage
#bb18
#revenge
#offended
#fuming
#anger
#rage
#bitter
#angry
Total
0
5
10
15
20
25

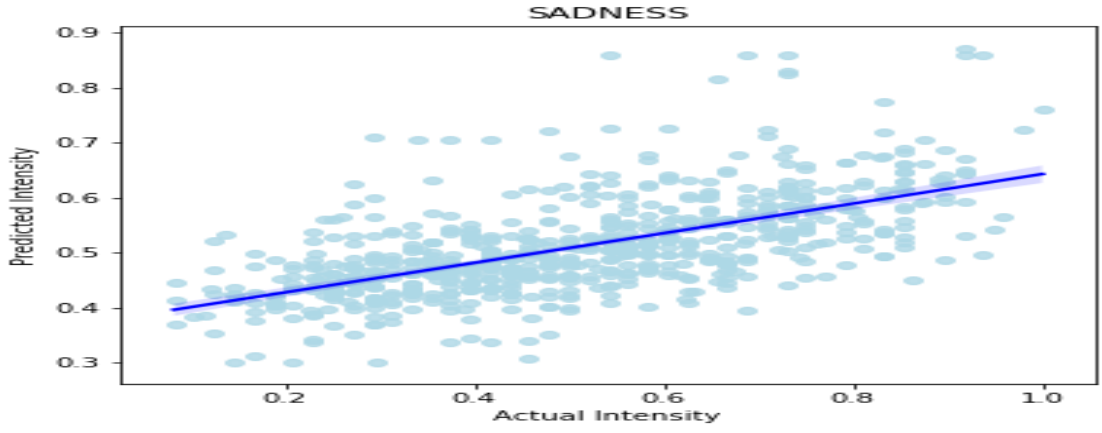

10 Most Used Hashtag for Joy
Hashtag
#glee
#love
#GBBO
#laughter
#happy
#smile
#optimism
#hilarious
#lively
#musically
Total
0
10
20
30
40
50

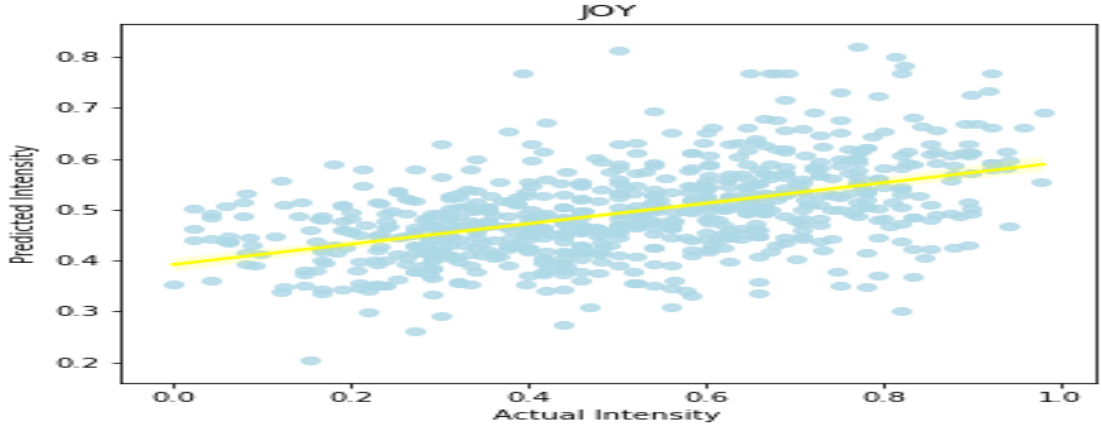

10 Most Used Hashtag for Sadness
Hashtag
#sober
#depressing
#dark
#blues
#unhappy
#sadness
#depression
#sad
#serious
#lost
Total
0
5
10
15
20
25
30
35


Relationship between punct_count and intensity
intensity
punc_count
Predicted Intensity
Actual Intensity

