Twitter Sentiment Analysis in R

Comparing R Functions and Google Cloud Natural Language Sentiment Analysis to Human-ranked Sentiment Analysis

CKME136 Data Analytics Capstone Course

Farhan Zia – 500543185

Chang School of Continuing Education

Ryerson University

December 2018

Twitter Sentiment Analysis in R

– Final report

1.0 Abstract

1.1 Dataset

https://www.kaggle.com/vincela9/charlottesville-on-twitter

The dataset I've selected pulls Twitter following the 2017 Unite the Right rally, also known as the Charlottesville riots – a white supremacist rally in Charlottesville, VA on Aug 11-12, 2017. The rally and it's counter-protest eventually turned violent, making international headlines with many drawing negative attention to President Trump's remarks following the events.

The dataset pulls Twitter posts over a 4-day period following Trump defending his stance that there was "blame on both sides", which was widely viewed as an endorsement for the rally.

1.2 Project description

I plan to use the tweets from the above dataset to assess whether the general sentiment was positive, negative or neutral.

The challenge in developing an accurate predictor model is inherent in Twitter as a platform:

- Little to no user profile data is verified (e.g., user location can be extraterrestrial or fictional), meaning that individuals can post commentary seemingly without consequence.
- Tweets may consist of multimedia content, meaning that text analysis would be ineffective at assessing the sentiment of emoji, images, video or URL links.

- Often, commentary on Twitter is in response to other users' tweets or content in URLs; this makes it very difficult to understand whether the tweet content is true or sarcastic.
- Twitter is known to contain a high number of 'bots' as users, referring to computer-based programs which produce tweet content on trending topics, typically attempting to polarize users' sentiment.
- Tweet content often contains hashtags, which are frequently used as shorthand ways to express sentiment; however, as these are usually a combination of words (e.g., #impeachtrump frequently appears in the dataset), they are not readable by a text analysis function.

In order to identify an overall sentiment, I will need to isolate variables in the dataset. The most likely variables to begin with will be:

Figure 1.1 – Key variables in dataset

Name	Description	Importance
user_statuses_count	Number of tweets by the specific user	Low
followers_count	Number of followers for the specific user	Low
full_text	Actual tweet text – inclusive of all emoji, URLs and hashtags	High
created_at	The time the tweet was sent	Low
quoted_status_id	Indicates whether the selected tweet was created in reply to an existing tweet – the original "replied-to" tweet may or may not be captured as part of the dataset	Low

hashtags	What hashtags, if any, were applied to the selected tweet	Med

The most important of the above will be the actual tweet text since this will provide the basis for the text analysis. The other variables are present merely to provide context.

I will rely on R for majority of the data cleaning and calculations and, upon further analysis of the data and review of existing literature, may consider adopting regression analysis or Naive Bayes classification to help sort tweets.

1.3 Project link

All project files and final coding can be found on GitHub at the following link: https://github.com/fxxzxx/CKME136_Capstone

2.0 Introduction

2.1 Social media and text analysis

Social media has emerged as the go-to medium for individuals to source information and to express opinion. Increasingly, analysts are mining those opinions in order to A) influence future opinions and decision, or B) deduct insights on public opinion.

Text-based sentiment analysis leverages machine learning to determine opinion based on a predetermined lexicon or dictionary that aligns keywords with an opinion. This, however, raises blind spots in analysis:

1. Social media is frequently composed of multimedia – images, gifs, videos, emoticons

- Emoticons may be used in place of text to express emotion
- Images or gifs may be used as memes – pop cultural references that often carry commentary

2. Context is key, for example:
    - Timing of a social post may occur before, during or after an event
    - Social posts are often conversational; argumentative statements may be misclassified
    - Text-based analysis does not account for conversational triggers such as sarcasm
3. Twitter-use norms
    - Informal language and slang-use
    - The use of hashtags to promote tweets and opinions
    - Most tweets tend to be neutral commentary and/or open-ended questions
    - Anonymity and freedom of consequence
    - Fake or bot-run accounts are frequent on Twitter[1]

The above are gaps that current literature is still searching to find ways to address.

Understanding that there are several limitations with text-based sentiment analysis, my goal is to identify which analysis type produces the more accurate predictor of opinion – naive bayes algorithm or logistic regression.

