Twitter Sentiment Analysis in R

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

---

Farhan Zia – 500543185

December 2018

Summary

- Text-based sentiment analysis, performed entirely using R
- Challenges for text analysis of Twitter data
  - Multimedia content – unicode emoji, URLs, embedded images
  - Linguistic challenges – assessing sarcasm, idioms and slang, popular culture references
  - Twitter use norms – replies and quotes, retweets
  - Hashtags – creation, use as expression, use for promotion, misspellings and errors
- Initial proposal – assessing degree of sympathy to a given topic
- Updated proposal – categorizing language in all tweets as positive, negative or neutral

Comparing sentiment analysis models

Human

Google

R

Literature review

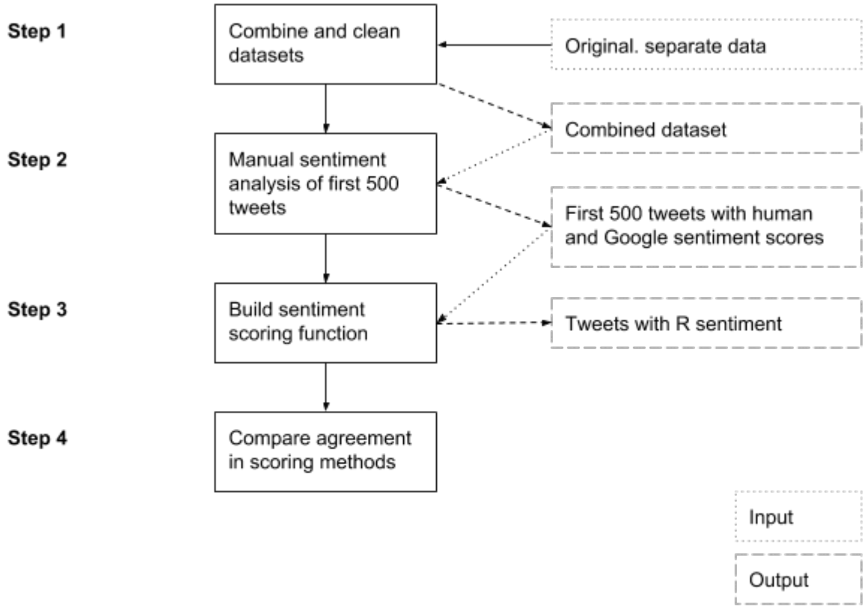
Paper	Model	Accuracy	Objective
Like it or not	Naive Bayes, k-nearest neighbour, support vector machines	Moderate	Recommender
Twitter Sentiment Analysis	Python, web application	Low	Classifier
The Good, The Bad, and the OMG!	n-gram	High	Classifier
Tweet sentiment analysis with classifier ensembles	Naive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

Figure 3.1 – Summary of literature review

Dataset

- 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
  - Follows Trump defending his stance that there was “blame on both sides”
  - Widely viewed as an endorsement for the rally
- Tweet text is the primary variable

Approach



Step 1
Combine and clean datasets
Original. separate data
Combined dataset
Step 2
Manual sentiment analysis of first 500 tweets
First 500 tweets with human and Google sentiment scores
Step 3
Build sentiment scoring function
Tweets with R sentiment
Step 4
Compare agreement in scoring methods
Input
Output

Sentiment scoring methods

Human

- Reviewed first 500 tweets of the dataset, classifying the overall sentiment of each tweet – positive, negative or neutral
- Differences between human and machine:
  - Assessing and categorizing tweets as a whole
  - Understanding sarcasm or references to popular culture
  - 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

Sentiment scoring methods

Google

- Object identification
- Sentiment analysis of: words, segments, 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)

Try the API

Google, headquartered in Mountain View, unveiled the new Android phone at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.

