Textual Analysis of 2016 Philippines' Presidential Debate

Benjamin Tabares Jr.

April 17, 2016

The recent Philippines' Presidential Debate (Pilipinas Debates 2016) drew unprecedented amounts of viewers and the usual lot of controversies and soundbites in the media. In our polarized media landscape, ensuing political analysis always suffer from political bias. Whether you trust the media, you will get a very different take on what was said.

So far, the candidates have spoken thousands of words in the two (2) debates held by two TV networks. In this simple textual analysis, I turn all of the words into data and make inferences about the candidates' characteristics that would otherwise be too difficult to discover.

NLP tools and methods can help bring some objectivity to better understand the current political discourse. Using different state-of-the-art NLP libraries and packages, we will try to answer questions about: Debate Dynamics - What can be inferred about the debaters performances? (FINISHED) Sentiment Analysis - How do the candidates feel about certain issues? (WIP) Topic Modelling - What did the candidates really talked about? (WIP) - What was the most important subject for each candidate?

#set working directory  
getwd()

## [1] "Z:/GitHub/Data Projects/Pilipinas\_Debates\_2016"

setwd("Z:/Github/Data Projects/Pilipinas\_Debates\_2016/transcripts")  
#load required lib  
library(tm)  
library(SnowballC)  
library(wordcloud)  
library(RColorBrewer)

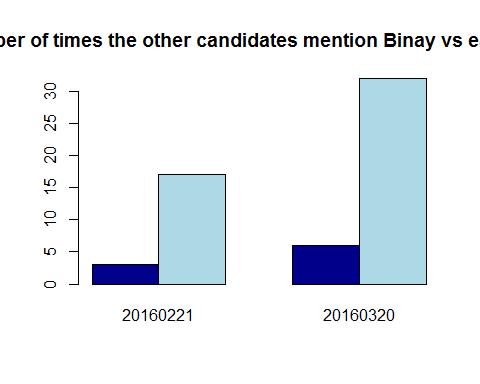
First let's answer the following question: how often does a certain candidate get mentioned by name by the other candidates VS how often do the other candidates mention each other's name. On the first debate, the candidates really didn't refer to each other much at all.

ref\_matrix = function (date){  
   
 #read character data  
 text = scan(paste0('debate\_', date, '\_full', '.txt'), what='x', quote=NULL)  
   
 speakers =  
 c( BINAY = '',  
 MAR = '',  
 DUTERTE = '',  
 MIRIAM = '',  
 POE = ''  
 )  
   
 #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('laughter', 'applause', 'intermission', 'and', 'the', 'all',  
 'ang','mga','you','hindi','that','kung','ating','said','naman',  
 'namin','natin','yung','pero','sila','nila','ako','yan','para',  
 'will','isang','ito','doon','alam','may','kanilang','dahil',  
 'lahat','have','tayo','kami','for','because','rin','ngayon',  
 'kaya','kaya','hong','walang','but','kahit','dapat','lang',  
 'mas','pwedeng','saan','yong','buong','yong','ayaw','lang',  
 'not','wala','there','are','was','would','ano','well','can',  
 'kasi','dito','our','akong','with','dun','gusto','ninyo',  
 'bakit','niya','like','ibang','talagang','did','yes','yun','this',  
 'anong','man','diyan','tulad','pwede','talaga','give','bilang','isa',  
 'itong','what','nasa','sinasabi','just','kayo','lamang','nga','luchi',  
 'mismo','other','akin','kanya','ikaw','cannot','aking','ginawa',  
 'kanila','ginawa','know','really','sapagkat','din','maraming','they',  
 'meron','mong','uulitin','yon',  
 'yon','yo','tony','below','anyway','eto','aba','when','madam','amin',  
 'nandun','pagka','mahigit','senadora','sya','inyong','parang','salamat',  
 'senador','nagiging','yeah','look','how','goes','huwag','see','say',  
 'kang','actually','sinabi','about','mayor','already','nung','importante',  
 'oras','tama','here','very','into','pati','why','sabi','those','always',  
 'has','first','ganoon','atin','dalawang','kailangan','papaano','sabihin',  
 'nilang','hanggang','nating','from','siguro','tayong','naging','after',  
 'siya','namang','nagsabi','nang','lagi','niyo','grace','nagsasabi','nyo',  
 'pong','basta','dyan','iyan','pang','tayong','which','should','sino','these',  
 'without','di','yang','ilang','noon','noong','pag','senator','iyon','kong',  
 'mag','maging','nandiyan','nang','per','pumunta','iyan','let','muna','each',  
 'aming','sana','also','basta','ganito'))  
 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 = DocumentTermMatrix(prez)   
 #reassign row names (each row is a speaker)  
 rownames(dtm) = names(speakers)  
   
