Emotion Classification Model Development Report

Chin Hui Chu

December 2024

1 Introduction

In this report, I detail my development of an emotion classification model designed to categorize tweets into eight distinct emotions: anger, anticipation, disgust, fear, sadness, surprise, trust, and joy. I utilized the RoBERTa base architecture and incorporated various preprocessing and feature engineering techniques to improve performance.

2 Data Preprocessing Pipeline

2.1 Text Cleaning and Normalization

In my preprocessing pipeline, I implemented several crucial steps to clean and normalize the text data. My key preprocessing steps include:

- Unicode escape sequence handling to properly decode special characters
- Emoji conversion to text descriptions using the emoji library
- Removal of remaining Unicode artifacts
- Whitespace normalization

2.2 Hashtag Integration

One of my notable feature engineering decisions was the integration of hashtags with the main text. Through this approach, I:

- Preserved hashtags as potential emotional indicators
- Ensured no duplicate hashtags were added
- Maintained the natural flow of the text while incorporating additional context

3 Feature Engineering

3.1 Text Tokenization

In my model, I used RoBERTa's tokenizer with specific configurations. My key features include:

- Maximum sequence length of 128 tokens
- Dynamic padding to handle variable-length inputs
- Truncation of longer sequences to maintain consistent input size

3.2 Data Structure and Organization

In organizing the data, I focused on:

- Mapping emotions to numerical labels using a standardized mapping
- Maintaining tweet IDs for tracking and validation
- Ensuring consistent data format for model training

4 Model Architecture and Training

4.1 Model Configuration

I chose to use RoBERTa-base with the following key architectural decisions:

- RoBERTa-base as the foundation model for its strong performance on text classification tasks
- Batch size of 256 to balance between training speed and memory constraints
- Learning rate of 2e-5 for fine-tuning

4.2 Training Strategy

In my training implementation, I included several optimizations with notable features:

- Differential weight decay for different parameter groups
- AdamW optimizer for better convergence
- Linear learning rate scheduling
- Gradient accumulation for stable training

5 Model Performance and Insights

5.1 Evaluation Metrics

I evaluated my model's performance using macro F1-score. My evaluation showed:

Mean F1 Score: 0.5743

Emotion	F1 Score
Anger	0.4331
Anticipation	0.7240
Disgust	0.5344
Fear	0.6162
Sadness	0.5952
Surprise	0.3903
Trust	0.5535
Joy	0.7475

Table 1: Per-class F1 Scores from my model

5.2 Key Insights

Through my analysis, I identified several key insights:

1. **Training Data Size Impact:** I conducted experiments with different training data sizes to understand their impact on model performance:
   - With 1% of training data: achieved mean F1 score of approximately 0.14
   - With 30% of training data: achieved public leaderboard score of 0.50360
   - With full training dataset: achieved 0.53203 on public leaderboard and 0.51757 on private leaderboard

   These results clearly demonstrated that increasing the training data size significantly improved model performance, though with diminishing returns beyond 30% of the data.

2. **Class Imbalance:** I observed varying F1 scores across emotions suggesting class imbalance issues, with emotions like "joy" and "anticipation" showing better performance than "surprise" and "anger".

3. **Text Length:** From the dataset, I observed that while most tweets fit within the 128 token limit, longer tweets containing multiple sentences or detailed emotional expressions might have their content truncated. However, since I did not explicitly test different token limits, this remains an area for potential future investigation.

4. **Hashtag Importance:** During my data analysis, I observed that many tweets expressed their emotional content primarily or exclusively through hashtags rather than in the main text. For example, a seemingly neutral tweet might be labeled as "joy" due to hashtags like "#happy" or "#blessed". This observation led to my decision to merge hashtags with the main text during preprocessing to ensure these crucial emotional indicators weren't lost.

6 Model Configuration and Hyperparameters

In developing my model, I carefully selected the following hyperparameters and configuration settings:

Parameter	**Value**
Base Model	RoBERTa-base
Maximum Sequence Length	128 tokens
Batch Size	256
Learning Rate	2e-5
Number of Epochs	1
Optimizer	AdamW
Weight Decay	0.01 for non-bias parameters 0.0 for bias and LayerNorm
Learning Rate Schedule	Linear with no warmup
Number of Labels	8 (one per emotion)
Random Seed	42

Table 2: Model Hyperparameters and Configuration

These hyperparameters were chosen based on common practices for fine-tuning transformer models and computational resource constraints. The relatively large batch size of 256 helped in processing the substantial amount of training data more efficiently. Given the large size of the training dataset (over 1.4 million tweets) and the computational time required, I made a practical decision to train the model for a single epoch. While additional epochs might have potentially improved the model's performance, the significant computational time required for each epoch made multiple-epoch training impractical within the project's time constraints.