Homework 2

Beau Harrison

Libraries

```
library(NLP) # Required for tm
library(tm) # Corpus
library(data.table) # rbindlist
library(quanteda) # tokenize
library(plyr) # join
library(readtext) # reading text files
```

Parse Files

```
# function takes DirSource for files and a string for speakerName
parseCorpus <- function(files, speakerName) {
  # Parse into R structures
  docs <- Corpus(files) # I didn't realize I was using the wrong Corpus until far too late, it works...
  if(length(files) > 1) {
    docFrames <- lapply(docs, function(doc) data.frame(doc$content))
    docFrame <- rbindlist(docFrames)
  } else {
    docFrame <- docs$content
  }
  # Clean up
  findString <- paste(char_toupper(speakerName), ': ')
  docFrame <- lapply(docFrame, function(text) gsub(findString, '', text))
  docFrame <- lapply(docFrame, function(text) gsub('\\([A-Z ]+\\)', '', text))
  # Tokenize
    # Seperate words and remove punctuation
  unigramTokens <- tokenize(paste(docFrame, collapse=''), removePunct=TRUE, removeNumbers=TRUE,
                            removeSymbols=TRUE, concatenator=' ')
  bigramTokens <- tokenize(paste(docFrame, collapse=''), removePunct=TRUE, removeNumbers=TRUE,
                           removeSymbols=TRUE, concatenator=' ', ngrams=2L)
    # Without stopwords
  unigramTokensNoStopwords <- removeFeatures(unigramTokens, stopwords('english'))
  bigramTokensNoStopwords <- removeFeatures(bigramTokens, stopwords('english'))
    # Put lower case versions in data.table
  unigrams <- data.table(token=tolower(unlist(unigramTokens)))
  bigrams <- data.table(token=tolower(unlist(bigramTokens)))
    # Without stopwords
  unigramsNoStopwords <- data.table(token=tolower(unlist(unigramTokensNoStopwords)))
  bigramsNoStopwords <- data.table(token=tolower(unlist(bigramTokensNoStopwords)))
    # Count instances
  unigramCount <- unigrams[, .N, by=token][order(N, decreasing=TRUE)]
  bigramCount <- bigrams[, .N, by=token][order(N, decreasing=TRUE)]
    # Without stopwords
  unigramCountNoStopwords <- unigramsNoStopwords[, .N, by=token][order(N, decreasing=TRUE)]
  bigramCountNoStopwords <- bigramsNoStopwords[, .N, by=token][order(N, decreasing=TRUE)]
    # Add canidate names
```

```
  unigramCount$canidate <- speakerName
  bigramCount$canidate <- speakerName
    # Without stopwords
  unigramCountNoStopwords$canidate <- speakerName
  bigramCountNoStopwords$canidate <- speakerName
  # Return all four sets of tokens
  return(list('unigramCount'=unigramCount, 'bigramCount'=bigramCount,
    'unigramCountNoStopwords'=unigramCountNoStopwords, 'bigramCountNoStopwords'=bigramCountNoStopwords))
}
```

Chi^2 calculation

```
chiSquared <- function(input) {
  DT <- data.table(join(input[canidate == "Clinton"][, list(token, clintonCount = as.numeric(N))],
                  input[canidate == "Trump"][, list(token, trumpCount = as.numeric(N))], type="full"))
  DT[is.na(clintonCount)]$clintonCount <- 0
  DT[is.na(trumpCount)]$trumpCount <- 0
  DT[, `:=`(totalCount, clintonCount + trumpCount)]
  DT <- DT[order(totalCount, decreasing=TRUE)][totalCount > 5]
  DT[, `:=`(totalClinton, sum(clintonCount))]
  DT[, `:=`(totalTrump, sum(trumpCount))]
  DT[, `:=`(chi2, (totalClinton + totalTrump) * (trumpCount * (totalClinton - clintonCount)
                - clintonCount * (totalTrump - trumpCount))^2/((trumpCount + clintonCount)
                * (trumpCount + (totalTrump - trumpCount)) * (clintonCount
                + (totalClinton - clintonCount)) * ((totalTrump - trumpCount)
                + (totalClinton - clintonCount))))]

  return(DT[order(chi2, decreasing=TRUE)])
}
```

Apply functions

```
clintonList <- parseCorpus(DirSource('CampaignSpeeches',pattern='clinton'), 'Clinton')
trumpList <- parseCorpus(DirSource('CampaignSpeeches',pattern='trump'), 'Trump')
# Combine data.tables
unigramCount <- rbind(clintonList$unigramCount, trumpList$unigramCount)[order(N, decreasing=TRUE)]
bigramCount <- rbind(clintonList$bigramCount, trumpList$bigramCount)[order(N, decreasing=TRUE)]
# Without stopwords
unigramCountNoStopwords <- rbind(clintonList$unigramCountNoStopwords,
                                 trumpList$unigramCountNoStopwords)[order(N, decreasing=TRUE)]
bigramCountNoStopwords <- rbind(clintonList$bigramCountNoStopwords,
                                trumpList$bigramCountNoStopwords)[order(N, decreasing=TRUE)]
```

