Emotion Detection from Plain Text

Ceci Chen, Hongxin Song, Lia Wang, Maggie Xu
DS-GA 1003 Machine Learning

NYU Center for Data Science

Overview

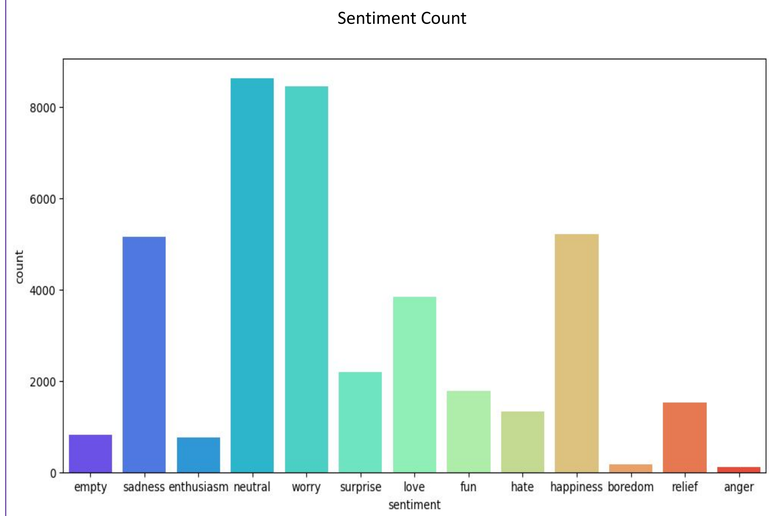
The detection of emotion from textual data has addressed a variety of applications that span mental health monitoring, digital content moderation, and consumer behavior analysis. It not only enriches and advances the interaction between humans and machines. Historically, models such as **Support Vector Machines (SVM)**, **Naïve Bayes (NB)**, and **classic machine learning algorithms** have been employed to analyze emotions in text, but these methods often show limitations in fully capturing human emotions. **BERT,** which stands for **Bidirectional Encoder Representations from Transformers**, is a new state-of-the-art model designed to pre-train on a vast corpus of text using both masked language modeling and next sentence prediction, proven to achieve significant improvements in accuracy. Our research aims to further this field by **integrating BERT's advanced capabilities with emotion detection from text**.

Data Collection	EDA & Tokenization	Modeling	Evaluation	Results
• Clean raw data and unify data format	• Analyze data distributions • Visualize sentiment categories • Split train-test dataset and convert into TF-IDF features	• Use Decision Tree, Random Forest, and Logistic Regression as Baseline Models • BERT	• Fine-tuning the model • Evaluate model performance with accuracy & confusion matrix	• Use BERT model improved accuracy rate from 27.3% given by baseline model (Decision Tree) to 42.22%

Data

The datasets we will use is text dataset extracted from Twitter posts from Kaggle, which is a collection of tweets annotated with the emotions behind the content.

- **Dataset Size**: 40,000 rows, 3 columns
- "**tweet_id**": Twitter user ID in integer type
- "**sentiment**": 13 pre-defined emotion categories assign to content in string type
- "**content**": raw tweets content in strings



Methodology

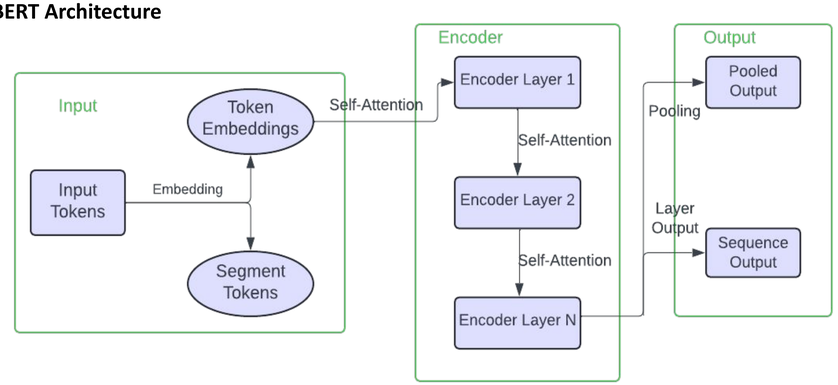
Data Preprocessing

- We first drop the unnecessary 'tweet_id' column and then clean the tweet content by removing tweet handles, html characters and non-letter characters beside space.
- To **balance** the data for further model training, we decided to drop rows with sentiments of **"empty", "enthusiasm", "boredom", and "anger"**, which are the sentiments with relatively smaller number of data points. We use the remaining 9 sentiments for further training.