 #how many times was VP Jejomar Binay referred to by other candidates  
 binay\_names = character()  
 if('binay' %in% colnames(dtm)){binay\_names = c(binay\_names, 'binay')}  
 if('jejomar' %in% colnames(dtm)){binay\_names = c(binay\_names, 'jejomar')}  
 if('vp' %in% colnames(dtm)){binay\_names = c(binay\_names, 'vp')}  
 if('vice president' %in% colnames(dtm)){binay\_names = c(binay\_names, 'vice president')}  
 dtm\_binay = dtm[,binay\_names]  
 BINAY = apply(dtm\_binay, 1, sum)  
   
 #how many times was Mar Roxas referred to by other candidates  
 mar\_names = character()  
 if('mar' %in% colnames(dtm)){mar\_names = c(mar\_names, 'mar')}  
 if('roxas' %in% colnames(dtm)){mar\_names = c(mar\_names, 'roxas')}  
 if('secretary' %in% colnames(dtm)){mar\_names = c(mar\_names, 'secretary')}  
 dtm\_mar = dtm[,mar\_names]  
 MAR = apply(dtm\_mar, 1, sum)  
   
 #how many times was Rodrigo Duterte referred to by other candidates  
 duterte\_names = character()  
 if('mayor' %in% colnames(dtm)){duterte\_names = c(duterte\_names, 'mayor')}  
 if('duterte' %in% colnames(dtm)){duterte\_names = c(duterte\_names, 'duterte')}  
 if('rudy' %in% colnames(dtm)){duterte\_names = c(duterte\_names, 'rudy')}  
 if('rodrigo' %in% colnames(dtm)){duterte\_names = c(duterte\_names, 'rodrigo')}  
 dtm\_duterte = dtm[,duterte\_names]  
 DUTERTE = apply(dtm\_duterte, 1, sum)  
   
 #how many times was Sen. Miriam Defensor-Santiago referred to by other candidates  
 miriam\_names = character()  
 if('santiago' %in% colnames(dtm)){miriam\_names = c(miriam\_names, 'santiago')}  
 if('miriam' %in% colnames(dtm)){miriam\_names = c(miriam\_names, 'miriam')}  
 dtm\_miriam = dtm[,miriam\_names]  
 MIRIAM = apply(dtm\_miriam, 1, sum)  
   
 #how many times was Sen. Grace Poe referred to by other candidates  
 poe\_names = character()  
 if('grace' %in% colnames(dtm)){poe\_names = c(poe\_names, 'grace')}  
 if('poe' %in% colnames(dtm)){poe\_names = c(poe\_names, 'poe')}  
 dtm\_poe = dtm[,poe\_names]  
 POE = apply(dtm\_poe, 1, sum)  
   
 #summary matrix  
 data.frame(BINAY=BINAY, MAR=MAR, DUTERTE=DUTERTE, MIRIAM=MIRIAM, POE=POE)  
}

dates = c(20160221,20160320)  
ref\_list = lapply(dates, ref\_matrix)  
names(ref\_list) = dates

On the first debate, the candidates really did not refer to each other much at all. Things change around the second debate, where the candidates refer to each other significantly more. Three (3) candidates stood out that got mentioned by name by the other candidates. Out of these trio, the other candidates collectively referred to Mar more often than they refer to each other. In particular, they throw jabs to each other 13 times in total (light blue bar), where as they ganged-up on Mar 45 times! (dark blue bar)