[1] "Bots on Twitter share two-thirds of links to popular websites: Pew ...." 9 Apr. 2018, https://techcrunch.com/2018/04/09/bots-on-twitter-share-two-thirds-of-links-to-popular-websites-pew/. Accessed 2 Dec. 2018.

3.0 Literature Review

3.1 Summary

Figure 3.1 – Summary of literature review

Paper	Model	Accuracy	Objective
Like it or not	Naive Bayes, k-nearest neighbour, support vector machines	Moderate	Recommender
Twitter Sentiment Analysis	Python, web application	Low	Classifier
The Good, The Bad, and the OMG!	n-gram	High	Classifier
Tweet sentiment analysis with classifier ensembles	Naive Bayes, logistic regression support vector machines	High	Classifier

3.2 Like it or not[2]

This paper does a good job at breaking down the foundations of opinion mining and sentiment analysis, especially in explaining how an opinion is expressed and classified. However, this paper largely focuses on evaluating different models to Twitter sentiment analysis.

Of those reviewed, the authors posit that the Naive Bayes model produces strong results for microblogs, like Twitter, especially when combined with another model

[2] Giachanou, Anastasia, and Fabio Crestani. "Like it or not: A survey of twitter sentiment analysis methods." ACM Computing Surveys (CSUR) 49.2 (2016): 28.https://dl.acm.org/citation.cfm?id=2938640.

(k-nearest neighbours algorithm, for example). However, the strongest results of any of the models reviewed were that when combining 4 or more classifier models or algorithms.

In my opinion, this is an obvious conclusion – that the more types of analysis performed on a dataset would show stronger accuracy.

3.3 Twitter Sentiment Analysis[3]

This paper relies on Python to analyse tweet sentiment on retail customer brand perspectives. A major limitation of this paper is that 84% of the tweets were given a null opinion. It also relies on the output to be a web application that requires a Linux server or LAMP to work, resulting in the paper being unable to realize a completed analysis.

The authors attempted to match words from the lexicon dictionary to the tweets in a JSON file, which they were unable to do because they tries to assign a value to each word in a tweet. An easier approach may have been to simply search for trigger keywords and phrases (e.g., “like” = positive, “hate” = negative) in the dictionary and classify tweets based on the inclusion of one of these words.

3.4 The Good, The Bad, and the OMG![4]

This paper also attempts to use a lexicon-based method to analyse Twitter data. However, this paper attempts to account for some of the blind spots I mentioned in my introduction – for example, emoticon use or hashtag use.

---

[3] Sarlan, Aliza, Chayanit Nadam, and Shuib Basri. "Twitter sentiment analysis." Information Technology and Multimedia (ICIMU), 2014 International Conference on. IEEE, 2014.https://www.semanticscholar.org/paper/Twitter-sentiment-analysis-Sarlan-Nadam/174533a803f512279f9e811259d8b96eadde0448.

[4] Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." Icwsm 11.538-541 (2011): 164.https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2857/3251.

The authors actually conduct their analysis on three different datasets that separately analysis on hashtags, emoticons and text. Again, the authors conclude that the more information included (hashtags, plus text, for example), the more accurate the results.

If this type of analysis were performed on the same dataset, it may be useful to see how the results changed if analysis were conducted in steps – for example, first classifying texts based on hashtags, then on emoticon use, then on lexicon matching.

3.5 Tweet sentiment analysis with classifier ensembles[5]

The authors approach sentiment analysis as a classification problem and, again, look to combined classification models as an analytical framework. They also introduce the bag-of-words model to break down tweets into groupings of singular words and track frequency.

