PH125.9x Choose Your Own Report Submission
Analysis of the Debates
42nd Parliament of Canada

G. W. Boone

May 7, 2020

Contents

Introduction

This report is the output of the final project from the Professional Certificate in Data Science HarvardX course of studies. The final project is an analysis of data as selected by the course participant. The focus for this report is qunatitatve analysis of the text of the debates of the 42nd Parliament of Canada. Analysis of the debates was selected to demonstrate comprehension of course material while using a less common data set in execution of the demonstration.

Although the debates of the 42nd Parliament of Canada are available for download in a .csv format the step was taken to build a script to scrape the data used in this analysis from ourcommons.ca website. Creation of the script demonstrates data collection from dynamic source materials, inclusion of elements in the data not available in the curated data sets and specifying the range of data to be collected.

In total the data collection script captured approximetly 150,000 records. Each record contains the fully transcribed text of the words spoken by the individual in the course of the debates in Canada's House of Commons. The script used for data collection is included as an Appendix to this report.

For the purposes of this report, a 15% random sample of the full data set captured by the data collection script is used. 15% equates to a sample consisting of 22,509 records of text of spoken words in the debates. The 22,509 records provide 2.46 million unique tokens (words) for analysis.

The analysis seeks to establishes closeness in the spoken content of the debate participants, ratio of positive sentiments to negative sentiments in debates, the frequency of spoken words by party affiliation, and supervised categorization of text content by party affiliation.

Methods And Analysis

Data Exploration

The underlying rmd file used to generate this report downloads the source files for generation of the analysis completed in this report.

An archive of the source file for the analysis is also made available at https://github.com/gwboone/ph125.9x-CYO/raw/master/CYO.zip

Detect And Handle Anomalies

Assessing the number of records by party provides a sense of how often individuals are speaking in the debates. An anomaly was detected where the person speaking affiliation is assigned to an individual, not the party they are affiliated. Closer inspection of the text reveals this is a special circumstance where the person speaking is nominating a person to function as Speaker of the House for the purposes of electing a permanent Speaker for the Parliament.

Table 1: debate counts

personSpeakingAffiliation	n
BQ	447
CCF	58
Charles Robert 2019-12-05 9:41 [p.1] Expand ...More Honourable members, pursuant to Standing Order 3, I invite Mr. Louis Plamondon, member for the electoral district of Bécancour—Nicolet—Saurel, to take the chair as the member presiding over the election of the Speaker. ...LessCollapse Clerk of the House of CommonsElection of the SpeakerPlamondon, LouisPresiding Officer for election of the SpeakerReferences to members	1

After removing the special circumstance record the summary of speakers by party shows those participating in debates are dominated by the 1st (government), 2nd(official opposition) and 3rd (the 3rd party) parties making up the membership of the hours of commons. The minor parties participate in debates a much lesser degree than major parties.

Table 2: debate counts

personSpeakingAffiliation	n
Lib.	9630
CPC	7315
NDP	4317
BQ	447
Ind.	374
GP	342
CCF	58
PPC	18
FD	7

Isolate Records For 42nd Parliament

The scope of debates to be analyzed is the 42nd Parliament of Canada. The data collection script captures the date of the person speaking as 'personSpeakingDate'. This value provides a convenient item to determine what Parliament number the person was speaking. The date ranges of Canada's most recent parliaments is available as a wiki entry here.

The debates data can be mutated to include the parliament number based on the date the person is recorded as speaking with the following code.

```r
debates <- debates %>%
      mutate(parl = ifelse(between(personSpeakingDate,"2001-01-29", "2004-05-23"),"37",
        ifelse(between(personSpeakingDate,"2004-10-04", "2005-11-29"),"38",
        ifelse(between(personSpeakingDate,"2006-04-03", "2008-09-07"),"39",
        ifelse(between(personSpeakingDate,"2008-11-18", "2011-03-26"),"40",
        ifelse(between(personSpeakingDate,"2011-06-02", "2015-08-02"),"41",
        ifelse(between(personSpeakingDate,"2015-12-03", "2019-09-11"),"42",
        "43")))))))
```

With an identifier for the Parliament avaiable in the data set it is possible to see how many speakers were recorded by Parliament session. It is quickly observed the data sample is heavily weighted to the 42nd Parliament. If other Parliaments were of interest for analysis, the data collection script in the Appendix can be modified to target specific parliaments by augmenting the URL "&ParlSes=All" to reflect the Parliament of interest. For example replacing 'All' with 39 (=L&Item=&ParlSes=39) to collect debates from the 39th Parliament of Canada.

