KLonData

Text analysis of republican presidential debates

Here we analyze the republican presidential debate transcripts, focusing on three candidates in particular: Trump, Cruz and Rubio.

First let's answer the following question:

How often does Trump get mentionned by name by the other two candidates during a debate, versus how often do the other two candidates mention each other's name ?

The transcripts were downloaded from: http://www.presidency.ucsb.edu/debates.php (http://www.presidency.ucsb.edu/debates.php), from August 6th, 2015 to March 21st, 2016.

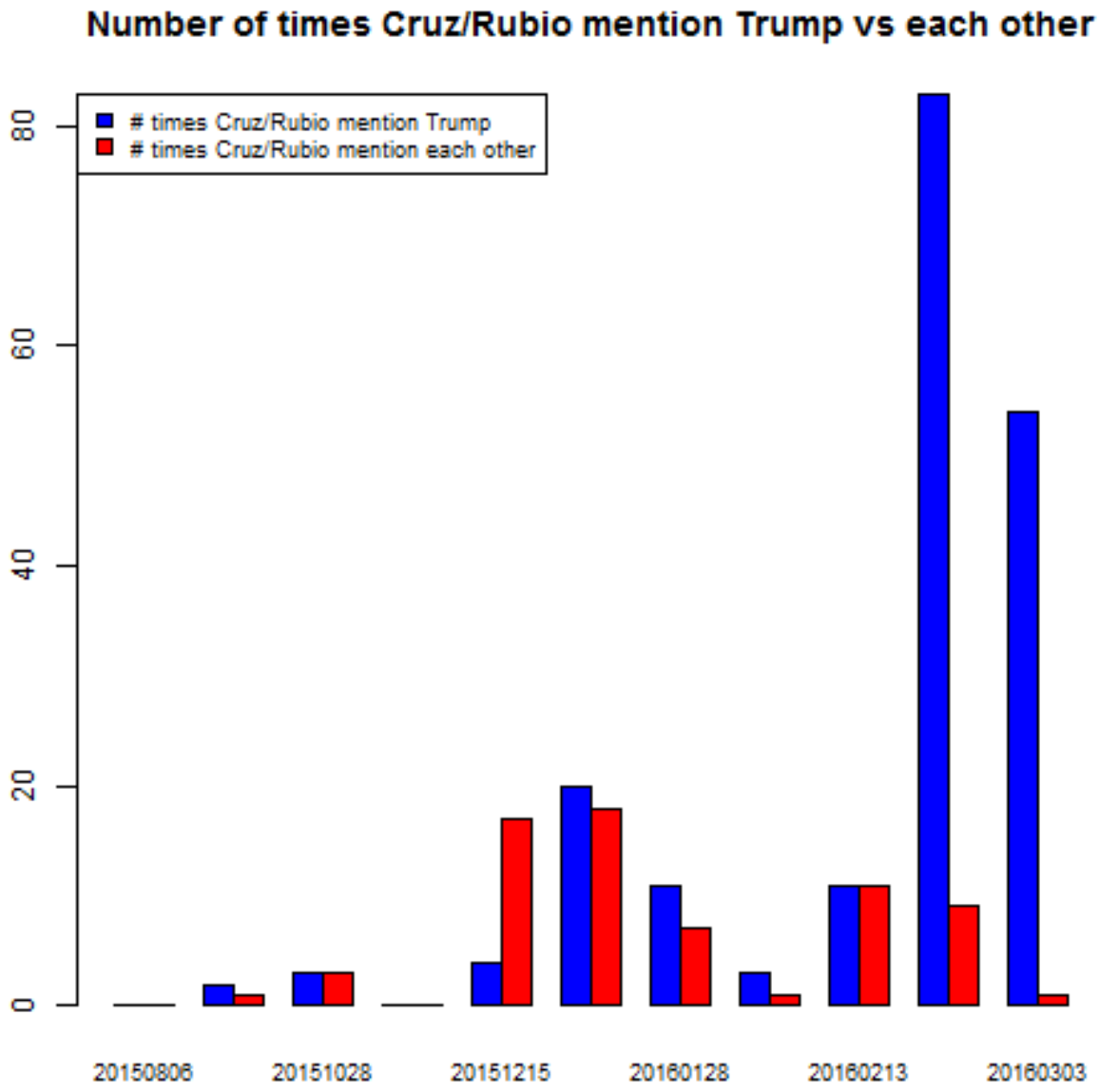
Here's the code:

```
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)

ref_matrix = function (date){

  #read character data
  text = scan(paste0('debate_', date, '.txt'), what='x', quote=NULL)