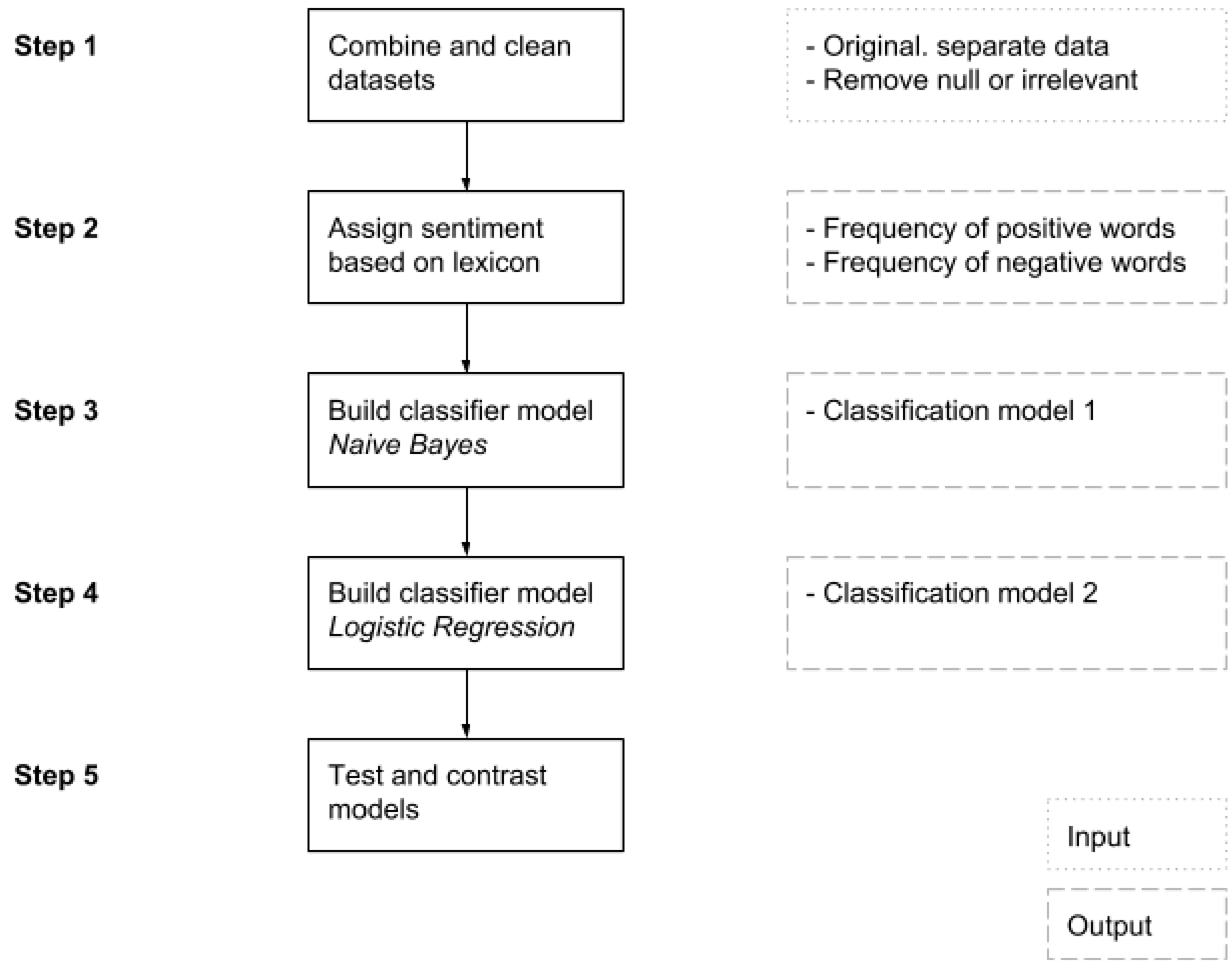
This is an interesting approach as the bag-of-words is often used for text classification of formal documents and applying this to tweets (which are very informal) actually proves to be highly accurate.

While the classification is accurate, the bag-of-words approach does rely heavily on frequency of words and this may overlook the context of a tweet and the informal nature of the Twitter platform.

[5] Da Silva, Nadia FF, Eduardo R. Hruschka, and Estevam R. Hruschka Jr. "Tweet sentiment analysis with classifier ensembles." Decision Support Systems 66 (2014): 170-179. https://www.researchgate.net/profile/Eduardo_Hruschka/publication/264242096_Tweet_Sentiment_Analysis_with_Classifier_Ensembles/links/56ca632808ae11063709d934/Tweet-Sentiment-Analysis-with-Classifier-Ensembles.pdf.

4.0 Approach

Figure 4.1 – Approach diagram



4.1 Step 1: Combine and clean datasets

Since the raw data is initially presented to compare tweets day by day, the 200,000 tweets are separated in four 50,000 sets. My first step will be to combine each of the datasets.

This does pose a challenge because, due to the nature of the subject matter of these tweets – the reactions after a protest, a violent anti-protest and a the US President's remarks after the fact – the definition of what is sympathetic or unsympathetic may change over the course of the days.

