HW2_Curran(v2)

Thomas Curran

4/26/2017

Debating with Quanteda: 1) Install the quanteda package: 2) Create DEBATES Object

3) Explore to what extent the Heap's Law applies for trump vs Clinton. Is it stronger or weaker for either candidate?

	Estimate	Std. Error	t value	Pr(>|t|)
log(trump_sum$Tokens)	0.8602	0.012	71.66	1.808e-86
(Intercept)	0.1655	0.03684	4.492	1.93e-05

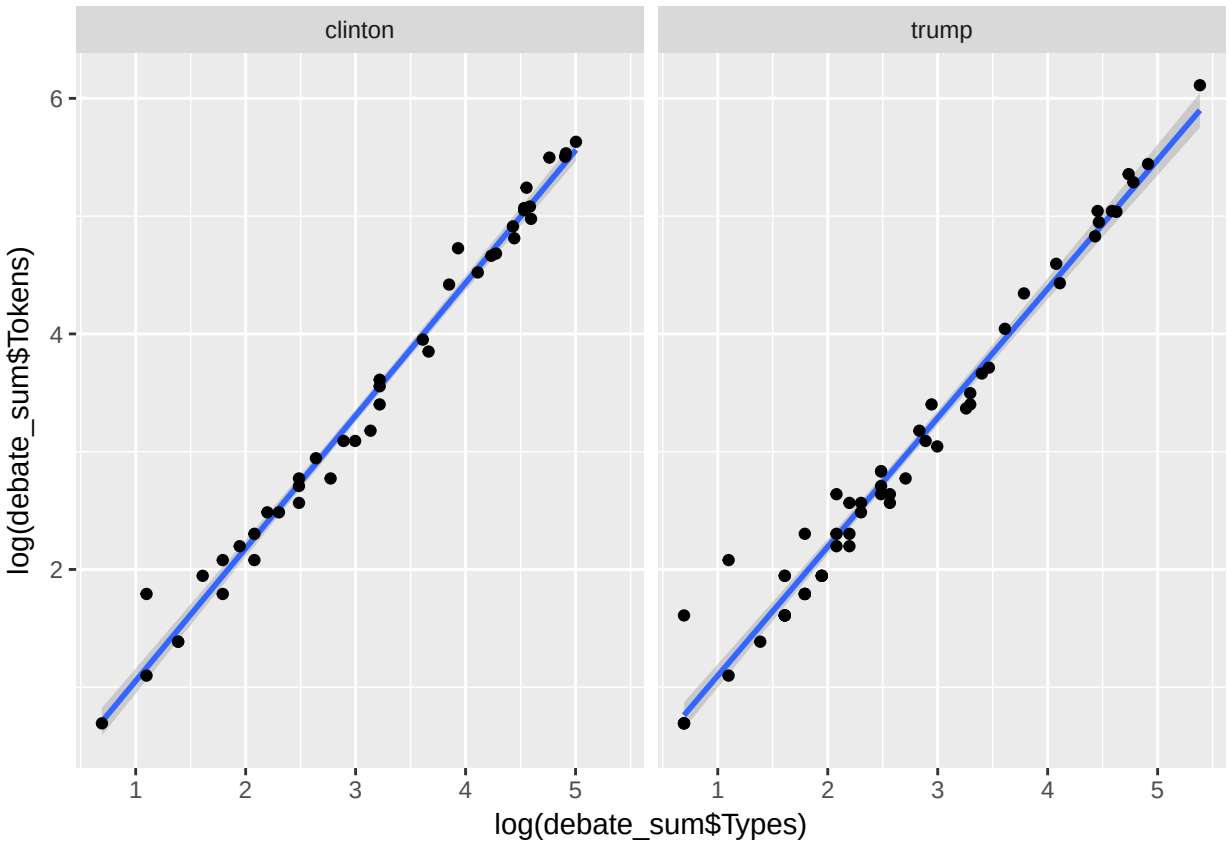
Table 2: Fitting linear model: log(trump_sum$Types$) $log(trump_sum$Tokens)

Observations	Residual Std. Error	R^2	Adjusted R^2
100	0.1509	0.9813	0.9811

	Estimate	Std. Error	t value	Pr(>|t|)
log(clinton_sum$Tokens)	0.9135	0.01375	66.42	2.656e-83
(Intercept)	-0.05524	0.05401	-1.023	0.3089

Table 4: Fitting linear model: log(clinton_sum$Types$) $log(clinton_sum$Tokens)

Observations	Residual Std. Error	R^2	Adjusted R^2
100	0.1856	0.9783	0.978



4)Analyze the lexical diversity across the candidates before and after they became their parties' respective candidates:

Note: the Readability Score for Trump before primary kept throwing errors even after using code suggestions posted in Piazza and in class

Readability Scores for Candidates Before and After Primary

candidates	scores
Clinton Before Nomination	6.551
Clinton After Nomination	5.757
Trump Before Nomination	3.452
Trump After Nomination	1

Note: The lexical diversity for Trump after the nomination was throwin an 'Inf' result, not sure what is causing that result.

