Capstone Report

Necky Tran

2023-04-11

Introduction

Sentiment analysis is a natural language processing approach used to identify the emotional tone behind a body of text. It is utilized by organizations to categorize opinions about a product, service or idea. Generally, these tools analyze text data from online sources such as product reviews, blog posts and forum comments to determine whether customers liked or disliked a product.

For the Brainstation Capstone Project, language models predicting sentiment were created to identify the emotional tone of Reddit comments. Video game companies such as Valve and Riot constantly release new balance/patch notes for their online competitive video games which are often discussed in their communities' subreddits. Sentiment models can be used here to identify the community's overall reaction to these gameplay changes in order to build a better player experience which would in turn generate more revenue.

Dataset - GoEmotions

The NLP research community has made several open source datasets for the purposes of emotion/sentiment classification. However, they were relatively small and only focused on 6 main emotions. A Google Research team created a large-scale dataset (GoEmotions) that covered a more extensive set of emotions with the hopes that it could be used for a broader scope of future potential applications.

GoEmotions is a dataset of 58k unique Reddit comments extracted from popular English-language subreddits and labeled with 27 emotional categories. The dataset was created from Reddit comments from 2005 to 2019. Curation measures were applied to remove internet biases of extensive offensive language and to limit each comment length to 3-30 tokens. Each comment was scored for different emotions by multiple reviewers who were native English speakers from India.

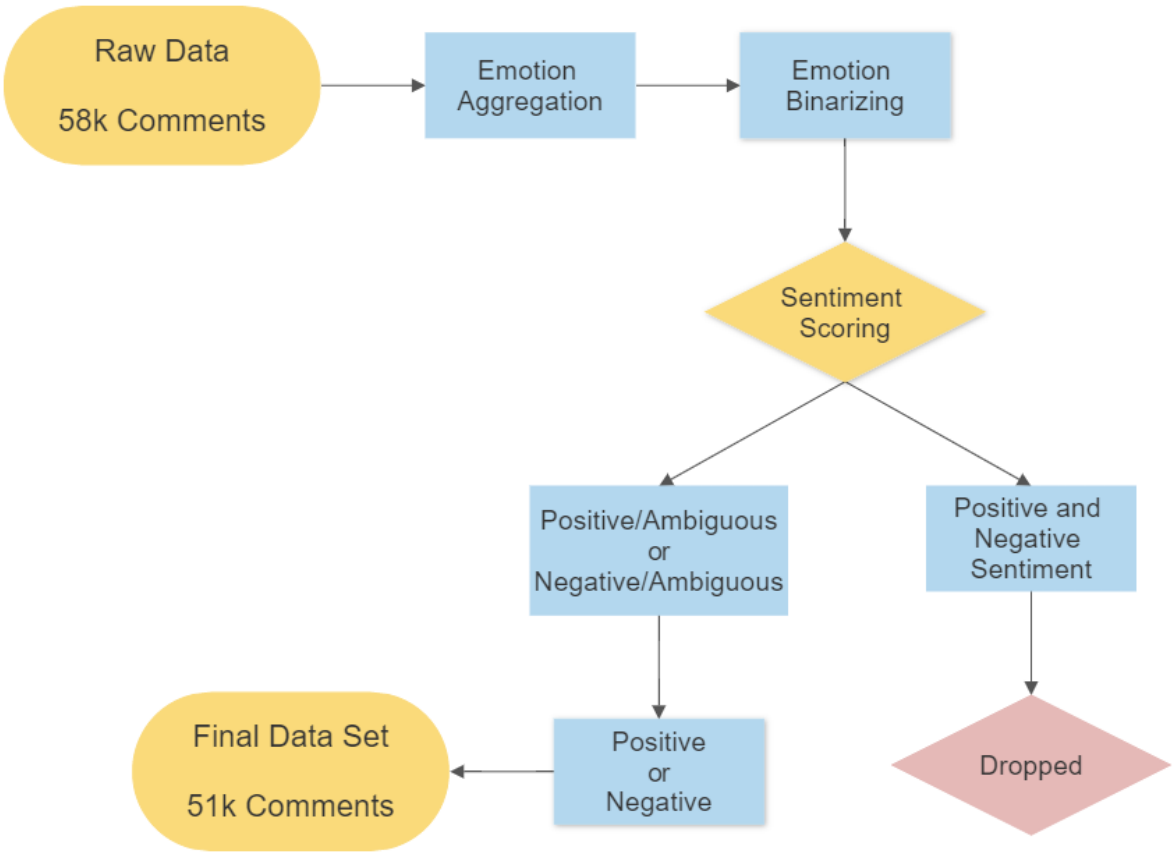
Positive		Negative		Ambiguous
admiration 👋	joy 😃	anger 😡	grief 😢	confusion 😕
amusement 😂	love ❤️	annoyance 😒	nervousness 😬	curiosity 🤔
approval 👍	optimism 🤞	disappointment	remorse 😔	realization 💡
caring 🤗	pride 😌	disapproval 👎	sadness 😞	surprise 😲
desire 😍	relief 😅	disgust 🤮		
excitement 🤩		embarrassment 😳		
gratitude 🙏		fear 😨		