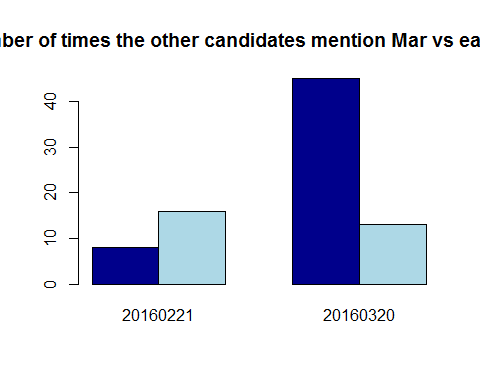
###########################  
# First, let's answer the question: How does Binay get mentioned by name by the other four candidates during a debate,  
# versus how often do the other candidates mention each other's name?  
###########################  
binay = sapply(ref\_list, function(df) sum(df[rownames(df) != 'BINAY', 'BINAY']))  
the\_rest = sapply(ref\_list, function(df) sum(df[rownames(df) == 'MAR', 'DUTERTE'], df[rownames(df) == 'DUTERTE', 'MAR'],  
 df[rownames(df) == 'MAR', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'MAR'],  
 df[rownames(df) == 'MAR', 'POE'],df[rownames(df) == 'POE', 'MAR'],  
 df[rownames(df) == 'DUTERTE', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'DUTERTE'],  
 df[rownames(df) == 'DUTERTE', 'POE'],df[rownames(df) == 'POE', 'DUTERTE'],  
 df[rownames(df) == 'MIRIAM', 'POE'],df[rownames(df) == 'POE', 'MIRIAM']))  
m = t(as.matrix(data.frame(binay, the\_rest)))  
# MAR  
mar = sapply(ref\_list, function(df) sum(df[rownames(df) != 'MAR', 'MAR']))  
the\_rest = sapply(ref\_list, function(df) sum(df[rownames(df) == 'BINAY', 'DUTERTE'], df[rownames(df) == 'DUTERTE', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'POE'],df[rownames(df) == 'POE', 'BINAY'],  
 df[rownames(df) == 'DUTERTE', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'DUTERTE'],  
 df[rownames(df) == 'DUTERTE', 'POE'],df[rownames(df) == 'POE', 'DUTERTE'],  
 df[rownames(df) == 'MIRIAM', 'POE'],df[rownames(df) == 'POE', 'MIRIAM']))  
n = t(as.matrix(data.frame(mar, the\_rest)))  
# DUTERTE  
duterte = sapply(ref\_list, function(df) sum(df[rownames(df) != 'DUTERTE', 'DUTERTE']))  
the\_rest = sapply(ref\_list, function(df) sum(df[rownames(df) == 'BINAY', 'MAR'], df[rownames(df) == 'MAR', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'POE'],df[rownames(df) == 'POE', 'BINAY'],  
 df[rownames(df) == 'MAR', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'MAR'],  
 df[rownames(df) == 'MAR', 'POE'],df[rownames(df) == 'POE', 'MAR'],  
 df[rownames(df) == 'MIRIAM', 'POE'],df[rownames(df) == 'POE', 'MIRIAM']))  
o = t(as.matrix(data.frame(duterte, the\_rest)))  
# POE  
poe = sapply(ref\_list, function(df) sum(df[rownames(df) != 'POE', 'POE']))  
the\_rest = sapply(ref\_list, function(df) sum(df[rownames(df) == 'BINAY', 'MAR'], df[rownames(df) == 'MAR', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'BINAY'],  
 df[rownames(df) == 'MAR', 'MIRIAM'],df[rownames(df) == 'MIRIAM', 'MAR'],  
 df[rownames(df) == 'MAR', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'MAR'],  
 df[rownames(df) == 'MIRIAM', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'MIRIAM']))  
p = t(as.matrix(data.frame(poe, the\_rest)))  
# MIRIAM  
miriam = sapply(ref\_list, function(df) sum(df[rownames(df) != 'MIRIAM', 'MIRIAM']))  
the\_rest = sapply(ref\_list, function(df) sum(df[rownames(df) == 'BINAY', 'MAR'], df[rownames(df) == 'MAR', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'POE'],df[rownames(df) == 'POE', 'BINAY'],  
 df[rownames(df) == 'BINAY', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'BINAY'],  
 df[rownames(df) == 'MAR', 'POE'],df[rownames(df) == 'POE', 'MAR'],  
 df[rownames(df) == 'MAR', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'MAR'],  
 df[rownames(df) == 'POE', 'DUTERTE'],df[rownames(df) == 'DUTERTE', 'POE']))  
q = t(as.matrix(data.frame(miriam, the\_rest)))  
  
  
barplot(m,main='NUmber of times the other candidates mention Binay vs each other',  
 beside=TRUE,col=c('darkblue','lightblue'),  
 #legend=c('# of times the other candidates mention Binay','# of times the other candidates mention each other'),  
 legend.text = FALSE)



m

## 20160221 20160320  
## binay 3 6  
## the\_rest 17 32

barplot(n,main='NUmber of times the other candidates mention Mar vs each other',  
 beside=TRUE,col=c('darkblue','lightblue'),  
 #legend=c('# of times the other candidates mention Mar','# of times the other candidates mention each other'),  
 legend.text = FALSE)



n

## 20160221 20160320  
## mar 8 45  
## the\_rest 16 13

## Who mentioned Mar's name the most?  
mar\_binay = sapply(ref\_list, function(df) sum(df[rownames(df) == 'BINAY', 'MAR']))  
mar\_binay

## 20160221 20160320   
## 5 21

mar\_duterte = sapply(ref\_list, function(df) sum(df[rownames(df) == 'DUTERTE', 'MAR']))  
mar\_duterte

