Twitter Sentiment Analysis

DSC 680 Applied Data Science

Bellevue University

Spring 2021

Conrad Ibañez

Abstract

Sentiment analysis is an important topic in data science and is very applicable in the real world. It is essentially a form of text classification where words in phrases are evaluated to determine their sentiment. Sentiment can either be positive, negative, or neutral.

We have observed how powerful and influential words and sentiment are in society. Sentiment can influence people's thoughts and actions. In today's digital world, there are numerous platforms and applications where people can express their thoughts. One of the most popular platforms is Twitter. We have seen how tweets can influence people to revolt against totalitarian governments and dictators, ignite activism and protests, and impact elections. In the business world, tweets have affected the stock market, cryptocurrencies, the price of goods such as collectibles, and corporate profitability.

Many companies have started to focus more attention on tweets to understand sentiment surrounding their business. Positive tweets are usually good signs, while negative tweets tend to indicate trouble and a bad perception. Bad perception oftentimes leads to tainted brand image and negative publicity causing poor performance. By monitoring Twitter data and predicting sentiment, entities can have a competitive advantage.

Background

Wherever there is text, some form of sentiment can be derived. Informational resources are usually neutral sentiment. However, whenever people write about their own perspectives or opinions causing bias, sentiment can usually be classified as positive or negative. We can sense positive or negative sentiment in newspaper or online editorials and essays, customer reviews, and Twitter feeds.

When we read about other people's thoughts or opinions, it oftentimes influences our own perspective and actions. Many of us have probably watched a Netflix movie or purchased an Amazon item based on positive customer reviews. We may have voted in the last election based on sentiment broadcasted from Twitter feeds or news outlets. Sentiment has a major impact on society today, and thus sentiment analysis has become increasingly important.

There has been much research into sentiment analysis, and there exist many algorithms for predicting sentiment. The overall goal is to analyze the meaning of words in phrases and derive the sentiment. A basic approach is to identify certain keywords to form a library that will identify whether phrases are positive, negative, or neutral.

Problem Statement

Twitter is a web platform where users can convey information including their thoughts and opinions. Users can subscribe to each other's Twitter feed, so they can all frequently receive messaging from each other. Millions of users tweet messages everyday including famous people, such as political leaders, actors and actresses, and professional athletes. Therefore, Twitter data is a good dataset to perform sentiment analysis.

This project will analyze Twitter data and complete sentiment analysis. It will investigate the different techniques and algorithms that can be used to determine whether a tweet is positive, negative, or neutral. This project will address various questions including the following:

1. What Twitter data is available for sentiment analysis?
2. What are the common keywords that are associated with positive sentiment?
3. Which are common for negative sentiment?
4. How do the different approaches for sentiment analysis work?
5. What are the benefits versus the disadvantages of the approaches?

6. How is the performance for sentiment analysis measured?
7. Does one approach perform better than the others?
8. How is the output of sentiment analysis used?
9. What issues or challenges exist with sentiment analysis?
10. What improvements can be done to improve identifying sentiment?

