

Emotion Classification & Text Rewriting Web App

1. Overview

This project builds a full-stack NLP system for English emotion classification and tone-aware text rewriting. It includes a backend API, multiple classification models, a text generation module, a React-based frontend, and is deployed via Google Cloud Platform and Firebase.

GitHub Repo: <https://github.com/yyf-yin/nlp-model>

2. Dataset and Exploratory Data Analysis (EDA)

Initial Attempt: Amazon Electronics Reviews

The first dataset used was the UCSD Amazon electronics reviews, aiming to build a sentiment classifier. However:

- Label imbalance: Most reviews were positive; negative reviews were scarce.
 - Lack of diversity: Many reviews were repetitive and short (e.g., great product), making it hard to detect nuanced emotions.
- => Performance was unsatisfactory.

Data Source Evaluation

- X/Twitter or Reddit API: rich content but costly.
 - GoEmotions Dataset by Google Research: over 58k Reddit comments annotated with 27 emotions + neutral.
- => Selected GoEmotions. Cleaned using data-processing.py to produce goemotions_clean.csv.

3. Methodology

3.1 Baseline Model: TF-IDF + Logistic Regression

- Text: TfidfVectorizer (10k features, unigrams + bigrams)
- Model: OneVsRestClassifier(LogisticRegression)
- MultiLabelBinarizer for labels
- Split: 80/20

Evaluation:

- Accuracy: 0.1425
- Micro F1: 0.2505
- Macro F1: 0.1657
- Weighted F1: 0.2211

Emotion Classification & Text Rewriting Web App

3.2 Improved Model: BERT Embedding + Random Forest

- Text encoding: SentenceTransformer('all-MiniLM-L6-v2')
- Classifier: RandomForestClassifier(n_estimators=200)
- Threshold: 0.3

Evaluation:

- Micro F1: 0.3775
- Macro F1: 0.3164
- Weighted F1: 0.3814

3.3 Text Rewriting: LangChain + Gemini API

- Input: original text, target tone, intended audience
- LangChain prompts Google Gemini 1.5 Flash
- Revises or generates fluent English accordingly.

4. System Architecture

Backend (Flask, app.py):

- /predict: TF-IDF classifier
- /revise: Gemini via LangChain

Frontend (React + Ant Design, App.jsx):

- Real-time prediction on input
- Emotion tags shown instantly
- Input tone + audience => revised text

Deployment:

- Backend on Google Cloud Run
- Frontend on Firebase
- Only TF-IDF model deployed due to model size limits.

5. Results

Below is a summary of model performances on the test set:

Model	Accuracy	Micro F1	Macro F1	Weighted F1
TF-IDF + LR	0.1425	0.2505	0.1657	0.2211
BERT + RF	-	0.3775	0.3164	0.3814

Emotion Classification & Text Rewriting Web App

6. Limitations

- TF-IDF model underperforms; BERT+RF model too large to deploy
- Emotion overlap and imbalance cause misclassification
- Gemini-based rewriting depends on API availability

7. Future Work

- Deploy compact transformer-based models (e.g., DistilBERT + MLP)
- Consider embedding + vector DB to reduce memory cost
- Add emotion visualizations, user feedback, and logging features
- Enable streaming inference for BERT-based models