```

```
speakers =
  c(  CRUZ    = '',
      TRUMP   = '',
      RUBIO   = ''
  )

#assign text to the right speaker
for(word in text){
  #if word ends with :
  if(substr(word,nchar(word),nchar(word))==':'){
    #if word corresponds to one of the speakers of interest
    if(word %in% paste0(names(speakers), ':')){
      #set current speaker
      currentSpeaker = substr(word,1,nchar(word)-1)
    }
    else{
      #if the current speaker is not one of the speakers of interest, set it to NA
      currentSpeaker = NA
    }
  }
  else if(!is.na(currentSpeaker)){
    #if the current speaker is of interest, save what he is saying
    speakers[currentSpeaker] = paste(speakers[currentSpeaker], word)
  }
}

#preprocess text
prez = Corpus(VectorSource(speakers))
prez = tm_map(prez, tolower)
prez = tm_map(prez, removeWords, stopwords('english'))
#remove additional unwanted words
prez = Corpus(VectorSource(speakers))
prez = tm_map(prez, tolower)
prez = tm_map(prez, removeWords, stopwords('english'))
#remove additional unwanted words
prez = tm_map(prez, removeWords, c('will', 'going', 'applause', 'get', 'say', 'want', 'let', 'c
                                   'got', 'one', 'two', 'also', 'ever', 'even', 'need', 'every',
prez = tm_map(prez, removePunctuation, preserve_intra_word_dashes = FALSE)
prez = tm_map(prez, stemDocument)
prez = tm_map(prez, stripWhitespace)
```

```
  prez = tm_map(prez, removeNumbers)
  prez = tm_map(prez, PlainTextDocument)

  #make document term matrix
  dtm &amp;amp;amp;amp;amp;amp;amp;lt;- DocumentTermMatrix(prez)

  #reassign row names (each row is a speaker)
  rownames(dtm) = names(speakers)

  #how many times was donald trump referred to by other candidates
  names = character()
  if('donald' %in% colnames(dtm)){names = c(names, 'donald')}
  if('trump' %in% colnames(dtm)){names = c(names, 'trump')}
  dtm_trump = dtm[,names]
  TRUMP = apply(dtm_trump, 1, sum)

  #how many times was ted cruz referred to by other candidates
 names = character()
  if('ted' %in% colnames(dtm)){names = c(names, 'ted')}
  if('cruz' %in% colnames(dtm)){names = c(names, 'cruz')}
  dtm_trump = dtm[,names]
  dtm_cruz = dtm[,names]
  CRUZ = apply(dtm_cruz, 1, sum)

  #how many times was marco rubio referred to by other candidates
  names = character()
  if('marco' %in% colnames(dtm)){names = c(names, 'marco')}
  if('rubio' %in% colnames(dtm)){names = c(names, 'rubio')}
  dtm_trump = dtm[,names]
  dtm_rubio = dtm[,names]
  RUBIO = apply(dtm_rubio, 1, sum)

  #summary matrix
  data.frame(TRUMP=TRUMP, CRUZ=CRUZ, RUBIO=RUBIO)
}

dates = c(20150806, 20150916, 20151028, 20151110, 20151215,
          20160114, 20160128, 20160206, 20160213, 20160225,
          20160303)

```

```
ref_list = lapply(dates, ref_matrix)

names(ref_list) = dates

trump = sapply(ref_list, function(df) sum(df[rownames(df) != 'TRUMP', 'TRUMP']))
cruz_rubio = sapply(ref_list, function(df) sum(df[rownames(df) == 'CRUZ', 'RUBIO'], df[rownames(d

m = t(as.matrix(data.frame(trump, cruz_rubio)))
barplot(m, main='Number of times Cruz/Rubio mention Trump vs each other', beside=TRUE, col=c('blu
        legend=c('# times Cruz/Rubio mention Trump', '# times Cruz/Rubio mention each other'),
        args.legend=list(x='topleft', cex=0.75), cex.names=0.75)
```



(https://klondata.wordpress.com/2016/03/06/7/unnamed-chunk-1-1/)
We can see that at the beginning of the race, the candidates really didn't refer to each other much at all.