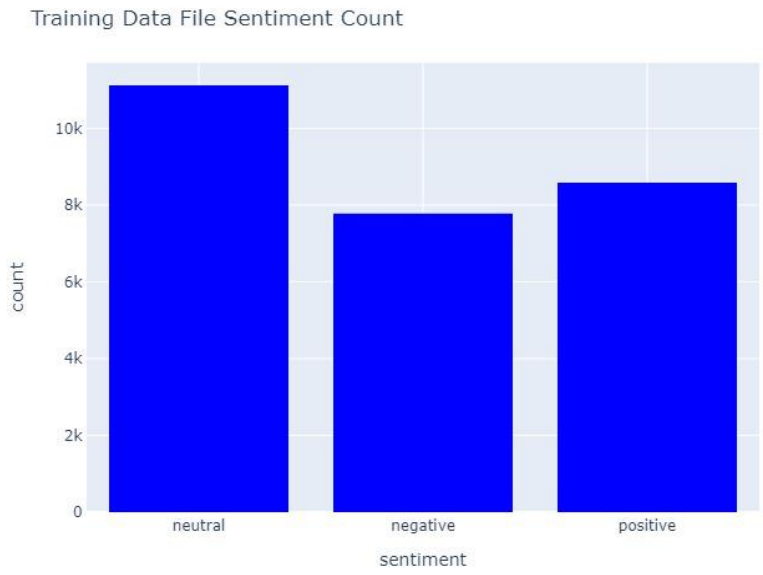
Data Understanding

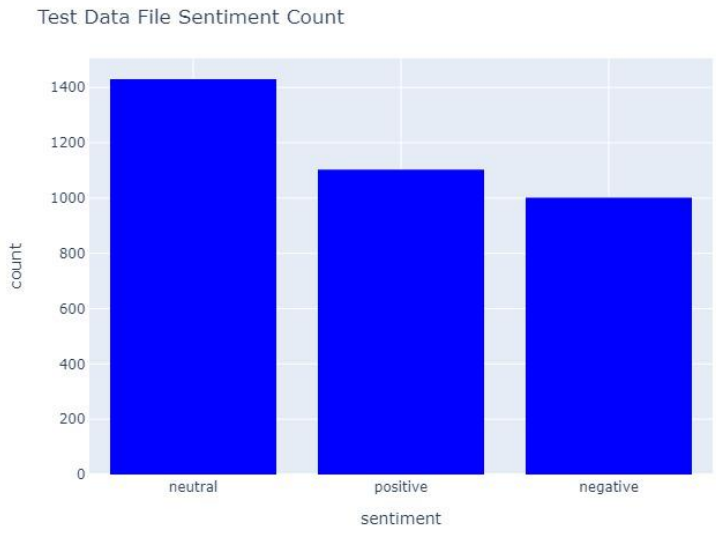
The dataset for this project can be found on Kaggle- https://www.kaggle.com/c/tweet-sentiment-extraction/data. It is the Tweet Sentiment Extraction dataset that was used as part of a Kaggle competition in 2020 that included over 2,000 teams and 37,000 entries. The dataset includes three csv files: a train file, a test file, and a sample submission file. The train and test files both contain a list of tweets that have been classified as positive, negative, or neutral.

Methods

Sentiment analysis utilizes natural language processing (NLP). In this graduate program, we only touched briefly on NLP and did some basic tokenization with data, so I used the information and code from this website- https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk - as the foundation for my sentiment analysis application written in Python. The code uses the Natural Language Toolkit (NLTK) package and WordNet as the lexical database for the English language. NLTK includes classes and functions that lemmatizes or groups together different forms of the same word.

We first explore the data and analyze the number of tweets that fall into the different categories of sentiment. There is sufficient amount of tweets for each of the categories for this project.





The application built in Python was able to identify some of the top words that signal the three different sentiments from the Twitter data:

Positive

[('good', 950), ('love', 795), ('happy', 733), ('day', 476), ('thanks', 438), ('great', 370), ('fun', 287), ('nice', 268), ('hope', 247), ('awesome', 234)]

Negative

[('miss', 525), ('sad', 347), ('sorry', 294), ('bad', 294), ('..', 265), ('hate', 248), ('...', 230), ('feel', 219), ('get', 219), ('suck', 208)]

Neutral

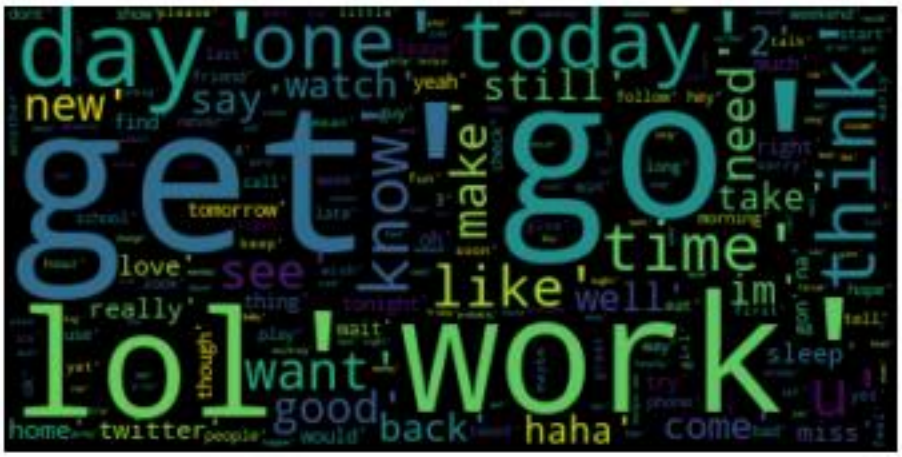
[('...', 1392), ('get', 1239), ('go', 1147), ('..', 638), ('day', 629), ('work', 625), ('like', 462), ('lol', 462), ('time', 458), ('know', 442)]

WordClouds present this information well.

Positive Train WordCloud

Negative Train WordCloud

Neutral Train WordCloud



Informative words from the dataset and the ratio among different sentiments included the following:

sad = True negati : positi = 151.5 : 1.0
thank = True positi : negati = 109.1 : 1.0
hate = True negati : positi = 105.2 : 1.0
suck = True negati : positi = 92.0 : 1.0
awesome = True positi : negati = 84.0 : 1.0
http = True neutra : positi = 76.0 : 1.0
stupid = True negati : positi = 65.8 : 1.0
fail = True negati : positi = 51.1 : 1.0
beautiful = True positi : negati = 46.2 : 1.0
till = True neutra : positi = 44.0 : 1.0
amazing = True positi : negati = 42.6 : 1.0
thanks = True positi : negati = 41.7 : 1.0
sick = True negati : positi = 40.8 : 1.0
love = True positi : negati = 40.5 : 1.0
hour = True neutra : positi = 35.4 : 1.0
hurt = True negati : positi = 34.6 : 1.0
poor = True negati : positi = 33.7 : 1.0
close = True neutra : positi = 33.2 : 1.0
lose = True negati : positi = 32.9 : 1.0
mother = True positi : negati = 32.6 : 1.0

Most of the informative words on the list makes logical sense. You would expect negative tweets to contain such words as “sad”, “hate”, and “suck” since those are associated to bad language. On the other hand, “thank”, “awesome”, “beautiful”, and “amazing” are all

reflective of positive sentiment. There may be a little more difficulty identifying neutral sentiment since some of the main words like ‘http’, ‘till’, or ‘hour’ are not really associated with any kind of thoughts or feelings.

