Homework\_Sonia

BIS581 Paul Dunn

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

#load libraries  
library(tidytext)  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(ggraph)  
library(igraph)

##   
## Attaching package: 'igraph'

## The following object is masked from 'package:tidyr':  
##   
## crossing

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

remove (twts) Load our data

#get data  
options(stringsAsFactors = FALSE)  
  
twts <- read.delim("tweet.txt", header=FALSE, stringsAsFactors = FALSE, sep='|')

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :  
## EOF within quoted string

sotu <- read.delim("part-m-00000.txt", header=FALSE, stringsAsFactors = FALSE, sep='|')

Sentiment Analysis is a technique we can use to try to calculate the emotion or tone of a text. The method I will demonstrate here uses a library called tidytext. There are three different lexicons available to you in this library. AFINN, bing, NRC AFINN categorizes words into positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. Bing categorizes words into positive and negative. AFINN assigns scores between -5 and 5. The easiest way to understand how this works using BING as an example is that each word in a sentence is assigned either -1 or +1 and then the sentence is totalled. If the result is negative, then the sentence has a negative emotion, while if the result is positive, then overall the sentence is positive. A zero would mean a neutral sentence. You can look at any of the lexicons as below (replace bing with afinn or nrc)

get\_sentiments("nrc")

## # A tibble: 13,875 x 2  
## word sentiment  
## <chr> <chr>   
## 1 abacus trust   
## 2 abandon fear   
## 3 abandon negative   
## 4 abandon sadness   
## 5 abandoned anger   
## 6 abandoned fear   
## 7 abandoned negative   
## 8 abandoned sadness   
## 9 abandonment anger   
## 10 abandonment fear   
## # ... with 13,865 more rows

Let’s get a single chunk of text from our data to work with. We’ll grab a single ContactMethod from our data. Then we split it up into individual words with a function called “unnest\_tokens” (don’t worry about what it is, just that it will break the sentence into individual words)

sotu\_rows <- sotu %>% mutate (h\_number = row\_number())  
sotu\_Tidy <- sotu\_rows %>% unnest\_tokens(word, V1)  
#unnest\_tokens also changes everything to lower-case.

Let’s take a look at the word list we now have:

sotu\_Tidy %>% count(word, sort = TRUE) %>% head(10)

## word n  
## 1 â 525365  
## 2 realdonaldtrump 154148  
## 3 rt 143013  
## 4 the 124934  
## 5 ã 91878  
## 6 to 86742  
## 7 https 76707  
## 8 å 76310  
## 9 tco 75935  
## 10 of 51362

Notice that many of the words are what we would call “stop words”, that is, at least for us, they don’t have any sentiment. Words like “if, an, the” so let’s remove them, then we’ll look at the list again:

sotu\_Tidy <- sotu\_Tidy %>% anti\_join(stop\_words)

## Joining, by = "word"

sotu\_Tidy %>% count(word, sort = TRUE) %>% head(10)

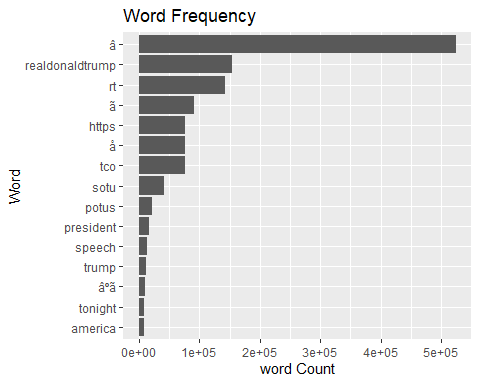
## word n  
## 1 â 525365  
## 2 realdonaldtrump 154148  
## 3 rt 143013  
## 4 ã 91878  
## 5 https 76707  
## 6 å 76310  
## 7 tco 75935  
## 8 sotu 41212  
## 9 potus 21568  
## 10 president 15950