I will also have to clean the data of any unusable tweets, for example:

- Tweets in languages other than English
- Tweets that are without text, for example:
- Tweets that are only retweets (without the addition of new text)
- Tweets that only contain images, gifs, videos or links
- Tweets that are only emoticons

4.2 Step 2: Assign sentiment based on lexicon

As mentioned above there is a definition concern that I still have to decide how to proceed. I will have to review the details of the events timeline before finalizing the decision to combine each of the four sets. I may have to consider keeping the four sets separate and redefining what qualifies for sympathetic/unsympathetic on a daily basis.

This may be more work but I can reconcile the different days by assigning a sympathetic/unsympathetic score. After scoring, i can then combine the total dataset for further analysis. This process will require that I select a lexicon dictionary, some of which are below:

- WordStat
- Sentiwords
- SenticNet

Each of the above assigns sentiment in a slightly different way – simple positive/negative or multiple degrees/scores of positivity or negativity along a scale.

Alternatively, I may have to develop a lexicon dictionary for this specific project, combining terms from the above lexicon(s) that would specifically outline what terms would qualify as sympathetic/unsympathetic. Again, this may be difficult to implement but it would help clarify what qualifies for either classification ahead of time in order to eliminate any possible errors in subsequent steps.

I am still building an understanding of this section and intend to seek assistance on how best to approach the above.

4.3 Step 3: Build classifier model – Naive Bayes

As a classification system, each tweet will need to be categorized as sympathetic or unsympathetic. Using bi/multinomial Naive Bayes, I can determine the effective probability. This would operate based on a frequency count, similar to the bag of words scenario presented by Da Silva[4] above.

I am still building an understanding of this section and intend to offer further insight on the process moving forward.

4.4 Step 4: Build classifier model – Logistic Regression

The regression model, as I understand it from the literature above, will assign an overall score to each tweet based on the appearance of sympathetic/unsympathetic language. This essentially works as a tally system of the positive/negative terms, then assigns a category based on the greater sum of terms.

I am still building an understanding of this section and intend to offer further insight on the process moving forward.

4.5 Step 5: Test and contrast models

In the above sections, I've noted the various challenges that I'll face, especially regarding clarifying and classifying the dictionary lexicon and what terms qualify as sympathetic/unsympathetic.

One possible option to help justify and test any assumptions I make in the above is to test my lexicon against a pre-processed dataset such as those available on Kaggle[6].

Comparing the output scores of my model(s) versus those included with the dataset would help provide support for a major foundation of my project.

Moving forward from there, I would be able to run tests and full analysis on my own data.

5.0 Results and discussion

Visit the link below to view or download all relevant project files. All coding referenced can be found in the R markdown file "CKME136_Capstone.rmd".

GitHub link to project: https://github.com/fxxzxx/CKME136_Capstone.

5.1 Adjustments

Due to the difficulty finding sufficient literature that builds sentiment analysis models specifically in R, I've had to make several adjustments from my original proposals.

Increasingly, literature (as shown above) frequently relies on a mixture of programming languages to perform text-based sentiment analysis. The more interesting and flexible

[6] "Twitter sentiment analysis | Kaggle." 25 Feb. 2018, https://www.kaggle.com/imrandude/twitter-sentiment-analysis. Accessed 2 Dec. 2018.

models – recently being applied to emerging research such as emoji sentiment analysis – employ a mixture of python and other languages. These leverage natural language processing, Python CLIPS and NLTK and other techniques to conduct their analysis.

Unfortunately, I do not have a background in python or the associated machine learning techniques that can be easier to apply in that language. As a result, I've had to think of alternative ways to approach both sentiment analysis text classification and how to bring a new perspective to this field.

As such, I've pivoted to investigate how often an R-based sentiment analysis function will agree with a human's interpretation of sentiment and with a turnkey product's solution to sentiment analysis.

Additionally, I've had to pivot away from trying to assess degrees of sympathy in tweets as. Due to both a lack of skills in multiple coding languages (where creating a dictionary would be easier) and a limited amount of time (as creating a new dictionary to assess sympathy would require significant research), I would not have been able to deliver an effective result by the assigned due date of this project.