ANALYZE

See supported languages

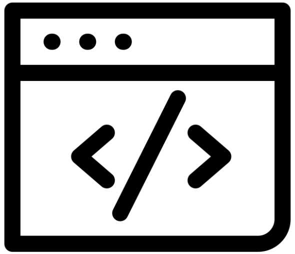
Entities | **Sentiment** | Syntax | Categories

Document & Sentence Level Sentiment

	Score	Magnitude
Entire Document	0.3	0.6
Google, headquartered in Mountain View, unveiled the new Android phone at the Consumer Electronic Show.	0	0
Sundar Pichai said in his keynote that users love their new Android phones.	0.6	0.6

Score Range -1.0 — -0.25 | -0.25 — 0.25 | 0.25 — 1.0

Sentiment scoring methods

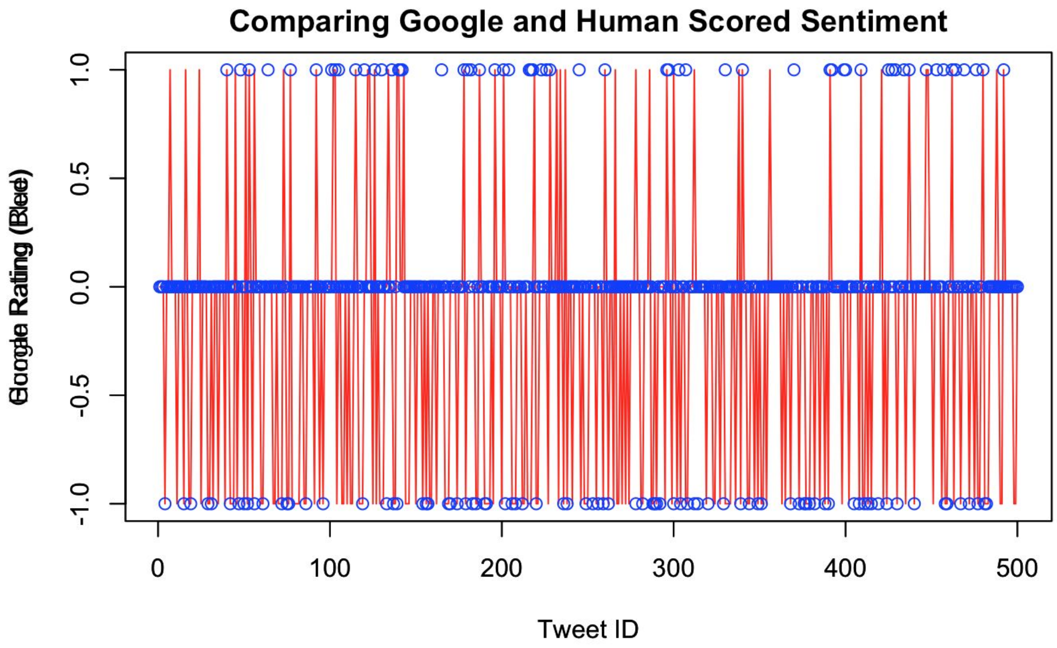


R

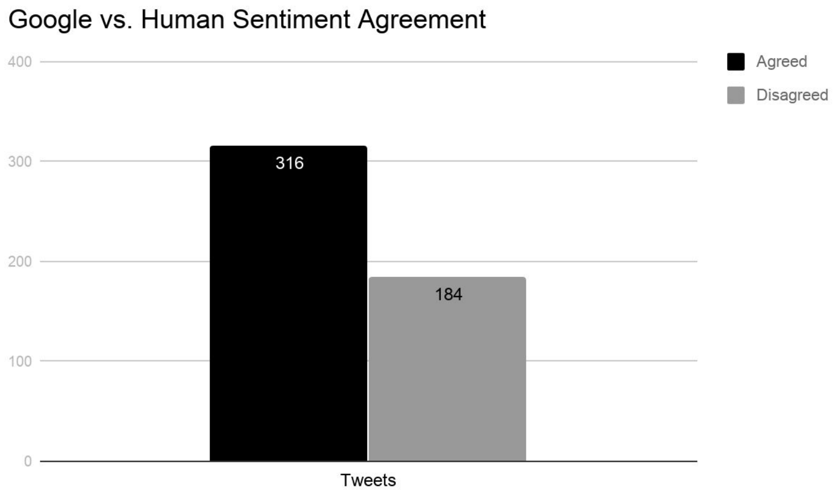
- Tokenize text
- Function compares text to Bing dictionary lexicon
- Each word matching lexicon is scored as positive, negative or neutral; overall tweet sentiment mathematically determined