Table 3: debate by parliament

parl	n
42	18144
41	2883
43	1481

We can explore who speaks most. Is there an individual who dominates times speaking in the House of Commons?

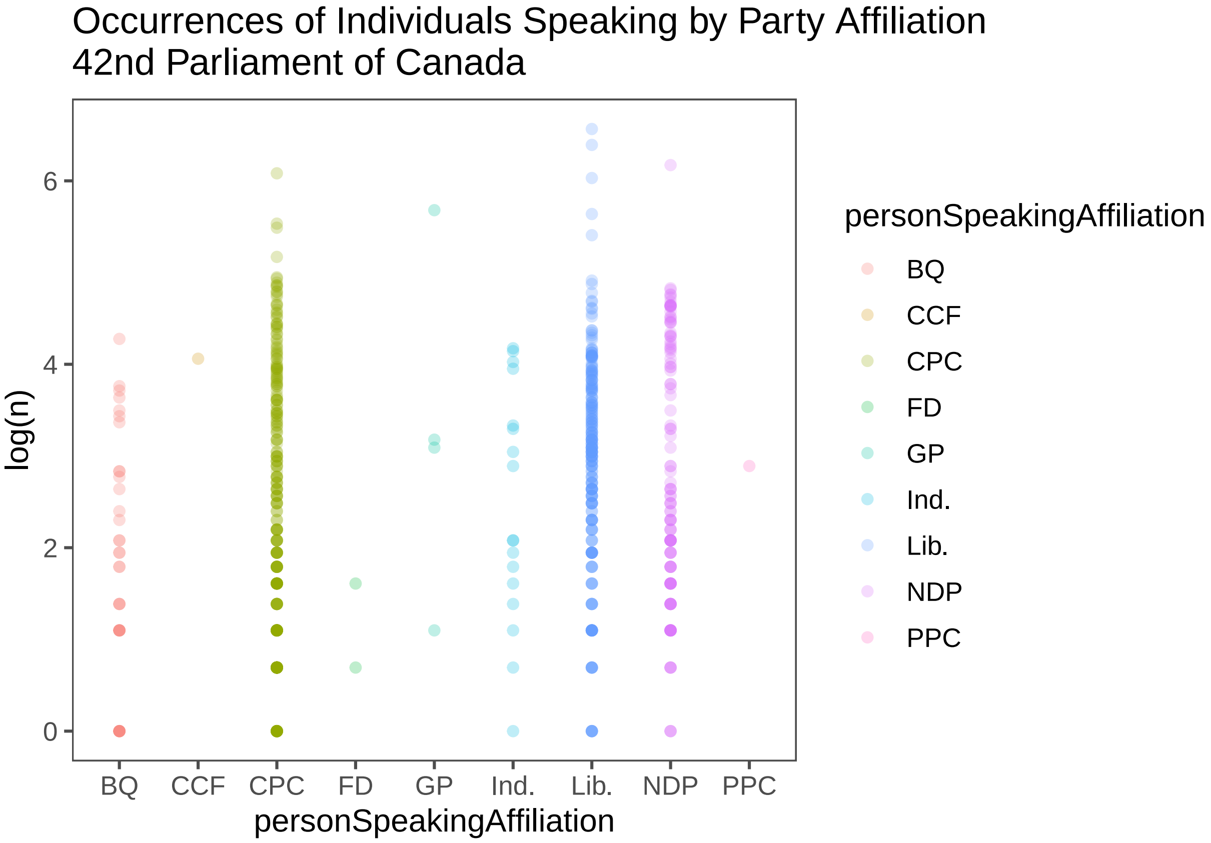
Table 4: who speaks the most often?

personSpeaking	personSpeakingAffiliation	n
Hon. Geoff Regan	Lib.	1059
Kevin Lamoureux	Lib.	710
Right Hon. Justin Trudeau	Lib.	597
Carol Hughes	NDP	479
Bruce Stanton	CPC	438

There are a couple of contenders for most often speaking reveled in the top 5 of occurrences of speaking in debates. Most notably one member, Geoff Regan, has nearly twice as many occurrences speaking than the Prime Minister during the 42nd Parliament (Justin Trudeau). Cross referencing with ourcommons.ca we see that Geoff Regan had the role of Speaker in the 42nd Parliament. The Speaker presides over the House of Commons and ensures that everyone respects its rules and traditions. The Speaker must be impartial and apply the rules to all Members equally. Not being an active participant in debates, Geoff Regan will be removed from the records for analysis.

