Convention Text Analysis: Tokenization

Jennifer Lin

11/25/2019

Contents

Introduction

The process of tokenization boils text down to the simplest forms so that only the root words are counted and noise is diminished. Here, I eliminate the filler, stopwords, and prefixes/suffixes of words in DNC and RNC convention speeches to see if there are differences in outcomes based on methods used to process the text

```
# Load packages
library(quanteda)
library(readtext)
library(psych)
library(effsize)
```

DNC Speeches

Below, I show the process I used to tokenize the DNC corpus of text

```
# Load in data
DNC <- readtext("~/Desktop/Data/DNCtexts/*.txt", docvarsfrom = "filenames")

# Create the corpus
DNC_data <- corpus(DNC)
```

```
########## Cleaning and Creating tokens #####################

DNC.tokens <- tokens(DNC_data)

# Remove punctuations and numbers
DNC.tokens <- tokens(DNC.tokens, remove_punct = TRUE, remove_numbers = TRUE)

# Remove stopwords (i.e. if, but, and)
DNC.tokens <- tokens_select(DNC.tokens, stopwords("english"),
    selection = "remove")

# Take endings away (ie. -ed, =ing)
DNC.tokens <- tokens_wordstem(DNC.tokens)

# Convert all words to lowercase
DNC.tokens <- tokens_tolower(DNC.tokens)

# Convert to dfm
DNC.dfm <- dfm(DNC.tokens)
```

To analyze the text, I load the dictionary below.

```
load(file = "~/Desktop/Working/Moral-Psychology/SpeechAnalysis/quanteda/DNC/data_diction
dict_lg <- dictionary(data_dictionary_MFD)
```

This dictionary is applied to the analysis of the text.

```
DNCdata <- dfm_lookup(DNC.dfm, dictionary = dict_lg)
head(DNCdata)
```

```
## Document-feature matrix of: 6 documents, 10 features (45.0% sparse).
## 6 x 10 sparse Matrix of class "dfm"
##                     features
## docs                 care.virtue care.vice fairness.virtue fairness.vice
##   dnc.akbari.txt              4         6               1             0
##   dnc.albright.txt            7         7               1             0
##   dnc.alexander.txt           5         2               0             0
##   dnc.allenjohn.txt           4         1               1             0
##   dnc.amaru.txt               5         1               1             0
##   dnc.asian.txt               9         6               1             0
##                     features
## docs                 loyalty.virtue loyalty.vice authority.virtue
##   dnc.akbari.txt                 3            0                7
##   dnc.albright.txt               5            0                3
##   dnc.alexander.txt              1            0                2
##   dnc.allenjohn.txt              9            0                5
##   dnc.amaru.txt                  2            0                3
##   dnc.asian.txt                  8            0                3
##                    features
## docs                authority.vice sanctity.virtue sanctity.vice
##   dnc.akbari.txt                 0               1             0
##   dnc.albright.txt               0               0             0
##   dnc.alexander.txt              0               0             0
##   dnc.allenjohn.txt              0               1             0
##   dnc.amaru.txt                  0               2             2
##   dnc.asian.txt                  0               0             0
```

RNC Speeches

The process can be repeated with the RNC speeches, as I demonstrate below

```
# Load in data
RNC <- readtext("~/Desktop/Data/RNCtexts/*.txt", docvarsfrom = "filenames")

# Create the corpus
RNC_data <- corpus(RNC)

########## Cleaning and Creating tokens #####################

RNC.tokens <- tokens(RNC_data)

# Remove punctuations and numbers
RNC.tokens <- tokens(RNC.tokens, remove_punct = TRUE, remove_numbers = TRUE)

# Remove stopwords (i.e. if, but, and)
RNC.tokens <- tokens_select(RNC.tokens, stopwords("english"),
    selection = "remove")

# Take endings away (ie. -ed, =ing)
RNC.tokens <- tokens_wordstem(RNC.tokens)

# Convert all words to lowercase
RNC.tokens <- tokens_tolower(RNC.tokens)

# Convert to dfm
RNC.dfm <- dfm(RNC.tokens)

############ Get Dictionary ###############

load(file = "~/Desktop/Working/Moral-Psychology/SpeechAnalysis/quanteda/RNC/data_diction
dict_lg <- dictionary(data_dictionary_MFD)

############# Analyze using MFD ############

RNCdata <- dfm_lookup(RNC.dfm, dictionary = dict_lg)
head(RNCdata)
```

```
## Document-feature matrix of: 6 documents, 10 features (40.0% sparse).
## 6 x 10 sparse Matrix of class "dfm"
##                      features
## docs                  care.virtue care.vice fairness.virtue fairness.vice
##   rnc.alvarado.txt              6         1               2             1
##   rnc.baio.txt                  3         1               1             0
##   rnc.barrack.txt              10         3               0             0
##   rnc.beardsley.txt             1         3               0             0
##   rnc.blackburn.txt             4         1               1             0
##   rnc.bondi.txt                 5         2              11             0
##                      features
## docs                  loyalty.virtue loyalty.vice authority.virtue
##   rnc.alvarado.txt                 2            0               10
##   rnc.baio.txt                     0            0                3
##   rnc.barrack.txt                  5            0                6
##   rnc.beardsley.txt                3            0                8
##   rnc.blackburn.txt                4            0               12
##   rnc.bondi.txt                    1            0                6
##                      features
## docs                  authority.vice sanctity.virtue sanctity.vice
##   rnc.alvarado.txt                 0              10             1
##   rnc.baio.txt                     0               3             0
##   rnc.barrack.txt                  0               2             0
##   rnc.beardsley.txt                0               1             0
##   rnc.blackburn.txt                0               1             0
##   rnc.bondi.txt                    1               1             0
```