27 Emotion categories in the dataset, separated by Sentiment

Wrangling

The raw dataset contains 211k observations and 58k unique comments. Each observation is a comment that is annotated for multiple emotions from a reviewer. The emotional scoring for each unique comment was aggregated and turned into a binary variable. The emotional scoring was only kept if multiple reviewers agreed upon the scoring. For example, if the emotional scoring for ‘admiration’ became a value of ‘2’ after aggregation, it was binarized to a value of ‘1’. This means that two reviewers needed to agree upon the emotion for it to have a value of ‘1’. Afterwards, the comments were given a sentiment (positive/negative/ambiguous) based on the emotional binarizing (a comment with a value of ‘1’ for ‘Joy’ would be assigned a ‘positive’ sentiment).

The final dataset was designed to have no overlap in sentiment. Comments with conflicting sentiments (positive and negative) were dropped from the dataset. Comments with a combinational sentiment of positive/negative and ambiguous were turned into positive or negative with the ambiguous sentiment being removed. For the purposes of sentiment analysis, comments scored for ambiguity were dropped from the dataset. The final dataset used for multiclass classification of sentiment contains 51,683 unique comments.



Data Wrangling Flowchart

Results

The balance of the target variable (Sentiment) for the final dataset was 40% positive, 38% Neutral, and 22% Negative. Classification models (Logistic Regression, KNN, Decision Tree, SVM) were trained on 75% of the final dataset leaving 25% as the testing set. Comment text was vectorized using a bag-of-words model. Model hyperparameters were optimized using gridsearch with a cross validation of 5. The Logistic Regression Model performed the best based on cumulative F1-scores and had an overall accuracy of 68% on the testing data.