Things change around the debate of December 15th, 2015, where Cruz and Rubio refer to each other significantly more.
Then in the next debate and every other debate afterwards, Cruz and Rubio collectively refer to Trump more often than they refer to each other.

In particular, in the last two debates, of February 25th, 2016 and March 3rd, 2016, they mention each other 10 times in total, where as they mention Trump 137 times !

Let's now turn our attention to the words themselves.

We're gonna change the code a little bit in order to collect all the transcripts in a single character string:

```
dates = c(20150806, 20150916, 20151028, 20151110, 20151215,
          20160114, 20160128, 20160206, 20160213, 20160225,
          20160303)
read_transcript = function(date){scan(paste0('debate_', date, '.txt'), what='x', quote=NULL)}

#read and collate all transcripts
text = unlist(sapply(dates, read_transcript))
```

After reusing the same bit of code as in the beginning to preprocess the text, we can answer a few intersting questions:

For each candidate, what are the top 50 most frequent words ?

```
get indexes of 50 most frequent words
indexes = apply(dtm, 1, function(v) head(order(v, decreasing=TRUE), 50))

# find the 50 most frequent words
freq_words = apply(indexes, 2, function(v) colnames(dtm)[v])
freq_words
```

```
##         CRUZ          TRUMP       RUBIO
##    [1,] 'donald'      'people'    'people'
##    [2,] 'president'   'country'   'president'
```

```
##     [3,] 'people'      'just'      'country'
##     [4,] 'now'         'think'     'now'
##     [5,] 'tax'         'said'      'america'
##     [6,] 'obama'       'now'       'states'
##     [7,] 'country'     'tell'      'united'
##     [8,] 'said'        'right'     'just'
##     [9,] 'question'    'great'     'world'
##    [10,] 'right'       'like'      'first'
##    [11,] 'back'        'way'       'issue'
##    [12,] 'just'        'look'      'think'
##    [13,] 'america'     'back'      'like'
##    [14,] 'washington'  'take'      'american'
##    [15,] 'years'       'lot'       'way'
##    [16,] 'court'       'much'      'important'
##    [17,] 'american'    'come'      'make'
##    [18,] 'clinton'     'make'      'years'
##    [19,] 'hillary'     'thing'     'immigration'
##    [20,] 'tell'        'never'     'money'
##    [21,] 'fight'       'years'     'tax'
##    [22,] 'law'         'china'     'isis'
##    [23,] 'amnesty'     'world'     'see'
##    [24,] 'isis'        'first'     'said'
##    [25,] 'day'         'talking'   'someone'
##    [26,] 'first'       'good'      'time'
##    [27,] 'look'        'money'     'believe'
##    [28,] 'think'       'win'       'government'
##    [29,] 'like'        'everybody' 'never'
##    [30,] 'state'       'trade'     'back'
##    [31,] 'crosstalk'   'care'      'barack'
##    [32,] 'new'         'something' 'military'
##    [33,] 'barack'      'deal'      'things'
##    [34,] 'flat'        'wall'      'right'
##    [35,] 'keep'        'really'    'americans'
##    [36,] 'percent'     'problem'   'economy'
##    [37,] 'plan'        'believe'   'made'
##    [38,] 'women'       'billion'   'obama'
##    [39,] 'bill'        'saying'    'today'
##    [40,] 'radical'     'crosstalk' 'fact'
##    [41,] 'stage'       'big'       'able'
##    [42,] 'government'  'president' 'hillary'
```

```
##   [43,] 'business'    'time'      'question'
##   [44,] 'john'        'wrong'     'donald'
##   [45,] 'marco'       'done'      'clinton'
##   [46,] 'everyone'    'excuse'    'plan'
##   [47,] 'islamic'     'jobs'      'point'
##   [48,] 'millions'    'laughter'  'support'
##   [49,] 'immigration' 'far'       'better'
##   [50,] 'men'         'jeb'       'place'
```

