Emotion Classification And Intensity Prediction

Codlab Username: KOTTURI KOUSHIL

Email: koushilkotturi2209@gmail.com

Abstract:

The goal of this project is to analyze sentiment and intensity from textual data, with a particular emphasis on tweets. The objective is to create a system that can automatically identify the different emotions that can be expressed in a tweet (such as anger , fear, joy or sadness) and forecast the degree of intensity that each emotion . The project investigates two methods:

1.Deep Learning Model:

This method extracts features from tweet text sequences by using a recurrent neural network of the Long Short-Term Memory (LSTM) type. After that, the model expands to forecast two outputs: A softmax activation layer is used for the emotion category in multi-class classification (one-hot encoded labels). Intensity score (if continuous) with regression using a linear activation layer.

2.Machine Learning Models:

This approach employs separate machine learning models for emotion classification and intensity prediction:

Emotion Classification: A model like Support Vector Machine (SVM) or Random Forest is trained to classify emotions based on features extracted from the tweets using TF-IDF (Term Frequency-Inverse Document Frequency).

Intensity Prediction: A model like Linear Regression is trained to predict the intensity score based on the same TF-IDF features.

1. Data Aquisition

The data for this project was collected from a provided website . This website provides a dataset containing tweets, their corresponding emotions, and a continuous intensity variable.

1.1 Understanding The Data

General Overview: I looked at the first few rows of the Data Frame to get a sense of the column names, data types, and sample values.

Detailed Information: I obtained a detailed summary of the Data Frame, including the index type, number of columns, data types of each column, the number of non-null values in each column, and the memory usage.

Missing Data Analysis: I identified the columns with missing values and quantified their extent. This helped me understand which parts of the data might require cleaning or imputation techniques.

Duplicate Row Check: I checked for duplicate rows in the Data Frame, ensuring the data wasn't skewed by identical entries.

1.2 Performing EDA

To gain deeper insights into the dataset's characteristics and distribution, I employed the pandas profiling library for Exploratory Data Analysis (EDA). Pandas profiling provides a comprehensive report summarizing various aspects of the data.

2. Text Preprocessing

I performed text preprocessing on the tweet data to prepare it for model training. This is a crucial step in Natural Language Processing (NLP) tasks, as raw text data often contains inconsistencies, noise,

and irrelevant information that can hinder model performance. Text preprocessing aims to clean, structure, and transform the text data into a format suitable for machine learning models.

2.1 Lower Casing

Converting all text to lowercase eliminates the distinction between uppercase and lowercase letters. This reduces the number of features the model needs to learn and improves consistency.

2.2 Punctuation Removal

Removing punctuation marks like commas, periods, question marks, etc., can be beneficial depending for our task. Punctuation might not contribute to sentiment analysis or intensity prediction, and removing them simplifies the data.

2.3 Chat Word Elimination

Chat abbreviations can be ambiguous for NLP models. Replacing them with full words provides a clearer and more consistent vocabulary for the model to learn from. By using full words, the model can focus on the actual meaning conveyed in the text, potentially leading to improved performance in tasks like sentiment analysis or intensity prediction.

2.4 Stop Word Removal

Stop words are common words like "the," "a," "an," "is," etc., which carry little meaning in sentiment analysis or intensity prediction. Removing stop words reduces the number of features and helps the model focus on more meaningful content.

2.5 Handling Emojis

Emojis can be ambiguous for machine learning models. Replacing them with clear textual meanings provides a consistent and interpretable representation for the model to learn from. The model can focus on the actual sentiment conveyed by the emoji (e.g., "sadness" instead of a sad face emoji) rather than the visual symbol itself, potentially leading to better predictions.

2.6 Stemming or Lemmatization

These techniques reduce words to their root forms (e.g., "running" to "run", "better" to "good"). This helps capture the meaning without considering variations in verb tenses or plurals. However, it might sometimes lead to loss of information, so it's important to choose the appropriate technique based on your data and task.

3. Visualization

In a word cloud, the size of a word reflects its frequency of appearance. This allows you to quickly identify the most frequently used words in the tweets after processing . These words are likely to be the most significant in conveying sentiment or intensity.

4. Feature Engineering

Feature engineering is a critical process that turns text data (tweets) into numerical values that your machine learning or deep learning model can use to interpret and assess sentiment or intensity. Numbers generated by text-to-number conversion methods such as TF-IDF (Term Frequency-Inverse Document Frequency) are not generated at random. Words are given weights according to their frequency throughout the whole dataset and their importance within a document (tweet). These weights turn into features in and of themselves, emphasizing the significance of particular words in sentiment analysis.
As a result, the model can concentrate on the words that provide the most information, possibly improving its ability to predict sentiment or intensity.

5.Model Building

Machine Learning Task :

An SVM (Support Vector Machine) model is chosen for emotion classification. SVMs are well-suited for classification tasks with potentially complex decision boundaries. Random Forest Regressor model is selected for intensity prediction. Random Forest Regressor is appropriate for predicting continuous numerical values (like intensity scores) based on the relationship with features.

Deep Learning Task:

Words are transformed into numerical vectors, or embeddings, in the first layer to represent semantic relationships.
Two outputs from the model are:
Emotion Classification: Employs "softmax" activation to predict the likelihood of each emotion category (one-hot encoded).
Intensity Prediction: Makes use of a basic linear layer to predict a continuous intensity score.
Paddled tweet sequences with corresponding encoded emotions and intensities are used to train the model. Based on the type of intensity data (numerical or categorical), loss functions are selected.