Lexical Diversity Scores for Candidates Before and After Primaries

candidates_lexdiv	scores_lexdiv
Clinton Before Nomination	148.7
Clinton After Nomination	292.3
Trump Before Nomination	189.6
Trump After Nomination	Inf

5) Do you have a hypothesis why patterns may be more or less pronounced between trump versus clinton? How could you test this?

The lexical diversity that exists before and after the primaries could be explained by the fact that during the primary debates there were more individuals debating than in the presidential debates. having more people debating meant less time speaking, resulting in shorter responses/fragments for each candidate. Given Heap's Law the shorter the fragment the lower the lexical diversity. Trump's lexical diversity may have the most signficant change simply because there were more people at the Republican debates as opposed to the Democratic debates which were only three people. Once it was the presidential debates, both candidates had more time to speak since they shared the time and stage with one other person, which resulted in longer fragments, and which according to Heap's Law lead to more lexical diversity.

6) Remove Stopwords from Corpus

```
##############################################################################
#6) Remove stopwords from corpus
##############################################################################
DEBATES_nostop <- dfm(DEBATES, remove=stopwords("english"))

Debates_clinton_nostop <- dfm(clinton, remove = stopwords("english"))

Debates_trump_nostop <- dfm(trump, remove = stopwords("english"))
```

7) Using the tokenize function, construct seperate bi-grams for the Hilary Clinton/Donald Trump parts of the corpus. Tabulte the ten most frequent bi-grams by speaker. Are these informative? Why or why not?

These bi-grams are informative as the are insights into themese in the debates as well as between candidates. The frequency of the bigrams shows the Clinton frequently spoke about former President Obama and Senator Bernie Sanders while Trump's bi-grams are less informative but still telling of the candidate's message. Trump's bi-grams seem to include more common language than Clinton's, this can also be explained by the readability score calculated for each candidate in the previous question. The Debates bi-gram is usefull because it reflects the campaign issues in the past election include health care and wall street. However, these n-grams do not provide context in terms of attitude or the message about each topic, it only reflects the frequency in which they appear in the corpus

Debate bi-gram

token	N
United States	83
Senator Sanders	83
President Obama	67
right now	66
health care	50
New York	46
make sure	46
Wall Street	42
Well first	39
will tell	37

Trump Bi-grams

token	N
right now	48
United States	37
will tell	33
take care	25

token	N
many people	24
just tell	23
Hillary Clinton	20
let just	19
many many	19
Middle East	19

Clinton bi-gram

token	N
Senator Sanders	83
President Obama	54
United States	46
make sure	44
health care	43
Wall Street	38
Affordable Care	33
Care Act	32
New York	29
think important	27

8) Using the collocation function in quanteda (which takes a tokenize object) construct collocations based on Chi2 test for each speaker. Order by Chi2 test statistic. What do you notice or what is strange? Can you provide a formal reasoning relating to the Chi2 test statistic formula?

In the Chi2 test for candidates, it appears that the count of the top two or three words is much greater than the next top bi-grams while the X2 statistic remains the same. The formal reasoning relating to the chi2 test statistic formula is that it describes the comparison between observered frequencies in the table with the frequences expected for independence. The large chi2 test tells us we can reject the t-test. Essenitally, the chi2 tells us that "Afforable" and "Care", accounting for capitlization, are extremely likely to appear togethor as opposed to apart. In other words, the expected frequency of "Affordable" is marginal probabily of "Care" occuring as the first of a bigram times the marginal probablity of "care occuring as the second."

Clinton Collocation

collocation	count	X2
Affordable Care	33	75676
Supreme Court	23	75676
Planned Parenthood	7	75676
electric grid	2	75676
Alicia Machado	2	75676
Situation Room	2	75676
Celebrity Apprentice	2	75676
Stronger Together	2	75676
implicit bias	2	75676
Los Angeles	2	75676

Trump Collocation

collocation	count	X2
Middle East	19	49978
Ronald Reagan	9	49978
gold standard	6	49978
Supreme Court	5	49978
Palm Beach	5	49978
college tuition	2	49978
Trojan horse	2	49978
block grant	2	49978
extreme vetting	2	49978
Douglas MacArthur	2	49978

9) Using the code provided in the lectures, identify collocations that are distinct to Trump vs Clinton. Based on your perception of each candidate, do the results of this analysis make sense? Provide a brief answer where
you summarize the most important results and your explanations.

