Sentiment Classification of Tweets about Apple and Google

By Jake Oddi

Overview

Objective: Analyze tweets about Apple and Google products and attempt to classify them based on the sentiment within the tweet.

Goals of Analysis:

- Thoroughly preprocess data and engineer features
- Construct reliable, predictive models for sentiment classification

Data Understanding

- 9093 tweets from 2013
- Sourced via CrowdFlower Open Source Datasets
- Tweets are about Apple/Google or either of their respective products or services
- Sentiment labeled by humans
  - ‘Positive emotion’
  - ‘Negative emotion’
  - ‘No emotion toward brand or product’
  - ‘I can’t tell’
- Column indicating product - unreliable

Data Cleaning

- No null values in either tweets or sentiment column, only product
- Converted all tweets to strings
- Renamed columns
- Tokenize tweets, remove stopwords

Initial EDA - Word Clouds

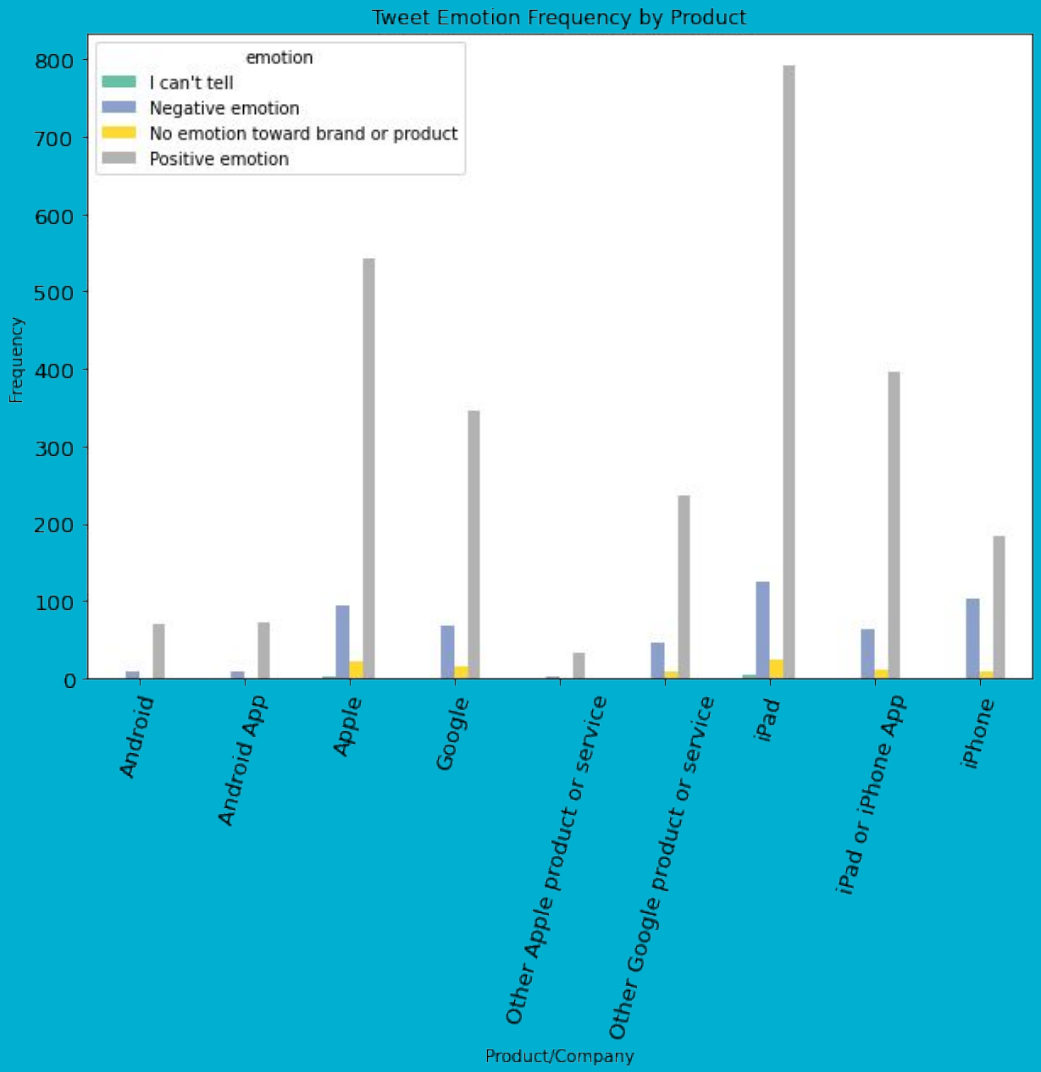
Positive

Negative

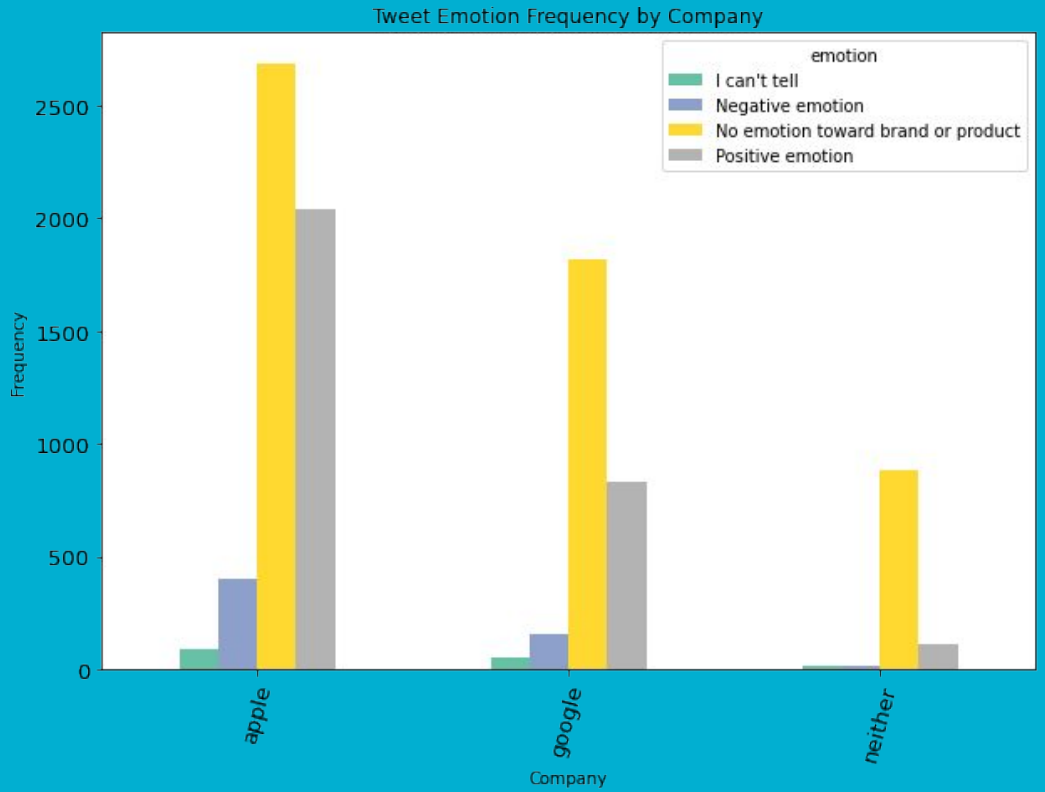
Neutral

Product vs Sentiment

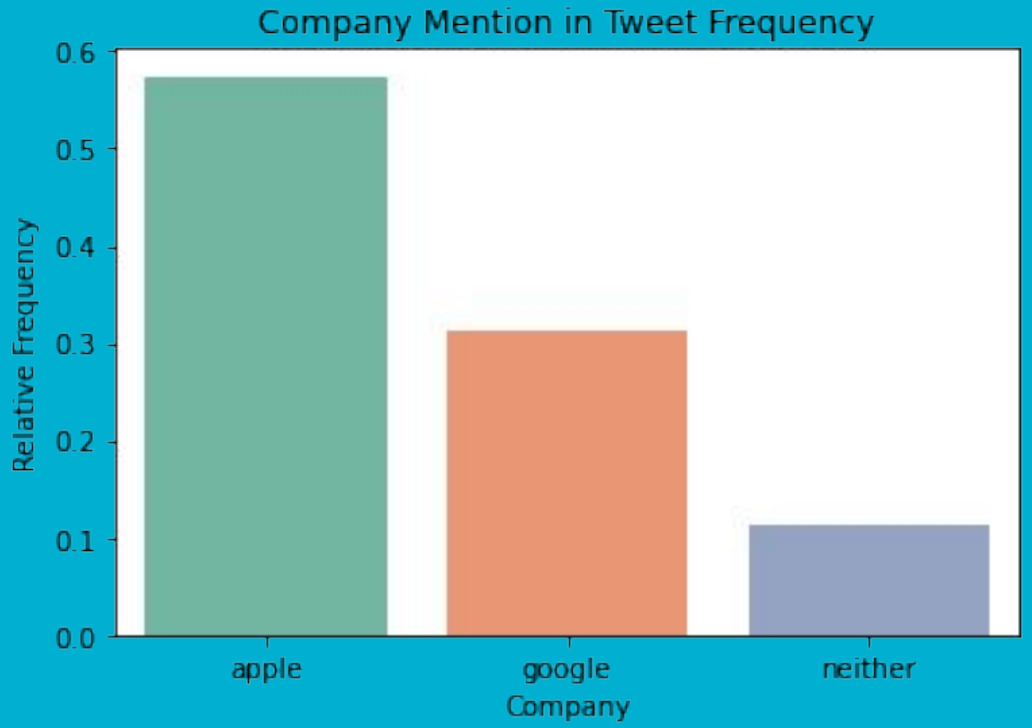
Frequency of different sentiments based on product/company