Model Training

We initiate our exploration by setting a baseline using traditional **decision tree, random forest**, and **logistic regression**, which have shown **limitations** in handling the complexity of NLP tasks. We then integrate **BERT** and conduct **fine-tuning** to find the best setup for the training. We conducted hyperparameter tuning focusing on optimizing the **learning rate,** integrating **dropout layers** to prevent overfitting, adjusting **batch size** for efficient learning, and determining the optimal number of **epochs** to ensure sufficient model training without excessive overfitting.

BERT Architecture

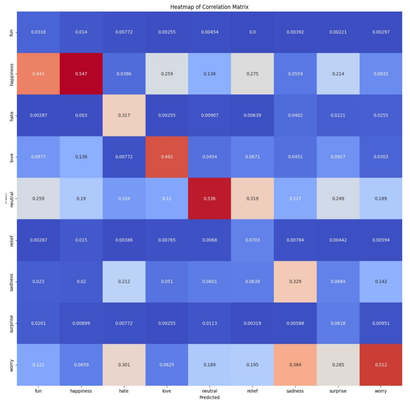


BERT's pre-training strategy is where it differs significantly from other simpler models. BERT is trained using **masked language modeling (MLM)**, as opposed to earlier models which were trained in a generative manner. The input sequence is randomly masked for a portion of the tokens during pre-training, and the model learns to predict those masked tokens based on the context that is left. This pre-training is followed by fine-tuning on particular downstream tasks, such as sentiment analysis or named entity recognition.

Results

Model	Test Accuracy
Baseline(Decision Tree)	27.30%
Logistic Regression	**35.80%**
Random Forest	35.24%

Model	Test Accuracy
Baseline	27.30%
BERT w/ batch size = 16	33.30%
BERT w/ batch size = 64	36.10%
BERT w/ batch size = 128	**37.60%**
BERT w/ drop-out = 0	37.27%
BERT w/ drop-out = 0.1	**42.22%**
BERT w/ drop-out = 0.2	41.38%
BERT w/ lr = 1e-5	40.64%
BERT w/ lr = 2e-5	40.77%
BERT w/ lr = 5e-5	37.36%



Conclusions

Summary:

- The **optimal tuned BERT model** has **batch size as 16**, **learning rate as 1e-5** and **drop out rate as 0.1**. It achieved the highest test accuracy of **42.22%**, surpassing other configurations and the baseline.
- For large model like BERT, **smaller learning rate** would typically achieve better results. **Adding some drop out** and **tuning batch size** would help improve the model's generalization ability.
- Within the confusion matrix, we can tell that **"happy" and "fun", "sad" and "worry"** are two common *misclassified* groups. **"Love"** is the emotion that is *least likely to be classified as other emotions*.

Limitations

- Despite efforts to **mitigate overfitting** (e.g., early stopping), complex models like BERT are prone to memorizing training data, especially with smaller or less diverse datasets.
- BERT may still **struggle with nuances** such as sarcasm, irony, or subtle expressions of emotions, which can be **misinterpreted** without additional contextual or multimodal data.
- The model trained on **specific types of text** (e.g., tweets) may not perform well on different text types (e.g., formal news articles, medical records) without further adaptation.

Future Work

Future exploration of this open-ended question may involves:

- Exploring other methods for tuning the hyperparameters to further increase the model accuracy.
- Experiment with alternative architectures or newer versions of transformer models that might offer advantages in specific aspects of sentiment analysis.
- To improve model robustness and fairness, expand the dataset to include more diverse sources of text and sentiments.

Reference

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10548207/
https://arxiv.org/abs/1810.04805
https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text/data