Emoji Classification😊

Anisha, Cheech, Sofie

Approach:

We decided to build a classifier for sentiment classification and then map the predicted sentiment to emojis.

Dataset:

We found two promising datasets which mapped text to sentiment:

- Falah/sentiments-dataset-381-classes
- dair-ai/emotion · Datasets at Hugging Face

The first dataset had 381 different classes, which would have mapped to a wide range of emojis. The drawback was that it contained only 1060 rows. The second dataset had only 6 classes (joy, sadness, anger, fear, love, and surprise), but 20000 rows. To optimize model performance, we chose having more data over having more classes. In order to display a wide range of emojis, we assigned 3 emojis to each sentiment. This makes sense given that a single emotion can often be represented by a range of emojis.

Data Preprocessing:

Relabeling from 381 labels down to 6 labels
Removing Stopwords using NLTK and tokenizing
Data balancing by generating more text data using ChatGPT. Joy was prominent with over 7000 data points. Balanced anger, fear, love, and surprise to an average of 6000 data points.

Model:

Started with a logistics regression model for multi-class classification. This is because logistic regression is an easy to implement model that works well with TF-IDF vectorized words. Then, we transitioned to random forest because it performs better for classification tasks. Ultimately, we ended up choosing an RNN to be our final model because it captures word order and context which is important when analyzing sentiment.

The RNN was a Keras Sequential Model. We start with a Text Vectorization Layer to convert text into token sequences. We increased our max_tokens size to be 100,000 to capture a wide range of language features. Tokenized sequences were then passed through an Embedding Layer, followed by a Bidirectional LSTM to capture contextual information from past and future words. Finally, the model has a Dense Layer with ReLU activation for nonlinearity and a Dense output with softmax activation to produce probability distributions for each of the six sentiment classes.

Evaluation:

TF-IDF + Logistic Regression: 93% accuracy
Random Forest: 75% accuracy
RNN: accuracy on test data = 89% accuracy