Question 1 Results

```
unigramChi <- chiSquared(unigramCount)
unigramChi[1:10]
```

```
##        token clintonCount trumpCount totalCount totalClinton totalTrump
##  1:    going           40        233        273        18897      22149
```

```
##  2:     very           18        127        145        18897      22149
##  3:    women           70         10         80        18897      22149
##  4: families           46          2         48        18897      22149
##  5:    let's           48          3         51        18897      22149
##  6:  economy           46          4         50        18897      22149
##  7:  hillary            9         73         82        18897      22149
##  8:      i'm           31        119        150        18897      22149
##  9:    don't           24        103        127        18897      22149
## 10:       as          132         63        195        18897      22149
##          chi2
##  1: 108.97915
##  2:  66.22454
##  3:  55.46514
##  4:  47.96352
##  5:  47.51346
##  6:  42.56775
##  7:  40.66069
##  8:  39.01061
##  9:  37.77413
## 10:  36.97961
```

```
bigramChi <- chiSquared(bigramCount)
bigramChi[1:10]
```

```
##               token clintonCount trumpCount totalCount totalClinton
##  1:        going to           31        208        239         5949
##  2:      each other           18          0         18         5949
##  3:    young people           19          1         20         5949
##  4:       i believe           19          1         20         5949
##  5:         look at            4         42         46         5949
##  6: hillary clinton            6         47         53         5949
##  7:       we should           20          2         22         5949
##  8:          i mean            1         29         30         5949
##  9:           do it            2         32         34         5949
## 10:         the top           14          0         14         5949
##     totalTrump     chi2
##  1:       7941 88.54452
##  2:       7941 24.05841
##  3:       7941 22.26349
##  4:       7941 22.26349
##  5:       7941 21.96095
##  6:       7941 21.57149
##  7:       7941 20.80273
##  8:       7941 19.15370
##  9:       7941 19.00148
## 10:       7941 18.70670
```

```
# Without stopwords
unigramChiNoStopwords <- chiSquared(unigramCountNoStopwords)
unigramChiNoStopwords[1:10]
```

```
##        token clintonCount trumpCount totalCount totalClinton totalTrump
##  1:    going           40        233        273         8444      10279
##  2:    women           70         10         80         8444      10279
##  3: families           46          2         48         8444      10279
```

```
##  4:    let's          48          3         51         8444      10279
##  5:  economy          46          4         50         8444      10279
##  6: together          65         17         82         8444      10279
##  7:  hillary           9         73         82         8444      10279
##  8:      i'm          31        119        150         8444      10279
##  9:    don't          24        103        127         8444      10279
## 10:     work          83         32        115         8444      10279
##          chi2
##  1: 103.72883
##  2:  58.33660
##  3:  50.02664
##  4:  49.62704
##  5:  44.53855
##  6:  38.83545
##  7:  38.73395
##  8:  36.45759
##  9:  35.45515
## 10:  34.25626
```

```
bigramChiNoStopwords <- chiSquared(bigramCountNoStopwords)
bigramChiNoStopwords[1:10]
```

```
##               token clintonCount trumpCount totalCount totalClinton
##  1:        going to           31        208        239         5949
##  2:      each other           18          0         18         5949
##  3:    young people           19          1         20         5949
##  4:       i believe           19          1         20         5949
##  5:         look at            4         42         46         5949
##  6: hillary clinton            6         47         53         5949
##  7:       we should           20          2         22         5949
##  8:          i mean            1         29         30         5949
##  9:           do it            2         32         34         5949
## 10:         the top           14          0         14         5949
##     totalTrump     chi2
##  1:       7941 88.54452
##  2:       7941 24.05841
##  3:       7941 22.26349
##  4:       7941 22.26349
##  5:       7941 21.96095
##  6:       7941 21.57149
##  7:       7941 20.80273
##  8:       7941 19.15370
##  9:       7941 19.00148
## 10:       7941 18.70670
```

Parse into R structures

```
# Get files into R
clintonListOrlando <- parseCorpus(DirSource('CampaignSpeeches',pattern='clinton-orlando'), 'Clinton')
trumpListOrlando <- parseCorpus(DirSource('CampaignSpeeches',pattern='trump-orlando'), 'Trump')
# Combine data.tables
unigramCountOrlando <- rbind(clintonListOrlando$unigramCount,
                             trumpListOrlando$unigramCount)[order(N, decreasing=TRUE)]
```

```
bigramCountOrlando <- rbind(clintonListOrlando$bigramCount,
                            trumpListOrlando$bigramCount)[order(N, decreasing=TRUE)]
# Without stopwords
unigramCountNoStopwordsOrlando <- rbind(clintonListOrlando$unigramCountNoStopwords,
                trumpListOrlando$unigramCountNoStopwords)[order(N, decreasing=TRUE)]
bigramCountNoStopwordsOrlando <- rbind(clintonListOrlando$bigramCountNoStopwords,
                trumpListOrlando$bigramCountNoStopwords)[order(N, decreasing=TRUE)]
```