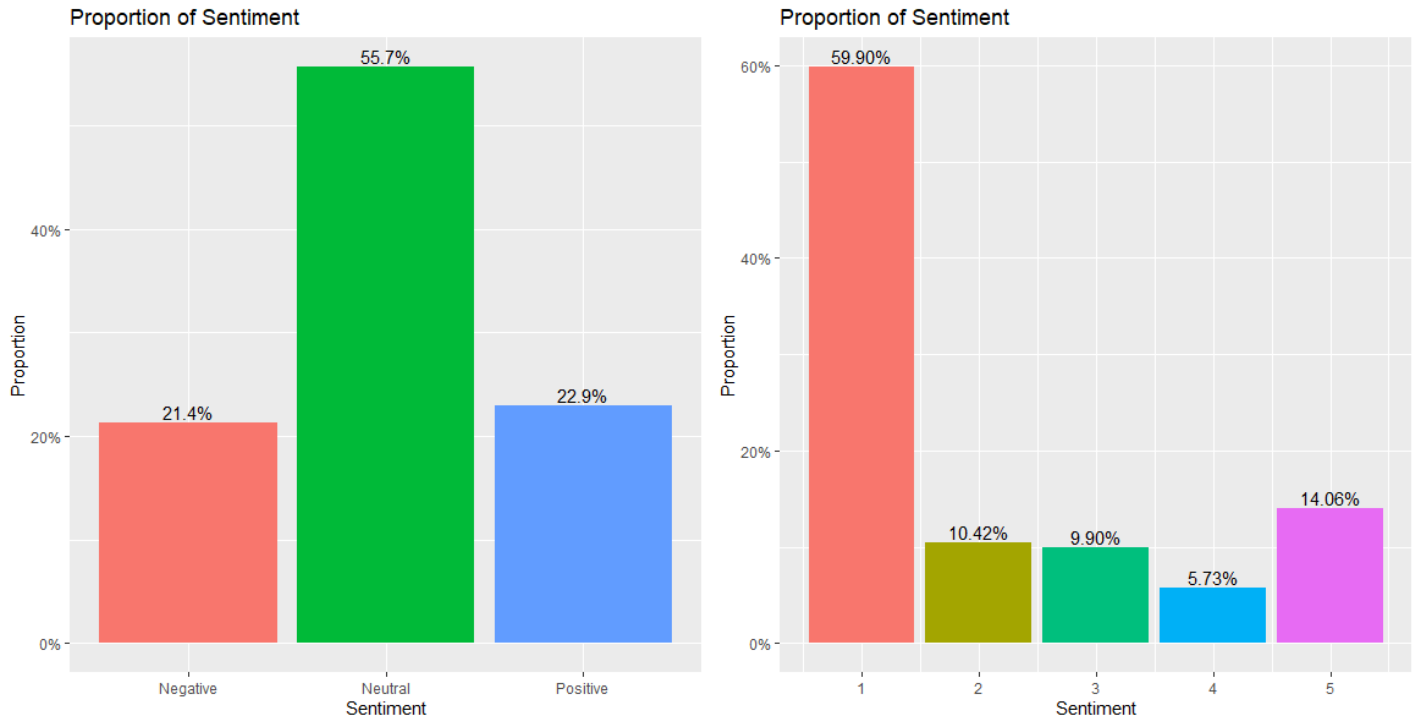
Logistic_Regression	Precision	Recall	F1-Score
Positive	0.80	0.72	0.76
Neutral	0.60	0.76	0.67
Negative	0.67	0.48	0.56

Insights

To gain insight into how the model performs on real life data, the trained models were used to predict the overall sentiment of scraped Reddit comments. The sentiment prediction of each model can be found on a hosted Shiny dashboard (https://ntran119.shinyapps.io/GoEmotions_Predictions/ (https://ntran119.shinyapps.io/GoEmotions_Predictions/)). 192 comments were scraped from a post encompassing comments of a gameplay patch for Dota2 (7.32e). Additionally, a pre-trained BERT model from hugging face was used to predict the sentiment of the same data to compare model performance.

From the results of the dashboard, we observed that the Logistic Regression model predicted that the scraped Reddit comments were ~30% positive. However, this does not seem to be the case when we look at the comments individually. It appears that the model does not take into account the context for how a word was used. For example, the model would predict a comment to always be positive if it encountered the acronym ‘lol’. Therefore, the model has a hard time correctly predicting negative sentiment if a positive-associated word is encountered.

In contrast to the GoEmotion models, the pre-trained BERT model did not have this same problem. It overall predicted the reaction of this dota patch to be mostly negative and does not automatically predict a comment to be positive when encountering the word ‘lol’. Additionally, the BERT model gives a rating of 1-5 for its predictions which can help distinguish a ‘great’ comment from a ‘good’ comment. It should be noted that the BERT model comes with its own pre-trained Tokenizer which is a transformer-based model and can account for context by understanding the sequence of words provided in text.



Sentiment Prediction (Left = Logistic Regression, Right = pre-trained BERT from HuggingFace)

Conclusion

Overall, the simplistic models trained on the GoEmotions dataset do not work well at predicting positive sentiment in real world cases. This is because the bag-of-words model does not care about the context of words. The pre-trained BERT model seemed to perform better overall on the scraped Reddit data, this might be due to it being trained on much more data and it being fine-tuned with 150k comment reviews which is three times larger than the GoEmotions dataset.

For potential next steps, it would be best to improve the model's ability to correctly predict negative sentiment. We can increase the overall balance of the data by bootstrapping negative comments that are already present in the dataset which can be done very quickly. An additional way to improve the model is by utilizing a recurrent neural network (RNN) NLP model that utilizes word embeddings. This would help our model to gain positional context for each word in a comment. However, for the purposes of sentiment prediction and NLP in general, it would be better to use a publicly available pre-trained model and fine-tune it rather than training one from scratch.