Predicting Emotion Intensity in Tweets

Objectives

1.Purely statistical model

Data Loading

Data was loaded from tab-separated files into Pandas Data Frames. Each dataset contained four columns: id, tweet, emotion, and degree. The degree column represents the intensity of the emotion in the tweet.

Model Training

A Ridge Regression model was chosen for its effectiveness in handling multicollinearity in high-dimensional data like TF-IDF vectors.

TF-IDF Vectorization and Ridge Regression Model

The model combines TF-IDF vectorization and Ridge Regression to predict emotion intensity in tweets. TF-IDF transforms text data into numerical features by measuring the importance of words based on their frequency and uniqueness across documents. This high-dimensional feature space is then used by Ridge Regression, a linear model that mitigates multicollinearity by introducing a regularization term. This pipeline effectively captures the nuanced expression of emotions in text and makes accurate predictions about their intensity. The model was trained and validated on datasets for anger, joy, sadness, and fear, achieving competitive Mean Squared Error (MSE) scores.

Term Frequency (TF): Measures how frequently a term occurs in a document. It is the ratio of the number of times a word appears in a document to the total number of words in the document.

- $\mathrm{TF}(t)=$ Number of times term t appears in a document/ Total number of terms in the document

Inverse Document Frequency (IDF): Measures how important a term is. While computing TF, all terms are considered equally important. However, certain terms like "is", "of", and "that" may appear frequently but have little significance. Therefore, we need to weigh down the frequent terms while scaling up the rare ones.

- IDF(t) = log(Total number of documents/Number of documents with term t in it)

TF-IDF: The TF-IDF score is the product of the TF and IDF scores for a term.

- $TF\text{-}IDF(t) = TF(t) \times IDF(t)$

The data was split into training (80%) and validation (20%) sets. Mean Squared Error (MSE) was used to evaluate model performance on the validation set.

Predictions and Saving Results

Predictions were made on test datasets, and the results were saved to new files. This process involved loading the test data, making predictions, and saving the predicted degrees.

Results

The Mean Squared Error (MSE) for each emotion's validation set is as follows:

Anger Model: MSE = 0.021378882821692174

Joy Model: MSE = 0.030532023503292736

Sadness Model: MSE = 0.02552532835528938

Fear Model: MSE = 0.024241308988040174

The models were then used to predict the emotion intensities in the respective test datasets, and the results were saved to files:

predicted_anger.txt

predicted_joy.txt

predicted_sadness.txt

predicted_fear.txt

Conclusion

This project successfully implemented a pipeline to predict the intensity of various emotions in tweets. The combination of TF-IDF vectorization and Ridge Regression proved effective for this task. Future improvements could include experimenting with different regression models, tuning hyperparameters, and incorporating more sophisticated text preprocessing techniques.

2. Deep learning model

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking model introduced by Google in 2018 for natural language processing (NLP) tasks. Unlike traditional models, BERT processes text bidirectionally, meaning it considers the context from both the left and right of each word simultaneously, allowing it to understand the nuanced meaning of words in context. BERT is pre-trained on a massive corpus of text, including the entire Wikipedia and

the Book Corpus dataset, using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, random words in a sentence are masked and BERT learns to predict them. In NSP, BERT learns to predict if a sentence logically follows another. After pre-training, BERT can be fine-tuned on specific tasks with relatively small datasets, making it extremely versatile and powerful for tasks such as text classification, sentiment analysis, and question answering. Its architecture is based on the Transformer model, which relies on self-attention mechanisms to process input sequences efficiently.

In this model, BERT was used to predict the intensity of emotions—anger, joy, sadness, and fear—in tweets. The process included several key steps:

Data Loading and Preprocessing:

- The code defines paths to datasets for different emotions (joy, sadness, fear, anger) split into training and testing sets. Each dataset is loaded using pd.read_csv() with specific column names (id, tweet, emotion, score).
- The 'NONE' values in the test dataset scores are replaced with NaN values (pd.NA).

Tokenization with BERT:

- BERT tokenizer (BertTokenizer) from the Hugging Face Transformers library is used to tokenize tweets.
- Tweets are tokenized to a maximum length of 128 tokens (MAX_LENGTH), ensuring uniformity with padding and truncation.

BERT Model Integration:

- The pre-trained BERT model (TFBertModel from 'bert-base-uncased') is employed as the base model. The model is set up to receive input ids and attention masks as inputs.
- The model architecture includes a regression head (Dense layer with linear activation) on top of BERT's pooled output.

Training and Prediction:

- For each emotion category, a separate model instance is created and trained. Training inputs (input_ids and attention_mask) are prepared using the tokenized training data.
- The model is compiled with an Adam optimizer and mean squared error loss ('mean_squared_error').
- Training is performed for 3 epochs with a batch size of 16. A validation split of 10% is used for monitoring model performance during training.

Prediction and Results:

- Test inputs are prepared similarly to training inputs using tokenized test data.
- The trained model predicts scores for the test set. Predicted scores are then integrated back into the test data for each emotion category.

Results are saved into separate files (predicted_{emotion}_emotion.txt) using to_csv().This method leverages BERT's advanced language understanding capabilities, making it highly effective for predicting emotion intensity in tweets. The project showcases BERT's versatility in various NLP tasks, particularly in sentiment analysis and emotion detection. This approach provides accurate predictions and highlights the potential of transformer-based models in understanding and analyzing human emotions in text.

conclusion

the combination of TF-IDF Vectorization and Ridge Regression models presents a pragmatic approach for predicting emotion intensity in tweet data. TF-IDF effectively transforms textual content into numerical features by weighing the importance of words based on their frequency and rarity across documents. This method captures essential textual nuances and is particularly adept at handling sparse and high-dimensional data typical of natural language processing tasks.

Ridge Regression, integrated with TF-IDF vectors, provides a robust framework for modeling emotion intensity. By introducing regularization, Ridge Regression mitigates multicollinearity and overfitting, ensuring a stable and interpretable model. It strikes a balance between model complexity and performance, making it suitable for applications where linear relationships between features and targets are assumed.

Compared to more complex models like BERT, TF-IDF and Ridge Regression are computationally efficient and easier to interpret, making them accessible for smaller datasets or scenarios where resource constraints are a consideration. However, they may not capture intricate semantic relationships and contextual nuances as comprehensively as deep learning models.

Overall, TF-IDF and Ridge Regression offer a reliable foundation for sentiment analysis tasks, providing competitive accuracy and interpretability. Future advancements may explore hybrid approaches that combine the strengths of traditional methods with deep learning techniques to enhance performance in capturing nuanced emotional expressions in text data.