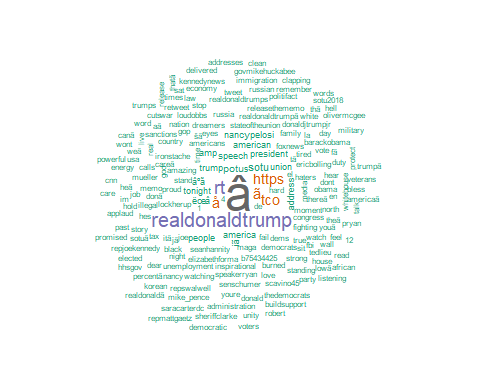
Great! Now, let’s look at the sentiment of our word list. First we’ll start with a very simple use of the lexicons. You will be able to see how it views each word. Try changing afinn to bing or nrc

sotu\_Tidy %>% inner\_join(get\_sentiments("nrc")) %>% head(10)

## Joining, by = "word"

## h\_number word sentiment  
## 1 1 question positive  
## 2 1 genius positive  
## 3 8 standing positive  
## 4 8 pretty anticipation  
## 5 8 pretty joy  
## 6 8 pretty positive  
## 7 8 pretty trust  
## 8 8 grim anger  
## 9 8 grim anticipation  
## 10 8 grim disgust

Let’s look at word clouds too since we’re at it. We’ll put all of a single column into a new data.frame. This command also removes all rows where Improvement is empty. 

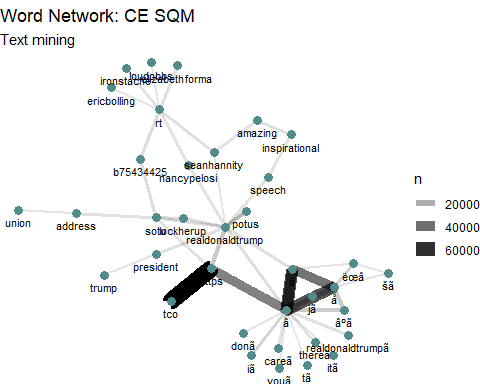
OK, there’s our bar, let’s make a word cloud: 

We did this based on individual words, now we can start using N-Grams. This means sequences of words, we can do bi-grams (2 words), tri-grams (3 words, etc.) If you think about Harry Potter and how it starts “The boy who lived” if we split this to bi-grams we get: the boy boy who who lived Let’s get started.

Let’s look at which bi-gram shows up most often (top 20)

## bigram n  
## 1 â â 241296  
## 2 https tco 75224  
## 3 ã â 50875  
## 4 å â 48863  
## 5 â ã 43457  
## 6 ã å 38248  
## 7 â https 34139  
## 8 â s 16904  
## 9 of the 16660  
## 10 â å 15650  
## 11 realdonaldtrump https 11113  
## 12 to the 10733  
## 13 â âºã 10481  
## 14 âºã å 10435  
## 15 potus realdonaldtrump 9863  
## 16 â t 9807  
## 17 the sotu 7832  
## 18 state of 7724  
## 19 in the 7701  
## 20 å ëœâ 7697

Let’s get rid of the common words since they don’t provide context

Plot it: 

Let’s try to remove the garbage words: We’ll make a custom list of “stop words” to remove and then use an anti-join to remove them.

custom\_stop\_list <- tibble(word = c("rt","realdonaldtrump","http","https","tco","amp","sotu","??","?","???","i?","?","potus"))  
sotu\_clean <- sotu\_Tidy %>% anti\_join(custom\_stop\_list)

## Joining, by = "word"

sotuWordCount <- sotu\_clean %>% count(word, sort=TRUE) %>% top\_n(15) %>% mutate(word = reorder(word,n))

## Selecting by n

ggplot(data=sotuWordCount, aes(x=word, y=n)) +  
 geom\_bar(stat="identity") + coord\_flip() + labs(x="Word",y="word Count",title="Word Frequency")