It would actually be interesting to see for each candidate, the words in his top-50 that are unique to him, i.e. that are not in the other candidates' top-50

```
find the number of times each word appears in the matrix
word_count = apply(freq_words, c(1,2), function(x) sum(x == freq_words))

#keep those words that appear only once
unique_words = word_count == 1

l = lapply(colnames(freq_words), function(name) freq_words[unique_words[,name], name])
names(l) = colnames(freq_words)
l
```

```
## $CRUZ
##  [1] 'washington' 'court'      'fight'      'law'        'amnesty'
##  [6] 'day'        'state'      'new'        'flat'       'keep'
## [11] 'percent'    'women'      'bill'       'radical'    'stage'
## [16] 'business'   'john'       'marco'      'everyone'   'islamic'
## [21] 'millions'   'men'
##
## $TRUMP
##  [1] 'great'     'take'      'lot'       'much'      'come'
##  [6] 'thing'     'china'     'talking'   'good'      'win'
## [11] 'everybody' 'trade'     'care'      'something' 'deal'
## [16] 'wall'      'really'    'problem'   'billion'   'saying'
## [21] 'big'       'wrong'     'done'      'excuse'    'jobs'
## [26] 'laughter'  'far'       'jeb'
##
## $RUBIO
```

```
##  [1] 'states'    'united'    'issue'     'important' 'see'
##  [6] 'someone'   'military'  'things'    'americans' 'economy'
## [11] 'made'      'today'     'fact'      'able'      'point'
## [16] 'support'   'better'    'place'
```

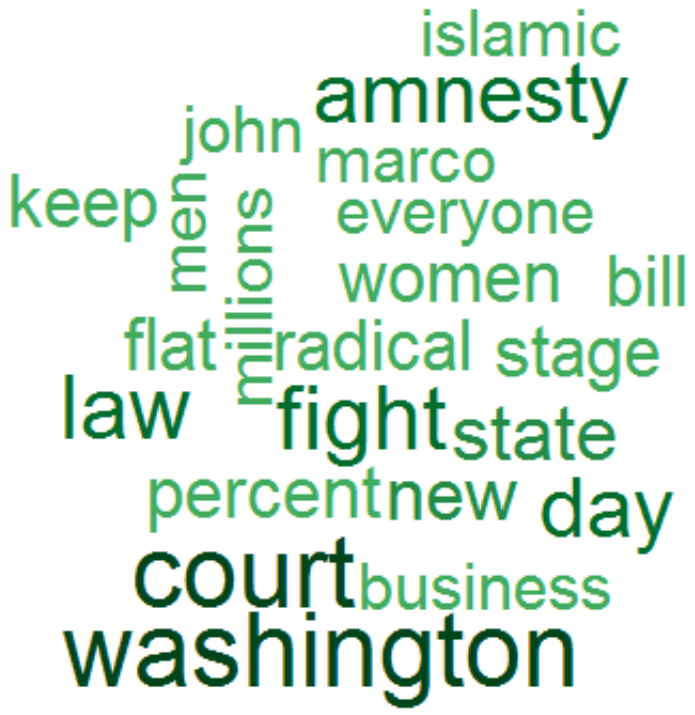
Let's go ahead and make word clouds out of that:

Trump:

```
freq = dtm['TRUMP', l$TRUMP]
wordcloud(l$TRUMP, as.vector(freq), colors=brewer.pal(9,'Blues'))
```

Cruz:

```
freq = dtm['CRUZ', l$CRUZ]
wordcloud(l$CRUZ, as.vector(freq), colors=brewer.pal(9,'Greens'))
```



Rubio:

```
freq = dtm['RUBIO', l$RUBIO]
wordcloud(l$RUBIO, as.vector(freq), colors=brewer.pal(9,'Oranges'))
```

For each candidate, what is the average word length ?

```
1 #number of words by speaker
2 nb_words = apply(dtm, 1, sum)
```

```
#word lengths
word_lengths = sapply(colnames(dtm), nchar)
#transform word count into total character count matrix
character_counts = t(apply(dtm, 1, function(v) v * word_lengths))
#total character count by speaker
total_character_counts = apply(character_counts, 1, function(v) sum(v))
#divide total character count by numer of words
round(total_character_counts / nb_words, digits=1)
```

```
##  CRUZ TRUMP RUBIO
##   6.3   5.9   6.3
```

Finally, for each candidate, how diversified is their vocabulary ?

To quantify this, we are gonna count the number of unique words per 1000 words:

```
apply(dtm, 1, function(v) round(sum(v != 0)/sum(v)*1000, digits=1))
```

```
##  CRUZ TRUMP RUBIO
## 248.8 176.3 218.0
```

We see that Trump uses fewer and shorter words than his opponents.

Posted in R and tagged R on March 6, 2016 by Karim L. Leave a comment