Analysis of Composite Data

Like the word count process, I created a composite data set with the two datasets using the `rbind` command in R.

```
DNCToken <- read.csv("~/Desktop/Working/Moral-Psychology/SpeechAnalysis/
  quanteda/DNC/DNCToken.csv", header = TRUE)
RNCToken <- read.csv("~/Desktop/Working/Moral-Psychology/SpeechAnalysis/
  quanteda/DNC/RNCToken.csv", header = TRUE)
```

```
getwd()
setwd("/Users/JenniferLin/Desktop/Working/Moral-Psychology/SpeechAnalysis/
  quanteda/Composite")

#Merge the data frames
compositeT <- rbind(DNCToken, RNCToken)

export(compositeT, "compTok.csv")
```

In the analysis below, I will load this dataset and run t-tests with the data.

```
# Load data
speech <- read.csv("~/Desktop/Working/Moral-Psychology/SpeechAnalysis/quanteda/Composite
    header = TRUE)
```

```
############### Combine Virtue and Vice #################

speech$Harm <- rowSums(speech[, c("care.virtue", "care.vice")],
    na.rm = TRUE)
speech$Fairness <- rowSums(speech[, c("fairness.virtue", "fairness.vice")],
    na.rm = TRUE)
speech$Ingroup <- rowSums(speech[, c("loyalty.virtue", "loyalty.vice")],
    na.rm = TRUE)
speech$Authority <- rowSums(speech[, c("authority.virtue", "authority.vice")],
    na.rm = TRUE)
speech$Purity <- rowSums(speech[, c("sanctity.virtue", "sanctity.vice")],
    na.rm = TRUE)
```

