# Intro to Text Analysis with R

January 23, 2016

By [Tal Galili](https://www.r-bloggers.com/author/rlover/)

One of the most powerful aspects of using R is that you can download free packages for so many tools and types of analysis.  Text analysis is still somewhat in its infancy, but is very promising.  It is estimated that as much as 80% of the world’s data is unstructured, while most types of analysis only work with structured data.  In this paper, we will explore the potential of R packages to analyze unstructured text.

R provides two packages for working with unstructured text – TM and Sentiment.  TM can be installed in the usual way.  Unfortunately, Sentiment has been archived in 2012, and is therefore more difficult to install.  However, it can still be installed using the following method, according to Frank Wang (Wang).

install.packages("devtools")

require(devtools)

install\_url("http://www.omegahat.org/Rstem/Rstem\_0.4-1.tar.gz")

install\_url("http://cran.r-project.org/src/contrib/Archive/sentiment/sentiment\_0.1.tar.gz")

install\_url("http://cran.r-project.org/src/contrib/Archive/sentiment/sentiment\_0.2.tar.gz")

The remaining required packaged can be installed as follows.

install.packages("plyr")

install.packages("ggplot2")

install.packages("wordcloud")

install.packages("RColorBrewer")

install.packages("tm")

install.packages("SnowballC")

Once initially installed, each can be loaded later as library(name).

The next step is to load the data.  I chose to download comments from a newspaper vent line (Charleston Gazette-Mail ).  This data was saved to a text file and loaded and processed as follows.

###Get the data

data <- readLines("https://www.r-bloggers.com/wp-content/uploads/2016/01/vent.txt") # from: http://www.wvgazettemail.com/

df <- data.frame(data)

textdata <- df[df$data, ]

textdata = gsub("[[:punct:]]", "", textdata)

Next, we remove nonessential characters such as punctuation, numbers, web addresses, etc from the text, before we begin processing the actual words themselves.  The code that follows was partially adapted from Gaston Sanchez in his work with sentiment analysis of Twitter data (Sanchez).

textdata = gsub("[[:punct:]]", "", textdata)

textdata = gsub("[[:digit:]]", "", textdata)

textdata = gsub("http\\w+", "", textdata)

textdata = gsub("[ \t]{2,}", "", textdata)

textdata = gsub("^\\s+|\\s+$", "", textdata)

try.error = function(x)

{

  y = NA

  try\_error = tryCatch(tolower(x), error=function(e) e)

  if (!inherits(try\_error, "error"))

    y = tolower(x)

  return(y)

}

textdata = sapply(textdata, try.error)

textdata = textdata[!is.na(textdata)]

names(textdata) = NULL

Next, we perform the sentiment analysis, classifying comments using a Bayesian analysis.  A polarity of positive, negative, or neutral is determined.  Finally, the comment, emotion, and polarity are combined in a single dataframe.

class\_emo = classify\_emotion(textdata, algorithm="bayes", prior=1.0)

emotion = class\_emo[,7]

emotion[is.na(emotion)] = "unknown"

class\_pol = classify\_polarity(textdata, algorithm="bayes")

polarity = class\_pol[,4]

sent\_df = data.frame(text=textdata, emotion=emotion,

                     polarity=polarity, stringsAsFactors=FALSE)

sent\_df = within(sent\_df,

                 emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

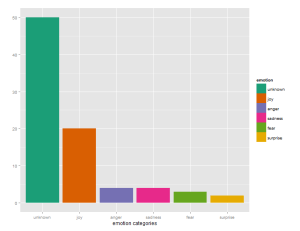
Now that we have processed the comments, we can graph the emotions and polarities.

ggplot(sent\_df, aes(x=emotion)) +

geom\_bar(aes(y=..count.., fill=emotion)) +

scale\_fill\_brewer(palette="Dark2") +

labs(x="emotion categories", y="")

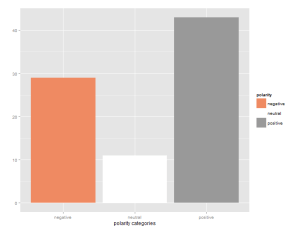
[](https://i1.wp.com/www.r-bloggers.com/wp-content/uploads/2016/01/distribution.png?ssl=1)

ggplot(sent\_df, aes(x=polarity)) +

  geom\_bar(aes(y=..count.., fill=polarity)) +

  scale\_fill\_brewer(palette="RdGy") +

  labs(x="polarity categories", y="")

[](https://i1.wp.com/www.r-bloggers.com/wp-content/uploads/2016/01/polarity.png?ssl=1)

We now prepare the data for creating a word cloud.  This includes removing common English stop words.

emos = levels(factor(sent\_df$emotion))

nemo = length(emos)

emo.docs = rep("", nemo)

for (i in 1:nemo)

{

  tmp = textdata[emotion == emos[i]]

  emo.docs[i] = paste(tmp, collapse=" ")

}

emo.docs = removeWords(emo.docs, stopwords("english"))

corpus = Corpus(VectorSource(emo.docs))

tdm = TermDocumentMatrix(corpus)

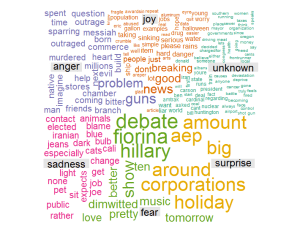
tdm = as.matrix(tdm)

colnames(tdm) = emos

comparison.cloud(tdm, colors = brewer.pal(nemo, "Dark2"),

                 scale = c(3,.5), random.order = FALSE,

                 title.size = 1.5)

[](https://i0.wp.com/www.r-bloggers.com/wp-content/uploads/2016/01/wordcloud.png?ssl=1)

What do we gain from this analysis beside an attractive word cloud?  We can analyze the word cloud itself.  The Sentiment package has identified the most frequently occurring, important words, and their likely association with emotions.  For instance, ‘guns’ was associated with anger, while ‘hillary’ was associated with fear.  ‘pet’ was associate with sadness, and ‘aep’ was associated with surprise.  With very little work, we have automatically extracted the important topics from the unstructured text.

More importantly, we also have a table of the comments themselves with the emotions and polarity attached.  If we desire, we can sort them by emotion or polarity and continue our analysis.  If this had been corporate satisfaction data, for example, we may want to dig deeper into angry comments and joyous comments for different reasons.  We may use this as a tool to intelligently select comments for Quality Assurance analysis rather than blind random selection.  Text and Sentiment Analysis may be in its infancy, but it is can also be the beginning for further analysis.

# References

* [Charleston Gazette-Mail . Readers Vent. 09 November 2015.](http://www.wvgazettemail.com/section/GZ0407&profile=1136)
* [Sanchez, Gaston. “Sentiment Analysis with “sentiment”.” 2012. Mining twitter with R. 09 November 2015.](https://sites.google.com/site/miningtwitter/questions/sentiment/sentiment)
* [Wang, Frank. Stack Overflow. 13 December 2013. 09 November 2015.](https://stackoverflow.com/questions/10233087/sentiment-analysis-using-r)