The initial proposal for this project was to categorize tweets as sympathetic or unsympathetic. This, however, has proved to be too challenging as the language is similar among tweets with competitive opinions. For example, there are tweets that are sympathetic to the white supremacists and others that are sympathetic to the anti-protestors but may share majority of the same language and would both be categorized by a dictionary as sympathetic.

Instead, I've opted for a more standard approach to sentiment analysis applied in existing dictionaries: reviewing high-level positivity or negativity of the overall text.

5.2 Human scored sentiment

Reading through existing literature, text-based sentiment analysis focuses on the capabilities and achievements of machine learning.

The machine-based function utilizes a dictionary lexicon in order to categorize each individual word in a tweet as either negative or positive, then assigns a category for the overall tweet based on the greater amount of (positive or negative) words. Of course, there are variations in approach, although the major difference tends to be in the dictionary lexicon used – whether sentiment words are assigned using a small number of categories (i.e., positive, negative or neutral) or a large number of categories (e.g., strongly positive, moderately positive, somewhat positive, neutral, etc.).

However, I haven't seen much in comparison to a sentiment score as evaluated by a human. Accordingly, I personally reviewed the first 500 tweets of the dataset, classifying the overall sentiment of each tweet – keeping in line with my 3 categories of positive, negative or neutral.

There are major differences here in how a human would categorize versus a machine, including:

- Assessing and categorizing tweets as a whole, rather than by individual words
- Understanding and accounting for sarcasm or references to popular culture
- Contextualizing commentary – e.g., based on a reply to a prior tweet or a quoted tweet
- Human bias – my own personal interpretation of tweets (e.g., what did the author intend)
- Human bias – my own stance on the issue discussed (e.g., when authors discuss polarizing topics such as white supremacy or political views)

- Understanding and processing hashtags

The first point above is the major gain by human scoring. While machines are currently capable of only using individual words to assess the overall sentiment of a tweet, a human can interpret the entire tweet at once.

This last point is also critical to sentiment analysis categorization as a machine also struggles to interpret hashtags. Some literature has attempted to create functions to decypher hashtag meaning or to create a dictionary of hashtags and their meanings. However, hashtag use can vary among users: some users create their own hashtags to create engagement[7], some others may misspell a popular hashtag (inadvertently, creating a new hashtag)[8], users and bots frequently use popular hashtags to promote their tweets or their own profiles[9], etc. In either case, this is knowledge gained to provide additional context to understanding the true sentiment of a tweet.

The intent of including a human-ranked sentiment score is to compare how often this score aligns with that of my R function.

---

[7] "How to Make a Custom Hashtag to Build Engagement | ThriveHive." 28 Jun. 2016, https://thrivehive.com/how-to-make-a-custom-hashtag-to-build-engagement/. Accessed 2 Dec. 2018.

[8] "A misspelled #NetNeutality hashtag is topping Twitter - The Verge." 14 Dec. 2017, https://www.theverge.com/2017/12/14/16776552/misspelled-net-neutality-net-neutrality-hashtag-twitter. Accessed 2 Dec. 2018.

[9] "How To Effectively Use Hashtags To Grow Your Business Following." 28 Dec. 2017, https://www.forbes.com/sites/forbescommunicationscouncil/2017/12/28/how-to-effectively-use-hashtags-to-grow-your-business-following/. Accessed 2 Dec. 2018.

5.3 Google natural language API

As social media engagement continues to increase, as does the pervasiveness of data analytics. Increasingly, sentiment analysis is becoming more popular and large corporations are taking notice[10].

Google, known for its advancements in machine learning – branded as Google AI (https://ai.google/) – and deep learning techniques, is one of the corporations building products and tools for use by private institutions. These products are intended to bridge the knowledge and skills gap for organizations looking to better understand data, without necessarily having the skills to conduct the analytics on their own.