Then, I run descriptive statistics and t-tests for each of the foundations sorted by convention.

```
### Harm ###
describeBy(speech$Harm, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean    sd median trimmed  mad min max range skew kurtosis
## X1    1 152 8.96 10.35    5.5    6.81 5.19   0  68    68 2.95    10.49
##      se
## X1 0.84
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 10.7 13.7      7    8.23 5.93   0  93    93 3.87     18.8 1.71
```

```
t.test(speech$Harm ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Harm by speech$Convention
## t = -0.91348, df = 94.648, p-value = 0.3633
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.529959  2.044761
## sample estimates:
## mean in group DNC mean in group RNC
##          8.960526         10.703125
```

```
cohen.d(speech$Harm, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -0.1522974 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.4463674  0.1417726
```

```
### Fairness ###
describeBy(speech$Fairness, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 152 1.97 2.97      1    1.31 1.48   0  18    18 2.79      9.2 0.24
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 2.84 4.68      1    1.87 1.48   0  31    31 3.76    18.19 0.59
```

```
t.test(speech$Fairness ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Fairness by speech$Convention
## t = -1.3846, df = 85.095, p-value = 0.1698
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.1354507  0.3821612
## sample estimates:
## mean in group DNC mean in group RNC
##          1.967105          2.843750
```

```r
cohen.d(speech$Fairness, speech$Convention)
```

```
## 
## Cohen's d
## 
## d estimate: -0.2461976 (small)
## 95 percent confidence interval:
##       lower       upper 
## -0.54083919  0.04844404
```

```r
### Ingroup ###
describeBy(speech$Ingroup, speech$Convention)
```

```
## 
##  Descriptive statistics by group 
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 152 3.37 3.92      2     2.7 1.48   0  31    31  3.2    16.22 0.32
## -------------------------------------------------------- 
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 5.34 5.86      4    4.17 2.97   0  27    27 2.16     4.48 0.73
```

```r
t.test(speech$Ingroup ~ speech$Convention)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  speech$Ingroup by speech$Convention
## t = -2.475, df = 87.724, p-value = 0.01524
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.5614537 -0.3892042
## sample estimates:
## mean in group DNC mean in group RNC 
##          3.368421          5.343750
```

```r
cohen.d(speech$Ingroup, speech$Convention)
```

```
## 
## Cohen's d
## 
## d estimate: -0.4316381 (small)
## 95 percent confidence interval:
##      lower      upper 
## -0.7281919 -0.1350843
```

```
### Authority ###
describeBy(speech$Authority, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 152 3.16 3.58      2    2.52 1.48   0  23    23 2.49     8.49 0.29
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis  se
## X1    1 64 8.97 8.81    6.5    7.37 5.19   0  46    46 2.09     4.92 1.1
```

```
t.test(speech$Authority ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Authority by speech$Convention
## t = -5.1034, df = 71.909, p-value = 2.62e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.080715 -3.540995
## sample estimates:
## mean in group DNC mean in group RNC
##          3.157895          8.968750
```

```
cohen.d(speech$Authority, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -1.029319 (large)
## 95 percent confidence interval:
##      lower      upper
## -1.3388306 -0.7198077
```

```
### Purity ###
describeBy(speech$Purity, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 152  2.2 3.32      1    1.49 1.48   0  19    19  2.4     6.88 0.27
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 4.06 3.55      3    3.62 2.97   0  16    16  1.1     0.94 0.44
```

```
t.test(speech$Purity ~ speech$Convention)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  speech$Purity by speech$Convention
## t = -3.591, df = 111.45, p-value = 0.0004915
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.8943068 -0.8359564
## sample estimates:
## mean in group DNC mean in group RNC 
##          2.197368          4.062500
```

```
cohen.d(speech$Purity, speech$Convention)
```

```
## 
## Cohen's d
## 
## d estimate: -0.550352 (medium)
## 95 percent confidence interval:
##      lower      upper 
## -0.8486683 -0.2520357
```

Additional Analyses

For the DNC, I was able to acquire the Invocations and Benedictions. However, due to the nature of the purposes of these speeches, the text would inflate the appeals to religion, and thus scores on the purity foundation. For the next set of analyses, I remove these speeches to see if it influences the results in any way.

```
table(speech$Type)
```

```
## 
## benediction      speech       video 
##           7         208           1
```

```
speech <- speech[!(speech$Type == "benediction"), ]
```