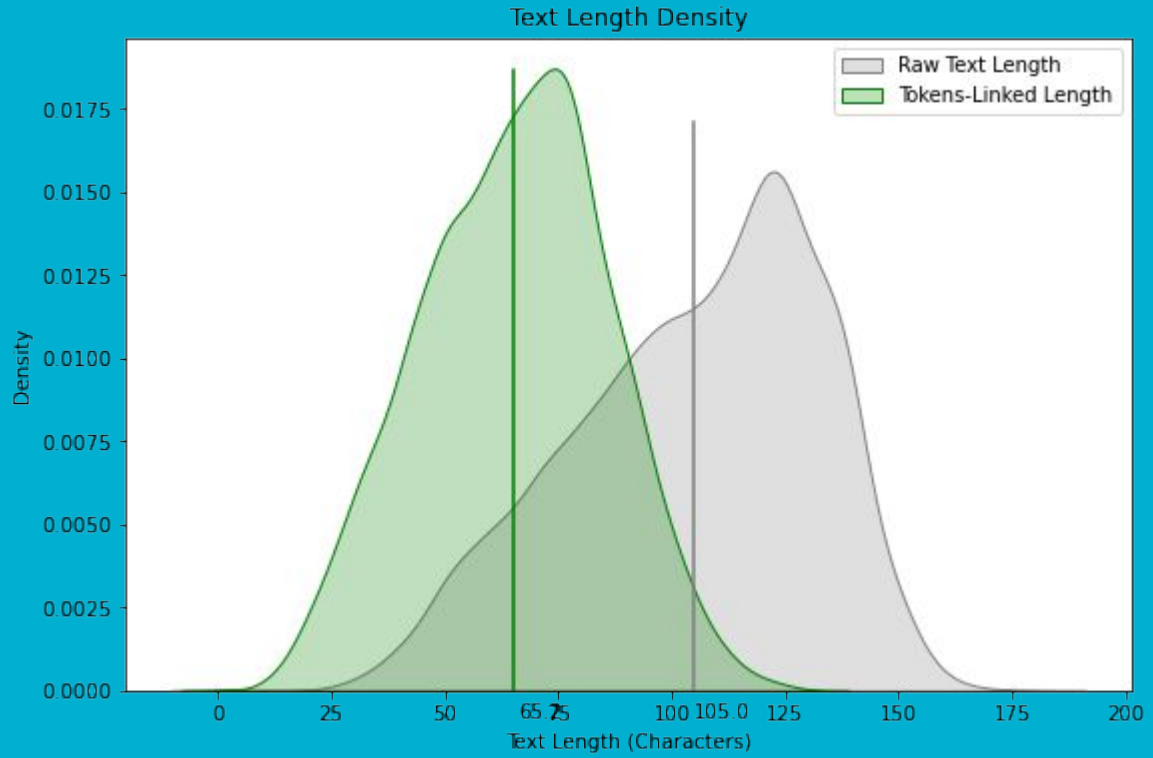
Company vs Sentiment



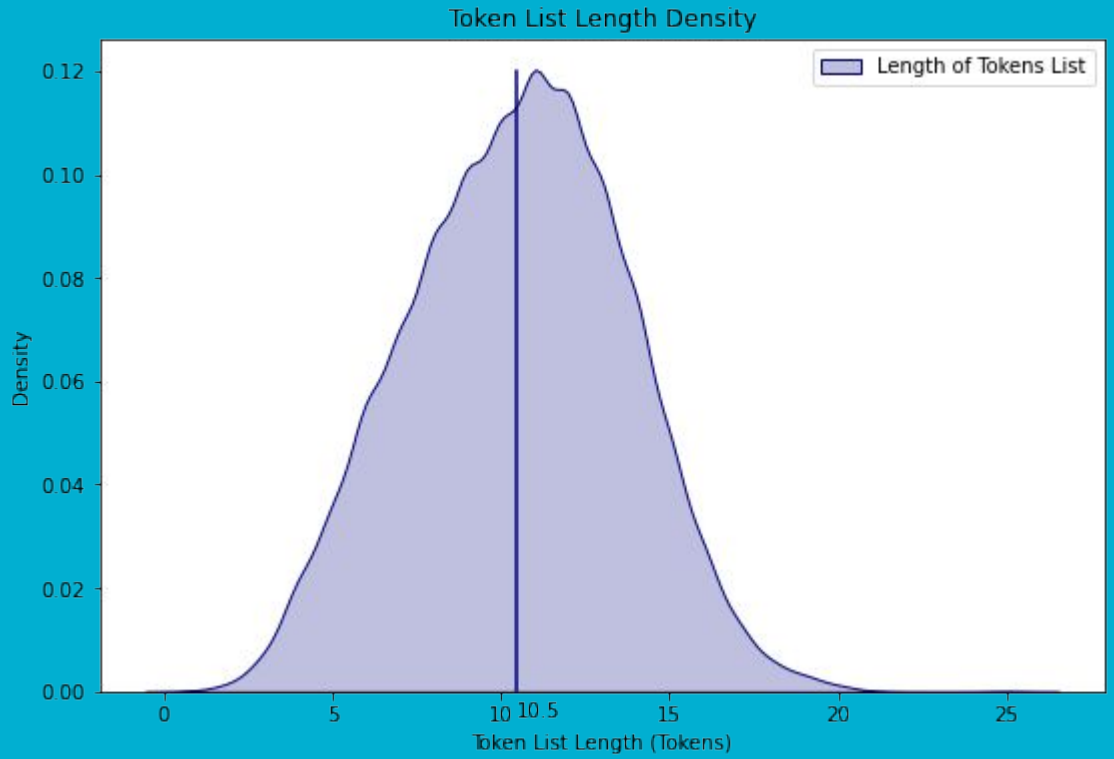
Company vs Sentiment (contd.)



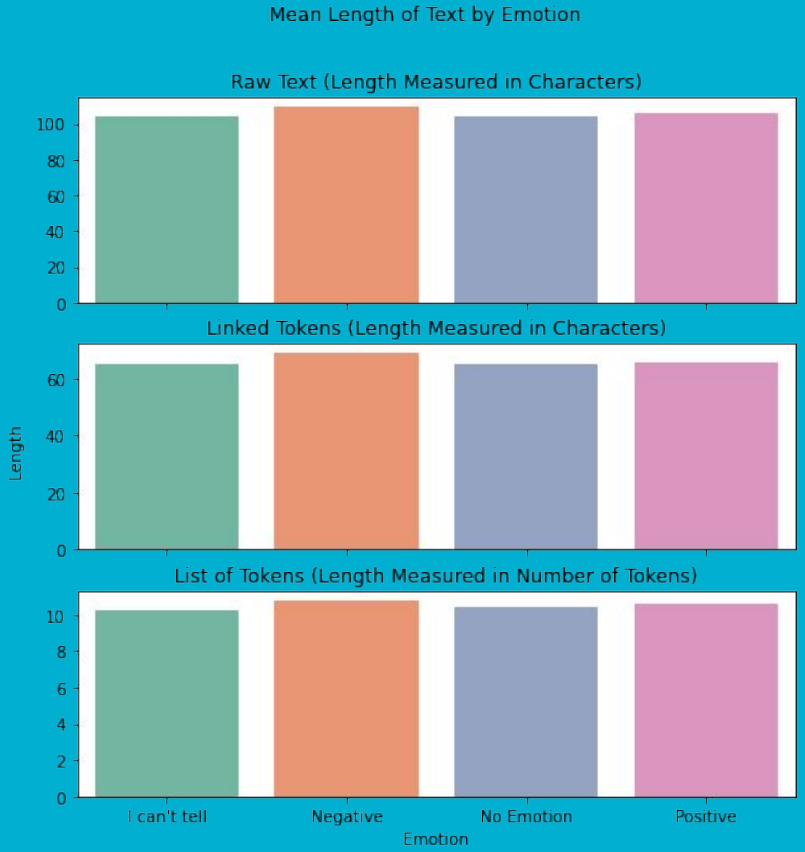
Text Length vs Sentiment – Text Length Distributions



Text Length Distributions (contd.)



Text Length vs Sentiment (contd.)



Modeling

1. Train/Test Split
2. TF-IDF Vectorizer
3. Support Vector Machine, Naive-Bayes, Random Forest
4. Create ensemble with cross validation
5. Evaluate

Model Evaluation

SVM performed best of three individual models. F1-Score was used because of class imbalance in target variable

SVM

- Train: 0.8838
- Test: 0.6579

Naive-Bayes:

- Train: 0.6962
- Test: 0.5922

Random Forest

- Train: 0.9919
- Test: 0.6380

Ensemble Cross Val

- Test: 0.6393

Conclusion and Next Steps

- Highest performer was SVM, though faith lies with ensemble
- Next Steps:
  - Lemmatization
  - GridSearch for model params
  - Cross validate all models
  - Word Embedding
  - VADER