Results and Conclusion

The NaiveBayesClassifier was used to train on the training dataset which contained over 5,000 rows. It achieved an accuracy of about 71% when the training dataset was split into 80/20 train and test sets. When the full training set and the full testing set from the files were used, an accuracy of 78% was achieved.

There was an issue observed with my application after using the same classifier from the full training and full testing set to create a CSV file with the predicted value as an attribute. When computing the accuracy on the file contents, the accuracy was only 47% when I expected it to be the same as the 78% that was achieved previously. I am not sure what is causing this discrepancy.

I also generated a file that listed the rows of data where there was a mismatch between the actual sentiment value and the predicted value. Out of the 3,534 test entries, there were 1,878 incorrect predictions. That means only 1,656 predictions were correct which resulted in the 47% accuracy. Below are example tweets with the incorrect predictions.

Error: Actual= negative Predicted= neutral tweet= its at 3 am, im very tired but i can`t sleep but i try it

Error: Actual= positive Predicted= neutral tweet= All alone in this old house again. Thanks for the net which keeps me alive and kicking! Whoever invented the net, i wanna kiss your hair!

Error: Actual= negative Predicted= neutral tweet= I know what you mean. My little dog is sinking into depression... he wants to move someplace tropical

Error: Actual= positive Predicted= neutral tweet= _sutra what is your next youtube video gonna be about? I love your videos!

Error: Actual= positive Predicted= neutral tweet= http://twitpic.com/4woj2 - omgssh ang cute ng bby.!

There seems to be entries where certain keywords appear that should have resulted in a more accurate prediction. For example, ‘can’t’ and ‘kiss’ are words that are associated to negative and positive sentiment, respectively. There are entries with incorrect predictions that have ‘thank’ and ‘love’ which were already associated with positive sentiment, so I am not sure if there may be problems with the implementation.

I have only touched the surface of learning about sentiment analysis. There are other methods such as logistic regression, decision trees, and neural networks that can be used as well. I hope to get more exposure to this topic in the near future.

Acknowledgements

This project wraps up my degree program. Thank you Bellevue University, professors, and staff for your help and support!

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau. "Sentiment Analysis of Twitter Data." In the proceedings of Workshop on Language in Social Media, ACL, 2011"Sentiment Analysis: A Definitive Guide." https://www.aclweb.org/anthology/W11-0705.pdf

Daityari, Shaumik. "How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK)." https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk -

Jarzynski, Przemyslaw. "Twitter Sentiment Analysis in Python." https://towardsdatascience.com/twitter-sentiment-analysis-in-python-1bafebe0b566

Jones, Alan. "Sentiment Analysis of Tweets." https://towardsdatascience.com/sentiment-analysis-of-tweets-167d040f0583

Kouloumpis, Efthymios & Wilson, Theresa & Moore, Johanna. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG!. ICWSM. http://scholar.google.com/scholar_url?url=https://ojs.aaai.org/index.php/ICWSM/article/download/14185/14034&hl=en&sa=X&ei=l8ehYNf8B8OuywStv7XYCw&scisig=AAGBfm1476sXHjHwpTKM23TE968Gzrov1w&nossl=1&oi=scholarr

NLTK: Learning to Classify Text. Retrieved from https://www.nltk.org/book/ch06.html

Mogyorosi, Marius. "Sentiment Analysis: First Steps With Python's NLTK Library." https://realpython.com/python-nltk-sentiment-analysis/

Pascual, Federico. "How to Do Twitter Sentiment Analysis with Machine Learning." https://monkeylearn.com/blog/sentiment-analysis-of-twitter/

Paul, Sayak. "Python Sentiment Analysis Tutorial." https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python

Shwartz, Steve. "12 Twitter Sentiment Analysis Algorithms Compared." https://www.aiperspectives.com/twitter-sentiment-analysis/

Selvaraj, Natassha. "A Beginner’s Guide to Sentiment Analysis with Python." https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6

"Sentiment Analysis: A Definitive Guide." Retrieved from https://monkeylearn.com/sentiment-analysis/

Tweet Sentiment Extraction. Retrieved from https://www.kaggle.com/c/tweet-sentiment-extraction/data

Vu, Duong. "Generating WordClouds in Python."

https://www.datacamp.com/community/tutorials/wordcloud-python