Question 2 Results

```
unigramOrlandoChi <- chiSquared(unigramCountOrlando)
unigramOrlandoChi[1:10]
```

```
##           token clintonCount trumpCount totalCount totalClinton totalTrump
##  1: immigration            0         21         21         2113       2977
##  2:       don't            1         21         22         2113       2977
##  3:         and          164        159        323         2113       2977
##  4:        that           52         36         88         2113       2977
##  5:          as           28         14         42         2113       2977
##  6:         she            2         21         23         2113       2977
##  7:       those           11          2         13         2113       2977
##  8:    together           11          2         13         2113       2977
##  9:          up            7          0          7         2113       2977
## 10:        well            9          1         10         2113       2977
##          chi2
##  1: 14.967024
##  2: 12.436464
##  3: 12.183418
##  4: 11.396193
##  5: 11.036097
##  6: 10.248323
##  7:  9.972830
##  8:  9.972830
##  9:  9.875863
## 10:  9.702107
```

```
bigramOrlandoChi <- chiSquared(bigramCountOrlando)
bigramOrlandoChi[1:10]
```

```
##                token clintonCount trumpCount totalCount totalClinton
##  1: first responders            6          0          6          329
##  2:            a lot            6          0          6          329
##  3:          as well            6          0          6          329
##  4:       each other            6          0          6          329
##  5:          that we           10          3         13          329
##  6:           we are            7          1          8          329
##  7:           i have            7          1          8          329
##  8:         we don't            0         12         12          329
##  9:          will be            0         11         11          329
## 10:           of the            2         17         19          329
##     totalTrump      chi2
##  1:        547 10.044482
##  2:        547 10.044482
```

```
##  3:        547 10.044482
##  4:        547 10.044482
##  5:        547  8.719754
##  6:        547  8.587115
##  7:        547  8.587115
##  8:        547  7.317794
##  9:        547  6.700223
## 10:        547  6.050879
```

```
# Without stopwords
unigramOrlandoChiNoStopwords <- chiSquared(unigramCountNoStopwordsOrlando)
unigramOrlandoChiNoStopwords[1:10]
```

```
##           token clintonCount trumpCount totalCount totalClinton totalTrump
##  1:    together           11          2         13          596       1027
##  2: immigration            0         21         21          596       1027
##  3:        well            9          1         10          596       1027
##  4:       first            8          1          9          596       1027
##  5:       learn            6          0          6          596       1027
##  6:  responders            6          0          6          596       1027
##  7:         lot            6          0          6          596       1027
##  8:       don't            1         21         22          596       1027
##  9:        back            7          1          8          596       1027
## 10:     islamic            0         13         13          596       1027
##          chi2
##  1: 12.936155
##  2: 12.346706
##  3: 12.291316
##  4: 10.599020
##  5: 10.377290
##  6: 10.377290
##  7: 10.377290
##  8:  9.936931
##  9:  8.920834
## 10:  7.605221
```