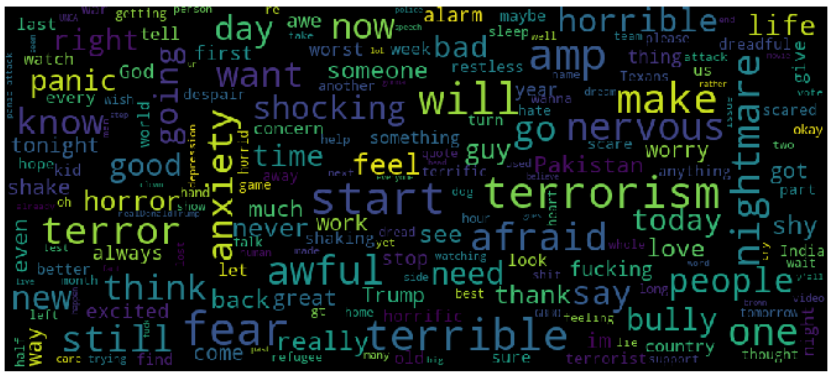

ANGER
Predicted Intensity
Actual Intensity
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8


FEAR
Predicted Intensity
Actual Intensity
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8
1.0


SADNESS
Predicted Intensity
Actual Intensity
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.7
0.8
0.9


JOY
Predicted Intensity
Actual Intensity
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8

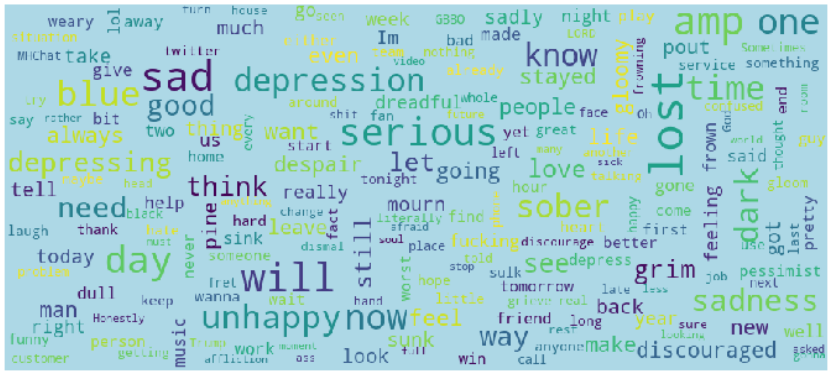
Fear



Anger

Joy

Sadness



Analysis:

From the results of the Exploratory Data Analysis, I see that there is no significant difference between the average intensity for the four emotions. It also appears that there is no significant pattern that indicates the relationship between intensity and the number of characters, words, and punctuation. Therefore, I don't use the char_count, word_count, and punct_count features to train the model.

The algorithms that I try to train the regression model are linear regression, ridge regression, bayesian, KNN regression, SVR, and decision tree regressors. I also tried using vectorization with Bag of Words and tf idf techniques. I use training data to train models, and data development to test models and see which models produce the best results. It turned out that the SVR model with the tf idf vectorization produced the lowest error, so I chose to make the final model.

Each emotion has its own model. From the plot and assessment metrics for each model, it can be seen that the four models produce better results than if the intensity was chosen randomly. This can be seen in the regression line, where on average if the real intensity increases, the intensity of the predictions also increases. It can also be seen that the anger model produces the lowest errors, while the joy model produces the highest errors.

The Pearson Correlation results also show a correlation of more than 0, which means there is a linear positive relationship between predicted and actual intensity. One thing that is interesting is that the Pearson Correlation on average shows lower results when gold scores are only taken between 0.5-1. This shows that the model works worse when the actual intensities of the predicted tweet is more than average.

But of course the prediction results are still far from perfect. It can be seen that the gradient of the four regression lines of the model has a gradient that is too low, where tweets with very low intensity are usually predicted to have higher intensities, and tweets with very high intensity are usually predicted to have lower intensities. It can also be seen on the plot that there are still many points that are far from the original intensity, such as in the fear model, where some tweets with high intensities are predicted to have low intensities. The Pearson Correlation of the joy model is also quite low when the gold score is between 0.5-1 (0.298). This proves that the Joy model cannot accurately predict tweets with high intensity.

There are several ways that I think can improve the results of the model. The first way is to correct the words in the tweets that are spelled incorrectly, so that the vectorization process will produce results that are more in line with reality. The second way is to increase the number of train tweets, because surely there are still many words contained in the test data that are not in the data train. The third way is to make models with different algorithms for each emotion, because maybe an algorithm works well on one emotion and less well on another emotion.