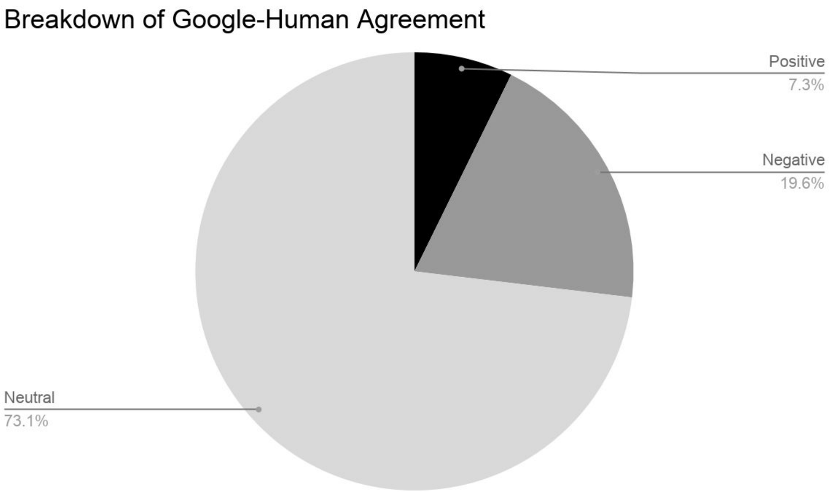
```r
# Function to assess sentiment of a single tweet by:
GetSentiment <- function(alldays_500){
  # Select a tweet, separate individual words and trim spaces
  senti_tweet <- glue(alldays_500$Tweet.Text, sep = "")
  senti_tweet <- trimws(senti_tweet)
  # Read the tweet text in a new file
  senti_text <- glue(read.file(senti_tweet))
  # Tokenize
  senti_tokens <- data.frame(text = senti_text) %>% unnest.tokens(word, Tweet.Text)
  # Run sentiment classifier function
  sentiment <- senti_tokens %>%
    # Extracting sentiment words from Bing dictionary
    inner_join(get.sentiments("bing")) %>%
    # Count positive and negative words
    count(sentiment) %>%
    spread(sentiment, n, fill = 0) %>%
    # Classify overall sentiment by identifying if there are more positive or negative words
    mutate(sentiment = positive - negative)
  # Add tweet ID
    mutate(ID = alldays_500$ID)
  # Classify sentiment as 1 = positive, 0 = neutral, -1 = negative to match Human_Rating and Google_Rating in
alldays file
    mutate(R_Rating =
             if (sentiment > 0) {print("1")}
              else if (sentiment < 0) {print("-1")}
                else if (sentiment == 0) {print("0")}
          )
  # Return sentiment dataframe
  return(sentiment)
}
```

Google vs. Human

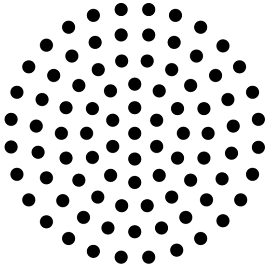


Google vs. Human

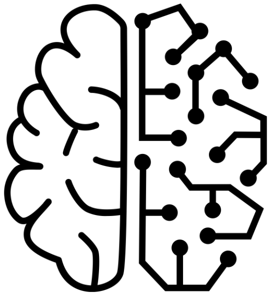




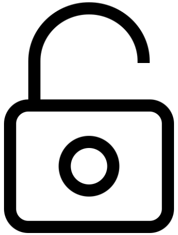
Conclusions



Select a stronger sample



Explore the cross-over between human and machine



Additional research and accessibility in text analysis

Thank you