## 20160221 20160320   
## 0 14

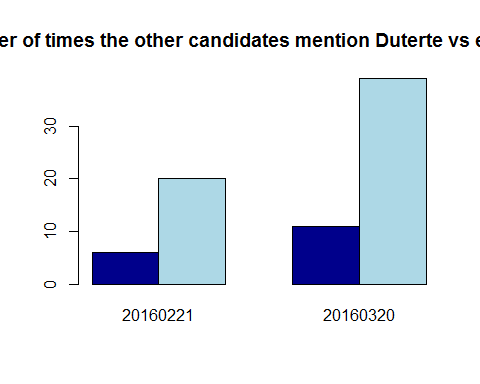
mar\_poe = sapply(ref\_list, function(df) sum(df[rownames(df) == 'POE', 'MAR']))  
mar\_poe

## 20160221 20160320   
## 2 10

mar\_miriam = sapply(ref\_list, function(df) sum(df[rownames(df) == 'MIRIAM', 'MAR']))  
mar\_miriam

## 20160221 20160320   
## 1 0

barplot(o,main='NUmber of times the other candidates mention Duterte vs each other',  
 beside=TRUE,col=c('darkblue','lightblue'),  
 #legend=c('# of times the other candidates mention Duterte','# of times the other candidates mention each other'),  
 legend.text = FALSE)



o

## 20160221 20160320  
## duterte 6 11  
## the\_rest 20 39

p

## 20160221 20160320  
## poe 2 1  
## the\_rest 18 45

q

## 20160221 20160320  
## miriam 10 1  
## the\_rest 16 63

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

dates = c(20160221,20160320)  
#collect all the transcripts in a single character string  
read\_transcript = function(date){scan(paste0('debate\_', date, '\_full', '.txt'), what='x', quote=NULL)}  
#read and collate all transcripts  
text = unlist(sapply(dates, read\_transcript))  
  
  
speakers =  
 c( BINAY = '',  
 MAR = '',  
 DUTERTE = '',  
 MIRIAM = '',  
 POE = ''  
 )  
  
#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)  
 }  
}  
  
  
##############################################################################  
# The candidates' vocabularies are further simplified by removing common words  
# that convey little meaning (e.g.,prepositions,pronouns) and stemming the  
# remaining, more substantive words.  
##############################################################################  
  
#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('laughter', 'applause', 'intermission', 'and', 'the', 'all',  
 'ang','mga','you','hindi','that','kung','ating','said','naman',  
 'namin','natin','yung','pero','sila','nila','ako','yan','para',  
 'will','isang','ito','doon','alam','may','kanilang','dahil',  
 'lahat','have','tayo','kami','for','because','rin','ngayon',  
 'kaya','kaya','hong','walang','but','kahit','dapat','lang',  
 'mas','pwedeng','saan','yong','buong','yong','ayaw','lang',  
 'not','wala','there','are','was','would','ano','well','can',  
 'kasi','dito','our','akong','with','dun','gusto','ninyo',  
 'bakit','niya','like','ibang','talagang','did','yes','yun','this',  
 'anong','man','diyan','tulad','pwede','talaga','give','bilang','isa',  
 'itong','what','nasa','sinasabi','just','kayo','lamang','nga','luchi',  
 'mismo','other','akin','kanya','ikaw','cannot','aking','ginawa',  
 'kanila','ginawa','know','really','sapagkat','din','maraming','they',  
 'meron','roxas','duterte','poe','mar','miriam',  
 'mong','uulitin','yon','yo','tony','below','anyway','eto','aba',  
 'when','madam','amin','nandun','pagka','mahigit','senadora','sya',  
 'inyong','parang','salamat','nagiging','yeah','look','how','goes','huwag','see','say',  
 'kang','actually','sinabi','about','already','nung','importante',  
 'oras','tama','here','very','into','pati','why','sabi','those','always',  
 'has','first','ganoon','atin','dalawang','kailangan','papaano','sabihin',  
 'nilang','hanggang','nating','from','siguro','tayong','naging','after',  
 'siya','namang','nagsabi','nang','lagi','niyo','grace','nagsasabi','nyo',  
 'pong','basta','dyan','iyan','pang','tayong','which','should','sino','these',  
 'without','di','yang','ilang','noon','noong','pag','iyon','kong',  
 'mag','maging','nandiyan','nang','per','pumunta','iyan','let','muna','each',  
 'aming','sana','also','basta','ganito'))  
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 = DocumentTermMatrix(prez)   
#reassign row names (each row is a speaker)  
rownames(dtm) = names(speakers)