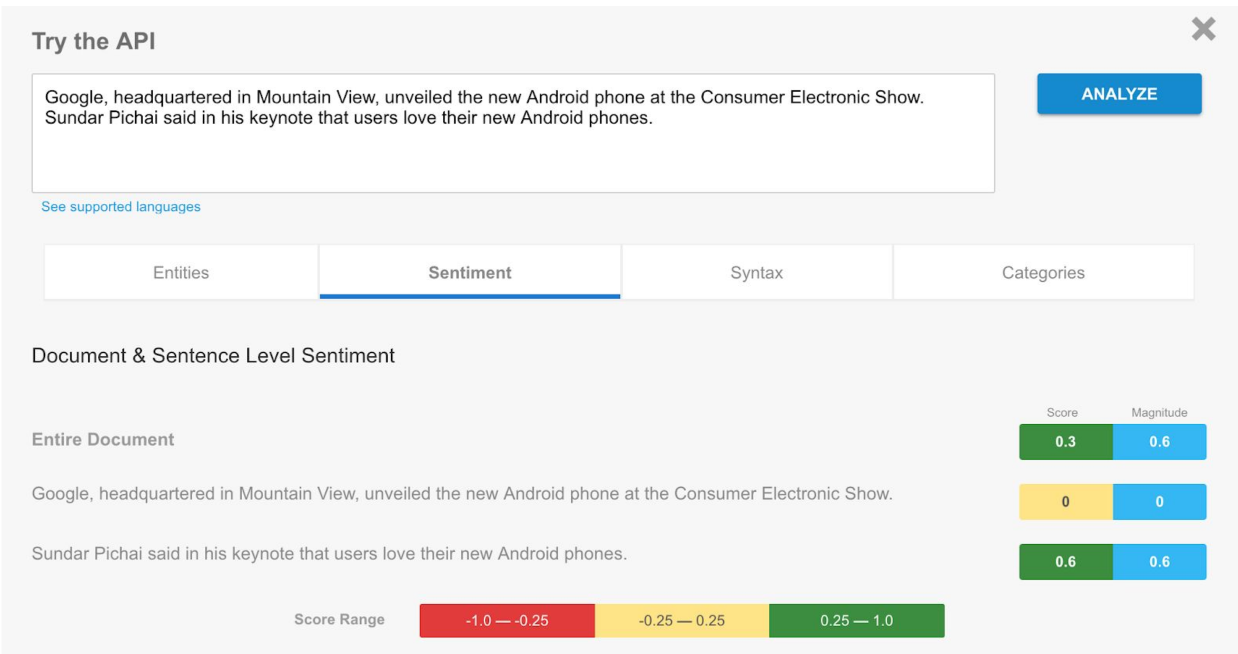
As such, Google has built a cloud-based tool for natural language processing of unstructured text segments. This product includes several processing capabilities, including:

- Object identification
- Sentiment analysis of:
  - Individual words,
  - Text segments, and
  - The entire text.
- Syntax dissemination (e.g., breaking down nouns, verbs, punctuation and references within the provided text)
- Content classification and relationships (e.g., what broad topics of discussion/interest the provided text references)

[10] "Why Sentiment Analysis Could Be Your Best Kept Marketing Secret." 30 Nov. 2018, https://www.forbes.com/sites/jiawertz/2018/11/30/why-sentiment-analysis-could-be-your-best-kept-marketing-secret/. Accessed 2 Dec. 2018.

Focusing on the sentiment analysis capabilities, the Google API breaks down the entire text into groups/strings of text and assigns sentiment for each section. Sentiment is assigned based on a range of positivity/negativity, allowing a range for neutrality in-between. The entire document is then assigned an overall sentiment. An example from Google is provided below:

Figure 5.1 – Google cloud natural language example



Each of the same 500 tweets that were assigned a human-ranked sentiment score were also processed using the Google API that you see above. The *Entire Document* sentiment score was used as an overall categorization of an individual tweet.

While Google assigns a range for each score, I was only concerned with whether the text was ranked as positive, negative or neutral. Each tweet's text was fed to the API in its entirety – including any emoji unicode, URLs, retweets or quoted tweets and hashtags. Based on the *Entire Document* score, that tweet was marked as either positive (1), negative (-1) or neutral (0), matching the same scoring scheme used in the human scored sentiment analysis.