Here, we can run the same analyses as above

```
### Harm ###
describeBy(speech$Harm, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean    sd median trimmed  mad min max range skew kurtosis
## X1    1 145 9.03 10.58      5    6.84 4.45   0  68    68 2.88      9.9
##      se
## X1 0.88
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 10.7 13.7      7    8.23 5.93   0  93    93 3.87     18.8 1.71
```

```
t.test(speech$Harm ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Harm by speech$Convention
## t = -0.86686, df = 97.539, p-value = 0.3881
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.488834  2.151549
## sample estimates:
## mean in group DNC mean in group RNC
##          9.034483         10.703125
```

```
cohen.d(speech$Harm, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -0.1436314 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.4398205  0.1525578
```

```
### Fairness ###
describeBy(speech$Fairness, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 145 2.04 3.02      1    1.38 1.48   0  18    18 2.72     8.73 0.25
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 2.84 4.68      1    1.87 1.48   0  31    31 3.76    18.19 0.59
```

```r
t.test(speech$Fairness ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Fairness by speech$Convention
## t = -1.2599, df = 86.888, p-value = 0.2111
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.0682271  0.4634857
## sample estimates:
## mean in group DNC mean in group RNC
##          2.041379          2.843750
```

```r
cohen.d(speech$Fairness, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -0.2224884 (small)
## 95 percent confidence interval:
##       lower       upper
## -0.51913043  0.07415357
```

```r
### Ingroup ###
describeBy(speech$Ingroup, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean sd median trimmed  mad min max range skew kurtosis   se
## X1    1 145 3.34  4      2    2.65 1.48   0  31    31 3.18    15.72 0.33
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 5.34 5.86      4    4.17 2.97   0  27    27 2.16     4.48 0.73
```

```r
t.test(speech$Ingroup ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Ingroup by speech$Convention
## t = -2.487, df = 89.912, p-value = 0.01473
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -3.5957313 -0.4021136
## sample estimates:
## mean in group DNC mean in group RNC
##          3.344828          5.343750
```

```r
cohen.d(speech$Ingroup, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -0.4305816 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.7293459 -0.1318172
```

```r
### Authority ###
describeBy(speech$Authority, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean   sd median trimmed  mad min max range skew kurtosis  se
## X1    1 145  3.2 3.63      2    2.56 1.48   0  23    23 2.46     8.21 0.3
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis  se
## X1    1 64 8.97 8.81    6.5    7.37 5.19   0  46    46 2.09     4.92 1.1
```

```r
t.test(speech$Authority ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Authority by speech$Convention
## t = -5.0532, df = 72.63, p-value = 3.13e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.044145 -3.493355
## sample estimates:
## mean in group DNC mean in group RNC
##           3.20000           8.96875
```

```r
cohen.d(speech$Authority, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -1.00745 (large)
## 95 percent confidence interval:
##      lower      upper
## -1.3188563 -0.6960441
```

```r
### Purity ###
describeBy(speech$Purity, speech$Convention)
```

```
##
##  Descriptive statistics by group
## group: DNC
##    vars   n mean  sd median trimmed  mad min max range skew kurtosis   se
## X1    1 145 1.88 2.8      1    1.32 1.48   0  17    17  2.4     7.35 0.23
## --------------------------------------------------------
## group: RNC
##    vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## X1    1 64 4.06 3.55      3    3.62 2.97   0  16    16  1.1     0.94 0.44
```

```r
t.test(speech$Purity ~ speech$Convention)
```

```
##
##  Welch Two Sample t-test
##
## data:  speech$Purity by speech$Convention
## t = -4.3471, df = 98.999, p-value = 3.349e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.174683 -1.184800
## sample estimates:
## mean in group DNC mean in group RNC
##          1.882759          4.062500
```

```r
cohen.d(speech$Purity, speech$Convention)
```

```
##
## Cohen's d
##
## d estimate: -0.7148291 (medium)
## 95 percent confidence interval:
##      lower      upper
## -1.0186177 -0.4110404
```

Graph of Results

I generate a graph to display the results of the analyses. These do not contain the invocation and benediction addresses.

Before I begin, I load some packages

```
# Load packages
library(car)
library(dplyr)
library(psych)
library(ggplot2)
library(GGally)
library("ggpubr")
library("reshape2")
library(scales)
```

I will display my data using bar graphs. To do this, I generate some summary statistics for each of the foundations

```
# Summary Statistice by convention

Harm <- speech %>% group_by(Convention) %>% summarize(mean = mean(Harm,
    na.rm = TRUE), sd = sd(Harm, na.rm = TRUE), n = n(), se = sd/sqrt(n),
    ci = qt(0.975, df = n - 1) * se) %>% mutate(type = "Harm")

Fairness <- speech %>% group_by(Convention) %>% summarize(mean = mean(Fairness,
    na.rm = TRUE), sd = sd(Fairness, na.rm = TRUE), n = n(),
    se = sd/sqrt(n), ci = qt(0.975, df = n - 1) * se) %>% mutate(type = "Fairness")

Ingroup <- speech %>% group_by(Convention) %>% summarize(mean = mean(Ingroup,
    na.rm = TRUE), sd = sd(Ingroup, na.rm = TRUE), n = n(), se = sd/sqrt(n),
    ci = qt(0.975, df = n - 1) * se) %>% mutate(type = "Ingroup")

Authority <- speech %>% group_by(Convention) %>% summarize(mean = mean(Authority,
    na.rm = TRUE), sd = sd(Authority, na.rm = TRUE), n = n(),
    se = sd/sqrt(n), ci = qt(0.975, df = n - 1) * se) %>% mutate(type = "Authority")

Purity <- speech %>% group_by(Convention) %>% summarize(mean = mean(Purity,
    na.rm = TRUE), sd = sd(Purity, na.rm = TRUE), n = n(), se = sd/sqrt(n),
    ci = qt(0.975, df = n - 1) * se) %>% mutate(type = "Purity")
```

