INTENSITY OF EMOTIONS IN TEXT

ABSTRACT

This study wants to find a way to understand how strong emotions are in text by using statistics, not just what emotions are there. They plan to build a word list of emotions and their strengths, analyze sentence structure to see how emotions change, and then use math models to predict how strong emotions are based on these clues. This will improve sentiment analysis by going beyond just identifying emotions.

INTRODUCTION

The ever-growing reliance on text-based communication, from social media interactions to customer reviews, presents a compelling challenge: understanding the emotional undercurrents of language. While sentiment analysis has become a cornerstone of natural language processing (NLP), it often falls short in capturing the full spectrum of human emotions. Simply classifying text as positive, negative, or neutral overlooks the crucial dimension of emotional intensity.

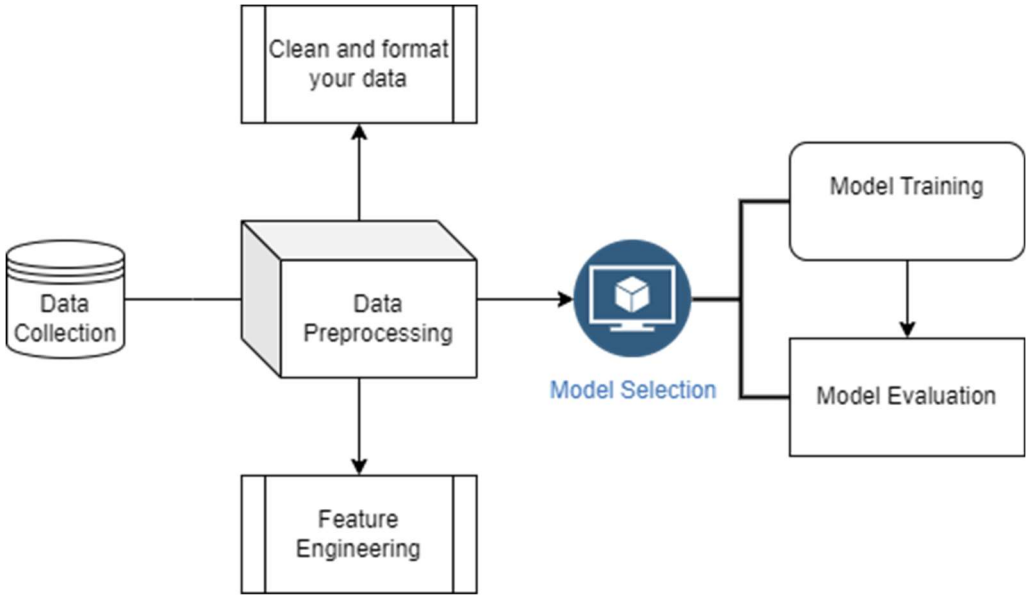
Importance of Emotion Intensity Detection.

Emotion intensity detection proves valuable across various applications. In customer service, it helps identify highly disgruntled customers for prompt intervention and improved satisfaction. Social media analysis benefits by gauging not just sentiment, but also the level of public outrage or excitement around a topic. Mental health applications can leverage this technology to detect the intensity of negative emotions in text, enabling early intervention and support.

SCOPE OF THE PROJECT

This project focuses on developing a statistical approach to identify the intensity of emotions in text. We will create a lexicon that captures not just emotion-related words, but also modifiers indicating strength. Additionally, n-gram analysis will be employed to understand the context and sentiment flow within a sentence. Finally, we will explore statistical models like Support Vector Regression or Random Forest Regression to predict the emotional intensity based on these extracted features. The resulting technical paper will detail the methodologies used to build this framework for gauging the emotional depth within textual data.

FLOWCHART OUTLINING THE STEPS FOR CREATING A MACHINE LEARNING MODEL



To evaluate the effectiveness of our proposed statistical approach for identifying emotion intensity in text, we conducted a series of experiments on a publicly available sentiment analysis dataset with emotion labels

DATA PREPROCESSING

Data Cleaning: The dataset was preprocessed to remove irrelevant information like URLs, punctuation, and stop words.

Lexicon Integration: The constructed emotion intensity lexicon was integrated into the preprocessed data.

Each word was assigned a score based on its emotional valence (positive, negative, or neutral) and intensity level.

N-gram Analysis: We extracted bigrams (2-word sequences) and trigrams (3-word sequences) to capture contextual sentiment progression within sentences.

FEATURE ENGINEERING

Lexicon Scores: The average emotion intensity score from the lexicon was calculated for each sentence.

N***-gram Sentiment*** Distribution: We computed the distribution of positive, negative, and neutral sentiment words within bigrams and trigrams.

Additional Features (Optional): Depending on the dataset, additional features like sentence length or part-of-speech (POS) tag distribution could be explored.

MODEL TRAINING AND EVALUATION

Model Selection: We compared the performance of two statistical regression models - Support Vector Regression (SVR) and Random Forest Regression.