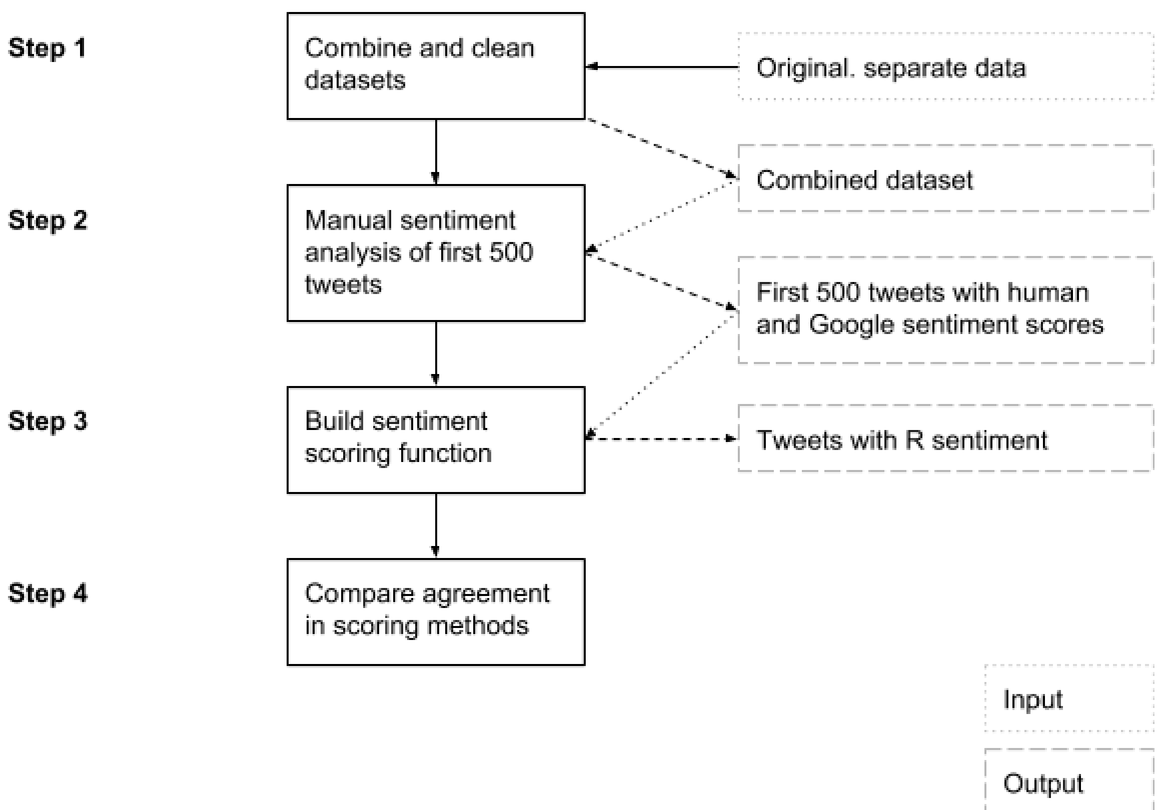
The intent of including a Google-ranked sentiment score is to compare how often this score aligns with that of my R function and the human-ranked score.

The Google cloud natural language API is available at the following link: https://cloud.google.com/natural-language/.

5.4 Revised approach

The figure below reflects the revised approach taken given the adjustments detailed above:

Figure 5.2 – Revised approach



The above figure represents my streamlined approach to this analysis project as described below:

- Step 1 remains largely the same, focused on data preparation.

- Step 2 now accounts for the work done with by human and Google-scored sentiment processes
- Step 3 is updated to reflect the R-based function I have developed and applied. I have elected to use the Bing[11] dictionary lexicon to perform the sentiment analysis and will detail this further below.
- Step 4 focuses on the comparison between the three sentiment analysis methods.

5.5 R function and the Bing dictionary

In order for my sentiment function to assess individual words, I've had to first tokenize the tweet text. To do so, I've followed the model outlined in a Kaggle tutorial by R. Tatman[12].

Following the same author's approach, I've used another Kaggle tutorial to help form my function for sentiment analysis[13], which relies on TidyText to perform text analysis. It is upon the recommendation of this tutorial that I've selected the Bing dictionary as the sentiment lexicon as, per my understanding, the Bing dictionary works well with TidyText's analysis tools.

I've modelled my R function on the example provided in the tutorial.

---

[11] "Opinion Mining, Sentiment Analysis, Opinion Extraction." https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. Accessed 2 Dec. 2018.

[12] "Tokenization tutorial | Kaggle." https://www.kaggle.com/rtatman/tokenization-tutorial. Accessed 2 Dec. 2018.

[13] "Data Science 101: Sentiment Analysis in R Tutorial | No Free Hunch." 5 Oct. 2017, http://blog.kaggle.com/2017/10/05/data-science-101-sentiment-analysis-in-r-tutorial/. Accessed 2 Dec. 2018.