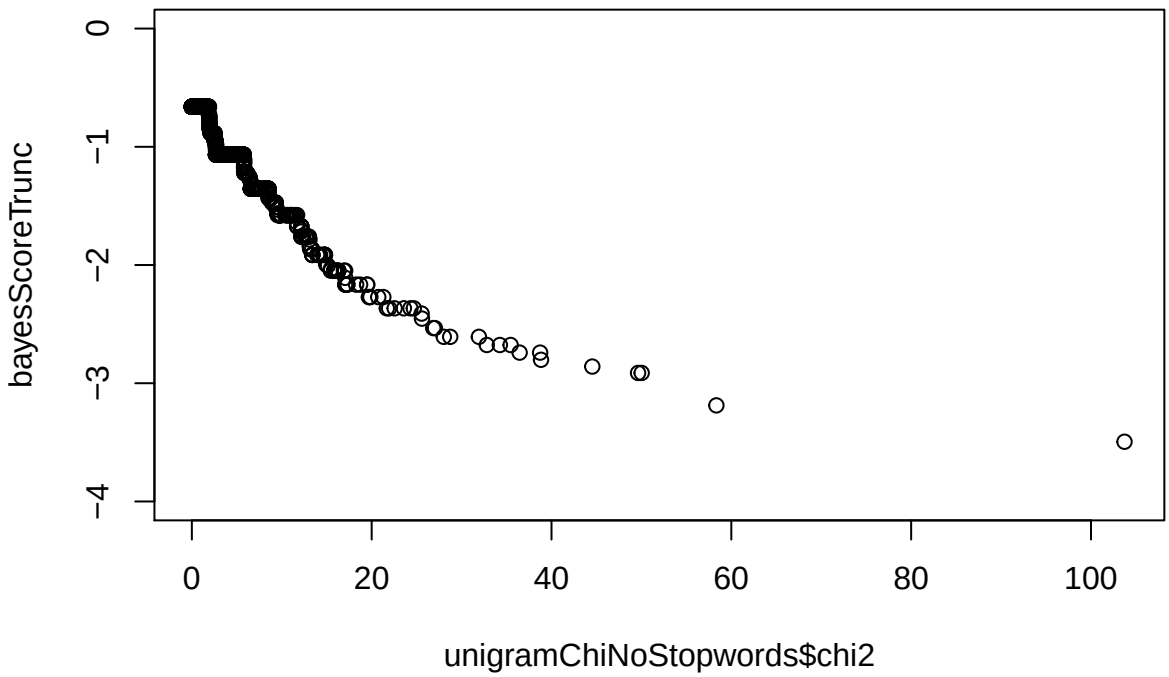
```
bigramOrlandoChiNoStopwords <- chiSquared(bigramCountNoStopwordsOrlando)
bigramOrlandoChiNoStopwords[1:10]
```

```
##                token clintonCount trumpCount totalCount totalClinton
##  1: first responders            6          0          6          329
##  2:            a lot            6          0          6          329
##  3:          as well            6          0          6          329
##  4:       each other            6          0          6          329
##  5:          that we           10          3         13          329
##  6:           we are            7          1          8          329
##  7:           i have            7          1          8          329
##  8:         we don't            0         12         12          329
##  9:          will be            0         11         11          329
## 10:           of the            2         17         19          329
##     totalTrump      chi2
##  1:        547 10.044482
##  2:        547 10.044482
##  3:        547 10.044482
##  4:        547 10.044482
```

```
##  5:        547  8.719754
##  6:        547  8.587115
##  7:        547  8.587115
##  8:        547  7.317794
##  9:        547  6.700223
## 10:        547  6.050879
```

Question 3 Results

```
REFERENCE <- corpus(readtext("CampaignSpeeches/*.txt", docvarsfrom="filenames"))
REFERENCE.dfm <- dfm(REFERENCE, tolower=TRUE, removeNumbers=TRUE, removePunct=TRUE,
                     removeSeparators=TRUE, stem=TRUE, remove=stopwords("english"))
refscores <- c(rep(-1,7), rep(1,6))
bs <- textmodel(REFERENCE.dfm, refscores, model="NB", smooth=1)
bayesScore <- sort(log(bs$PwGc[1, ]/bs$PwGc[2, ]), decreasing=FALSE) # Sort for Trump
# Plot Parameters
xMax <- ceiling(max(unigramChiNoStopwords$chi2))
xMin <- floor(min(unigramChiNoStopwords$chi2))
maxPoints <- length(unigramChiNoStopwords$chi2)
bayesScoreTrunc <- bayesScore[1:maxPoints] # Need to limit size of bayes score to match chi2
yMax <- ceiling(max(bayesScoreTrunc))
yMin <- floor(min(bayesScoreTrunc))
# Plot
plot(unigramChiNoStopwords$chi2, bayesScoreTrunc, xlim=c(xMin,xMax), ylim=c(yMin,yMax))
```



Question 4 Results

```
devosArticles <- corpus(readtext("DevosArticles/*.txt", docvarsfrom="filenames"))
devosArticles.dfm <- dfm(devosArticles, tolower=TRUE, removeNumbers=TRUE, removePunct=TRUE,
                         removeSeparators=TRUE, stem=TRUE, remove=stopwords("english"))
devosRefscores <- c(1,-1)
devosWs <- textmodel(devosArticles.dfm, devosRefscores, model="wordscores", smooth=1)
devosBs <- textmodel(devosArticles.dfm, devosRefscores, model="NB", smooth=1)
# Predict
predict(devosWs, devosArticles.dfm)
```

```
## Predicted textmodel of type: wordscores
##
##         textscore LBG se   ci lo   ci hi
## Fox.txt    0.2651 0.0144  0.2369  0.2934
## NYT.txt   -0.1985 0.0090 -0.2162 -0.1808
```

```
predict(devosBs, devosArticles.dfm)
```

```
## Predicted textmodel of type: Naive Bayes
##
##              lp(1)     lp(-1)     Pr(1) Pr(-1) Predicted
## Fox.txt -2154.5317 -2376.3014    1.0000 0.0000         1
## NYT.txt -5234.1854 -4850.7314    0.0000 1.0000        -1
```