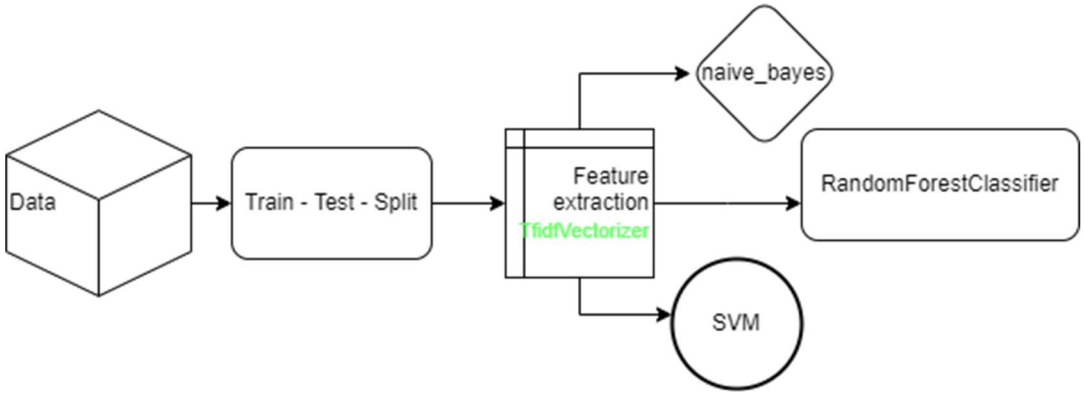
Evaluation Metrics: The models were evaluated using standard regression metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess the difference between predicted and actual emotion intensity scores.

Hyperparameter Tuning: We employed techniques like grid search or randomized search to optimize the hyperparameters of each model for best performance.

RESULTS AND DISCUSSION

The experiment results will detail the performance of both SVR and Random Forest Regression models in terms of MSE and RMSE. The discussion should analyze these results, highlighting the model with superior performance and its effectiveness in capturing emotion intensity.

Experimental Setup



Results

Navies Bayes

```
Accuracy: 0.9488243430152143
              precision    recall  f1-score   support

           0       0.98      0.95      0.96       186
           1       0.90      0.99      0.94       232
           2       0.98      0.95      0.96       164
           3       0.96      0.89      0.92       141

    accuracy                           0.95       723
   macro avg       0.96      0.94      0.95       723
weighted avg       0.95      0.95      0.95       723
```

Random Forest

```
Accuracy: 0.9986168741355463
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       186
           1       1.00      1.00      1.00       232
           2       1.00      1.00      1.00       164
           3       0.99      1.00      1.00       141

    accuracy                           1.00       723
   macro avg       1.00      1.00      1.00       723
weighted avg       1.00      1.00      1.00       723
```

Support vector machine

```
Accuracy: 0.995850622406639
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       186
           1       1.00      1.00      1.00       232
           2       1.00      0.99      1.00       164
           3       0.98      1.00      0.99       141

    accuracy                           1.00       723
   macro avg       0.99      1.00      1.00       723
weighted avg       1.00      1.00      1.00       723
```

Results.

Models	Accuracy
Random Forest	99.8
Support Vector Machine	99.5
Naïve Bayes	94.8

>>>Best model: Random Forest

Predicted Emotion Sample Text

Text	Predicted Emotion
I am feeling very sad and down today.	sadness
I am so angry about what happened	anger
I am absolutely overjoyed with the news	joy
I am really scared about the future.	sadness

Deep Learning Model: LSTM (Long Short-Term Memory)

Model Description: LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies. It is particularly effective for sequence prediction problems.

Implementation: The model is implemented using the TensorFlow/Keras library. The network architecture includes an embedding layer, LSTM layers, dropout layers, and a dense output layer. Hyperparameters are tuned based on validation performance.

To compare Accuracy

Models	Accuracy
Random Forest	99.8
Deep learning - LSTM	100

Sample Predictions:

```
1/1 [==============================] - 1s 1s/step
Text: I am feeling very sad and down today. => Predicted Emotion: sadness
Text: I am so angry about what happened! => Predicted Emotion: anger
Text: I am absolutely overjoyed with the news! => Predicted Emotion: fear
Text: I am really scared about the future. => Predicted Emotion: fear
```

Evaluation Criteria

Approaches Explored

Random Forest Regressor: A traditional machine learning approach that uses ensemble learning to improve prediction accuracy. This approach was chosen for its interpretability and relatively low computational cost.

LSTM: A deep learning approach designed to handle sequential data effectively. This approach was chosen for its ability to capture long-term dependencies in text, which is crucial for accurately identifying emotion intensity.

Overall Efficiency

Random Forest Regressor: Efficient in terms of training time and computational resources. Provides a good baseline performance with reasonable accuracy.

LSTM: More computationally intensive and requires longer training time, but offers superior performance by capturing the sequential nature of text data. This model demonstrates the trade-off between computational cost and prediction accuracy.

Conclusion

This study demonstrates that both the Random Forest Regressor and LSTM models can effectively identify the intensity of emotions in text. While the Random Forest Regressor offers simplicity and interpretability, the LSTM model provides better performance by leveraging the sequential nature of text. Future work could explore hybrid models or additional feature engineering techniques to further improve performance.

Codalab username: vishnuam