5.6 Sentiment score comparison

Unfortunately, at the time of writing this paper, my code does not correctly function. This means that I do not have a way to effectively assess how each method performed.

I have written code to attempt to compare the output of the three sentiment analysis models using data visualization and linear regression to show significance in how the models differ or agree.

6.0 Conclusions

Conclusions are brief given the lack of output in my R code. Below are considerations for future work in this area.

Continue to explore crossover between human and machine learning.

> This paper attempted to provide insight to how machine learning sentiment analysis differed from how a human processed the same information. The hope here was to identify ways to improve how machine learning can closer resemble human thinking.

Additional research and accessibility is needed to understand intent in text analysis.

> Part of the reason my paper was limited and had to pivot its direction was that not enough work has been done to understand author's intent in text analysis. If this does exist, it is not accessible enough to be performed in the R environment.
>
> This was limiting to my ability to assess whether a tweet was sympathetic or unsympathetic to the anti-protestors as the subject of the tweet dataset.

7.0 References

1. "Bots on Twitter share two-thirds of links to popular websites: Pew ...." 9 Apr. 2018, https://techcrunch.com/2018/04/09/bots-on-twitter-share-two-thirds-of-links-to-popular-websites-pew/. Accessed 2 Dec. 2018.
2. Giachanou, Anastasia, and Fabio Crestani. "Like it or not: A survey of twitter sentiment analysis methods." ACM Computing Surveys (CSUR) 49.2 (2016): 28.https://dl.acm.org/citation.cfm?id=2938640.
3. Sarlan, Aliza, Chayanit Nadam, and Shuib Basri. "Twitter sentiment analysis." Information Technology and Multimedia (ICIMU), 2014 International Conference on. IEEE, 2014.https://www.semanticscholar.org/paper/Twitter-sentiment-analysis-Sarlan-Nadam/174533a803f512279f9e811259d8b96eadde0448.
4. Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." Icwsm 11.538-541 (2011): 164.https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2857/3251.
5. Da Silva, Nadia FF, Eduardo R. Hruschka, and Estevam R. Hruschka Jr. "Tweet sentiment analysis with classifier ensembles." Decision Support Systems 66 (2014): 170-179. https://www.researchgate.net/profile/Eduardo_Hruschka/publication/264242096_Tweet_Sentiment_Analysis_with_Classifier_Ensembles/links/56ca632808ae11063709d934/Tweet-Sentiment-Analysis-with-Classifier-Ensembles.pdf.

6. "Twitter sentiment analysis | Kaggle." 25 Feb. 2018, https://www.kaggle.com/imrandude/twitter-sentiment-analysis. Accessed 2 Dec. 2018.

7. "How to Make a Custom Hashtag to Build Engagement | ThriveHive." 28 Jun. 2016, https://thrivehive.com/how-to-make-a-custom-hashtag-to-build-engagement/. Accessed 2 Dec. 2018.

8. "A misspelled #NetNeutality hashtag is topping Twitter - The Verge." 14 Dec. 2017, https://www.theverge.com/2017/12/14/16776552/misspelled-net-neutality-net-neutrality-hashtag-twitter. Accessed 2 Dec. 2018.

9. "How To Effectively Use Hashtags To Grow Your Business Following." 28 Dec. 2017, https://www.forbes.com/sites/forbescommunicationscouncil/2017/12/28/how-to-effectively-use-hashtags-to-grow-your-business-following/. Accessed 2 Dec. 2018.

10. "Why Sentiment Analysis Could Be Your Best Kept Marketing Secret." 30 Nov. 2018, https://www.forbes.com/sites/jiawertz/2018/11/30/why-sentiment-analysis-could-be-your-best-kept-marketing-secret/. Accessed 2 Dec. 2018.

11. "Opinion Mining, Sentiment Analysis, Opinion Extraction." https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. Accessed 2 Dec. 2018.

12. "Tokenization tutorial | Kaggle." https://www.kaggle.com/rtatman/tokenization-tutorial. Accessed 2 Dec. 2018.

13. "Data Science 101: Sentiment Analysis in R Tutorial | No Free Hunch." 5 Oct. 2017,

http://blog.kaggle.com/2017/10/05/data-science-101-sentiment-analysis-in-r-tutorial/. Accessed 2 Dec. 2018.