To generate the graphs, I need to bind these summary statistics to one dataframe

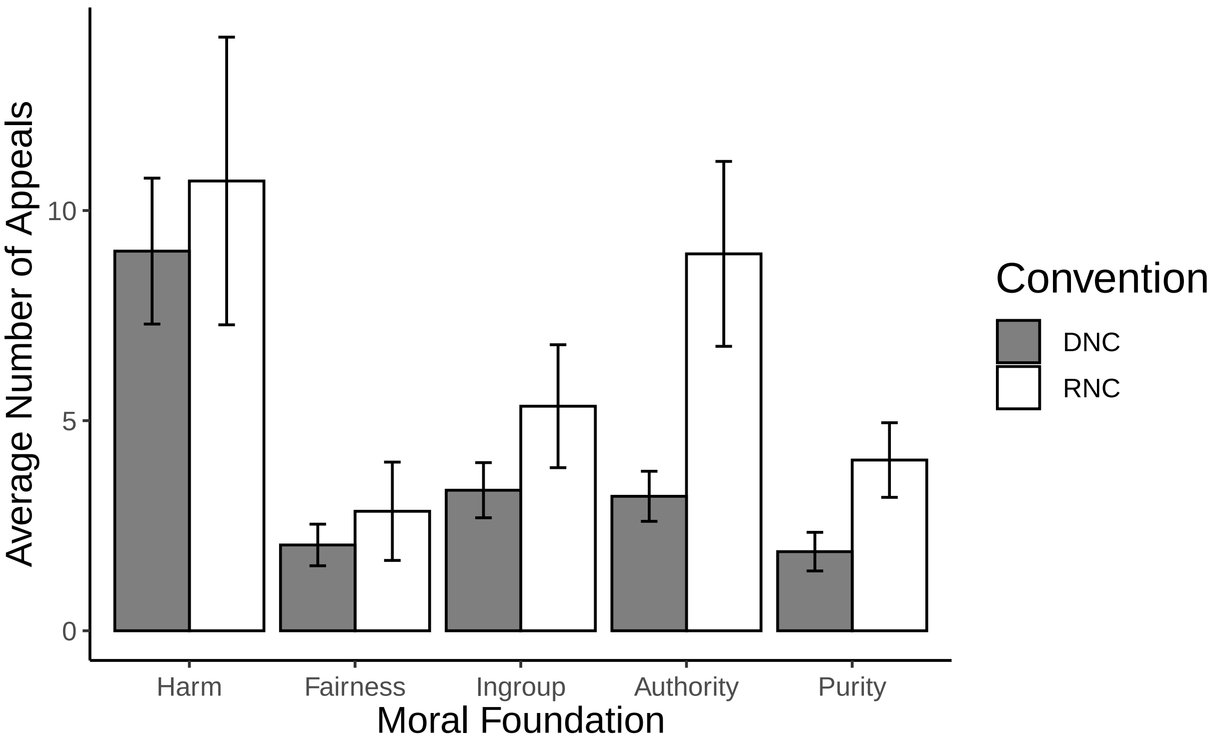
```
# Combine each of the outputs
token <- rbind(Harm, Fairness, Ingroup, Authority, Purity)

# Organize label order for the foundations
token$type <- factor(token$type, levels = c("Harm", "Fairness",
    "Ingroup", "Authority", "Purity"))
```

I generate the plot using `ggplot`

```
ggplot(token, aes(x = type, y = mean, fill = Convention)) + geom_bar(stat = "identity",
    position = position_dodge(), color = "black") + geom_errorbar(aes(ymin = mean -
    ci, ymax = mean + ci), width = 0.2, position = position_dodge(0.9)) +
    ggtitle("Moral Appeals in Political Speeches") + theme_classic() +
    xlab("Moral Foundation") + ylab("Average Number of Appeals") +
    labs(caption = "Source: 2016 RNC and DNC") + theme(text = element_text(size = 12,
    colour = "black"), axis.title = element_text(size = 14, colour = "black"),
    title = element_text(size = 16, colour = "black"), plot.caption = element_text(size
        color = "black"), axis.text.x = element_text(angle = 0,
        hjust = 0.5, vjust = 0.5), plot.title = element_text(hjust = 0.5)) +
    scale_x_discrete(labels = wrap_format(10)) + scale_fill_manual("Convention",
    values = c(DNC = "grey50", RNC = "white"))
```

Moral Appeals in Political Speeches



Source: 2016 RNC and DNC