Emotion Analysis Project – Progress Report

Repository: https://github.com/nannanav/emotion-analysis

Our understanding so far

We want to label short Reddit comments with the feelings they show. Right now we turn each comment into a list of word counts and train a logistic regression model. This simple setup helps us learn what matters before we build heavier models.

Dataset – GoEmotions

GoEmotions is a public Reddit dataset with labels made by human annotators.

- We load 43,410 comments for training, 5,426 for validation, and 5,427 for testing.
- Each comment can have more than one of the 28 emotion labels (joy, admiration, anger, sadness, etc.).
- The text is short, casual, and often messy, which matches real social media language.

Bag-of-Words set-ups we tried (Notebook cell 8)

Every vectorizer keeps tokens that appear in at least 2 comments and less than 80% of comments.

- Unigrams only with caps at 5k, 10k, and the full vocabulary.
- Bigrams only with the same caps (5k, 10k, all).
- Mixed unigram + bigram vocabularies at 5k, 10k, 15k, and full size.
- Unigram + bigram with 5k features plus a custom tokenizer that lemmatizes words.

How the models performed (Notebook cell 11)

We trained a One-vs-Rest logistic regression for each feature matrix and checked validation accuracy, micro-F1, macro-F1, and Hamming loss.

Configuration	Vocab size	Accuracy	F1 (micro)	What we saw
uni+bi_5k	5,000	0.354	0.495	Best balance of accuracy and F1.

uni+bi_5k_lemma	5,000	0.356	0.495	Lemmatizing changed scores by less than 0.001.
unigram_5k	5,000	0.350	0.488	More unigram features did not make things better.
unigram_all	13,077	0.349	0.488	Going above ~5k features did not lift F1.
bigram_5k	5,000	0.163	0.281	Bigram-only misses strong single-word cues.

- Adding more than about 5k features did not help any bag-of-words run. It only added training time.
- Combining unigrams and bigrams worked best when we kept the vocabulary small (5k).
- Bigram-only vocabularies performed poorly.
- Lemmatization gave almost the same numbers as the non-lemmatized version.

Next steps (Just a current thinking of what we want to do next, might change in the future)

- Try small neural networks on top of TF-IDF features to capture simple interactions.
- Fine-tune transformer models such as DistilBERT or BERT to read context-rich emotional cues.
- Adjust decision thresholds or calibrate probabilities so we can assign multiple emotions more accurately.
- Watch for rare emotions and test re-weighting or focal loss when we move to neural models.