CS410 Project Documentation

Twitter Sentiment Analysis for Brands to Understand Customer Opinion

Srikanth Reddy Pullaihgari – srp10@illinois.edu

Overview

This project analyses the sentiment towards a particular brand or company using the most recent twitter feed.

Implementation

Sentiment analysis here is the process of finding out if a given text is positive, negative, or neutral. The insights from the twitter feed are then quantified and presented in the form of overall percentage of positive, neutral and negative tweets. The key steps in implementing this are as follows:

1) Accept a brand or company name as input from the user
2) Scrape Twitter for the latest feed that includes the brand name. I have limited the feed to the last 1000 tweets for practical reasons. Scraping of a larger amount of data than that was adding a fairly higher overhead in terms of execution time
3) Clean the obtained from gleaning the feed by removing links, special characters such as hashtags, punctuations etc and tokenize the individual words
4) Complete the POS Tagging and Lemma of the words
5) Detect the language of the cleaned text and restrict the analysis to tweets in English (it will consider the tweet as long as the dominant language in the tweet is English)
6) Determine the subjectivity, polarity of the tweets and use them to gauge the sentiment. I have used the TextBlob library for this analysis
7) The application then produces the following as output:
    a. The input csv file with the tweets
    b. The cleaned tweets in an html format
    c. The sentiment scorecard with the positive, neutral and negative split of tweets in a html format
    d. Three graphs in png files– bar chart to highlight the sentiment polarity, a scatter plot of polarity vs subjectivity and finally a word cloud of the tweets

Installation and Usage

Please follow the following to run the software:

1) Clone the code from my github repository - https://github.com/srp10/CS410CourseProject

2) Navigate to the directory ‘code’ on your terminal after the cloning is completed
3) Complete the necessary installations as below
    a. Install python3
    b. Install snsscrape either using the library directly

       $ pip install snscrape

       or with the developer version

       $ pip3 install git+https://github.com/JustAnotherArchivist/snscrape.git
    c. Install Pycld2

       $ python -m pip install -U pycld2
    d. Install wordcloud

       $ pip install wordcloud
4) Run the command prompt:

   $ python3 sentiment_analysis.py
5) A prompt will appear requesting a brand name as input. Please input the name of a company
6) The output on the command prompt shows the quantified sentiment analysis as positive, negative and neutral along with a breakdown of actual number of tweets classified
7) The output files are generated in the output folder (within ‘Code’ directory):
    a. <brandName>_ text-query-tweets.csv
    b. <brandName>_ final_output.html
    c. <brandName>_scorecard.html
    d. <brand_name>_sentiment_bar_chart.png
    e. <brand_name>_polarity_subj_plot.png
    f. <brand_name>_wordcloud.png

Self-Evaluation

I set out to do a Twitter sentiment analysis based on brand name intended to get a snapshot of the customer sentiment around a brand at an instant in time based on Twitter conversations. I worked alone on this project and this turned out to be fairly more complex that I expected it to be. There are a few things I would improve upon if I had more time:

1) The user inrterface could have been improved upon for a much better experience. I would have likely wanted to condense all the output and graphs into a single html page that would be easier to consume
2) I used Textblob and restricted the tweets to English language as best as I possibly could. This resulted in polarities that didn’t generally lean towards neutral for most of the time. Despite this, I was wondering whether sorting of tweets into different polarities could be potentially improved. The ability of TextBlob to spot and interpret sarcasm seems limited. I would be interested in looking further beyond rule based sentiment analysis and exploring other methods in more detail, especially to account for some of the negative sentiment that is not always accurately captured

3) In addition to the word cloud and sentiment plot, I would definitely consider looking into aspect based analysis. This would help to group and show relevant related messages and display them which might be helpful in understanding key points of discussion on Twitter

Notes

For the input to the program, please use alpha numeric characters. The program will throw an error if any alphanumeric characters are encountered. For example, if the brand name is 'Moet & Chandon', please enter 'Moet Chandon'. Otherwise, an error message will appear.

For the most part, please use common names of brands. Words such as 'Inc' or 'Ltd' in names will still be run successfully though.

References

https://www.freecodecamp.org/news/python-web-scraping-tutorial/https://catriscode.com/2021/05/01/tweets-cleaning-with-python/https://www.geeksforgeeks.org/removing-stop-words-nltk-python/https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/https://towardsdatascience.com/4-python-libraries-to-detect-english-and-non-english-language-c82ad3efd430https://pypi.org/project/pycld2/https://pub.towardsai.net/scraping-tweets-using-snscrape-and-building-sentiment-classifier-13811dadd11d
https://www.geeksforgeeks.org/dropdown-menus-tkinter/

https://towardsdatascience.com/twitter-sentiment-analysis-in-python-1bafebe0b566