In this chi2 test, we are investigating the dissimilarities between the candidate's words. The null hypothesis in this case is that the probablity of observing a word or pair of words is independent across speakers, in other words that a phrase that appears in the corpus of one candidate, doesn't appear in the corpus of another. The table below essenitally joins two data frames of bigrams using the bi-gram as the key and adding columns for the number of times it appears by each speaker. The first several words appear to be either heavily used by one cadidate or ther other, for example the first phrase "many people" has a chi2 score of 4.084, Furthermore, the chi2 dissimilarities test shows that the two candidates have very different speaking patters has made evident by the frequency count for matched bi-grams and the corresponding chi2 generated. In class, comparing the Obama and Bush state of the unions has tokens such as "Social Secruity" having a chi2 score of 29.76, but in the case of Clinton versus Trump the highest chi2 was approximately 4, reinforcing that the speakers had very different speaking styles and different characteristics to their speaking patterns and topics.

```
tok <- quanteda::removeFeatures(quanteda::tokenize(DEBATES, remove_punct=TRUE), stopwords("english"))
tok <-unlist(tokens_ngrams(tok,n=2, concatenator=" "))
tok<- data.table("text"=names(tok),"token"=tok)
tok[,speaker:=DEBATES[["speaker"]]]
```

```
## Warning in `[.data.table`(tok, , `:=`(speaker, DEBATES[["speaker"]])):
## Supplied 1955 items to be assigned to 48438 items of column
## 'speaker' (recycled leaving remainder of 1518 items).
```

```
tok<-tok[,.N,by=c("speaker","token")]
pander(tok[order(N, decreasing=TRUE)][1:20])
```

speaker	token	N
trump	United States	50
trump	Senator Sanders	49
trump	President Obama	39
trump	right now	37
clinton	Senator Sanders	34
clinton	United States	33
clinton	right now	29
trump	New York	29
clinton	President Obama	28
clinton	health care	26
trump	Well first	25

speaker	token	N
trump	will tell	25
trump	Wall Street	25
trump	health care	24
trump	Well think	24
clinton	make sure	24
trump	make sure	22
clinton	many people	21
trump	Social Security	21
trump	think important	19

```
clinton_join<- tok[speaker=="clinton"][,list(token, clintoncount=as.numeric(N))]
trump_join <-tok[speaker=="trump"][,list(token, trumpcount=as.numeric(N))]

wide<-merge(clinton_join, trump_join)
wide[is.na(clintoncount)]$clintoncount <-0
wide[is.na(trumpcount)]$trumpcount<-0
wide[,':='(totalcount, clintoncount+trumpcount)]
wide<-wide[order(totalcount, decreasing=TRUE)][totalcount>0]
wide[,':='(totalclinton, sum(clintoncount))]
wide[,':='(totaltrump, sum(trumpcount))]
wide[, `:=`(chi2, (totalclinton + totaltrump) * (trumpcount * (totalclinton - clintoncount) -
clintoncount * (totaltrump - trumpcount))^2/((trumpcount + clintoncount) * (trumpcount +
(totaltrump - trumpcount)) * (clintoncount + (totalclinton - clintoncount)) * ((totaltrump -
trumpcount) + (totalclinton - clintoncount))))]

pander(wide[1:20][order(chi2, decreasing=TRUE)])
```

Table 13: Table continues below

token	clintoncount	trumpcount	totalcount	totalclinton
many people	21	12	33	5227
will tell	12	25	37	5227
Well think	12	24	36	5227
Well first	14	25	39	5227
New York	17	29	46	5227
United States	33	50	83	5227
Senator Sanders	34	49	83	5227
health care	26	24	50	5227
Middle East	12	19	31	5227
make sure	24	22	46	5227
Wall Street	17	25	42	5227
going get	11	17	28	5227
President Obama	28	39	67	5227
Care Act	13	19	32	5227
Affordable Care	14	19	33	5227
let just	15	15	30	5227
right now	29	37	66	5227
Social Security	16	21	37	5227
Supreme Court	13	15	28	5227
secretary state	13	15	28	5227

totaltrump	chi2
6105	4.084
6105	2.801
6105	2.378
6105	1.648
6105	1.563
6105	1.364
6105	0.8966
6105	0.6973
6105	0.688
6105	0.6798
6105	0.5415
6105	0.5285
6105	0.5097
6105	0.3908
6105	0.1825
6105	0.1817
6105	0.1277
6105	0.1241
6105	0.001034
6105	0.001034