Kevin Lamoureux also shows a statistically higher number of occurrences speaking than other members of the House. Cross checking ourcommons.ca we find that Kevin Lamoureux had the role of Parliamentary Secretary to the Leader of the Government in the House of Commons. This is a role in the House of Commons appointed by the Prime Minister to help Cabinet Ministers. Parliamentry Secretaries table documents or answer questions for a Minister, participate in debates on bills, attend committee meetings, speak on government policies and proposals, and serve as a link between parliamentarians and Ministers. This being the case, records identified as Kevin Lamoureux as the person speaking will be kept for analysis.

With the data scrubbed to remove occurrences that are not of interest or could skew analysis the data can be plotted to show the occurrences of persons speaking in the House of commons during debates. The log of occurrences of speaking provides a scale that shows the occurrences of individuals speaking by party affiliation.



As a final step in our data exploration and clean up we can select only those fields of interest for analysis from the set of debates records. This will help with processing the text in quantitative analysis by eliminating non value add items for this analysis such the document ID and the time stamp of the when the person was recorded as speaking in the House of Commons.

In the following section, Quantitative Analysis with the Quanteda, the advantage of the Quanteda's `readtext` function in working with text data will be explained. To format the debate data for `readtext` the selected data is written locally as a tab separated values file CYO.Debates.tsv

Quanteda for Quantitative Text Analysis

Creation of Corpus

The primary package applied to the text analysis of the debates of the 42nd Parliament of Canada is the Quanteda package for R. The Quanteda.org About page describes the Quanteda Initiative as a UK non-profit organization devoted to the promotion of open-source text analysis software. The initiative supports active development of these tools, in addition to providing training materials, training events, and sponsoring workshops and conferences. Its main product is the open-source quanteda package for R, but the Quanteda Initiative also supports a family of independent but interrelated packages for providing additional functionality for natural language processing and document management.

The basic workflow in applying Quanteda in text analysis is creation of the `corpus`, `tokens` and `document feature matrix` objects. For this analysis:

- The Corpus is a collection of original texts and document-level variables from the CYO.debates.tsv file.
- Tokens are words extracted from the corpus created from the CYO.debates.tsv file
- Document Feature Matrix representing frequencies of features in documents (rows found in CYO.debates.tsv) in a matrix

The data used for analysis is read with Quanteda's readtext package. Using readtext provides the advantage of using a function for reading the data that has been developed especially for Quanteda. The readtext package also provides the convenience of arguments for identifying the 'text' for analysis and document level variables (docvars) that will be used in grouping the data in later stages of text analysis. The corpus is created using Quanteda's corpus command

```r
# Read the data that will be used for analysis
    debates <- readtext("CYO.Debates.tsv",
      text_field = "speakingContent",
      docvarnames = c("personSpeakingAffiliation",
      "personSpeakingDate",
      "personSpeaking",
      "parl"))

# Create a corpus of the texts using quanteda corpus
    corp_debates <- corpus(debates)
```

Training & Test Sets

As the proportion of the total records collected from ourparliament.ca for analysis is 15% of the total records collected the Training set has been assigned as 90% of the records for analysis and 10% assigned to test. The 90/10 split has been used to provide adequate records for training planned classification of documents.

A reproducible random sample of records ids comprising of 10% of the total number of observations in corpus_debates is assigned to vector id_test. Creation of corp_debates$id_numeric facilitates matching test_ids to create a corpus of test records. Similarly, a corpus of unmatched records in id_test results in the creation of the training corpus.

Creation of training and test document frame matrix (dfm) in the same code chunk provides a final output of training and test corpus and document frame matrix. The corpus and dfms will be used for the majority of the analysis completed from this point forward. The test corpus is exclusively for use as final validation of the classification algorithm.

```r
# generate test id numbers without replacement
  set.seed(50)
  id_test <- sample(1:17008, 1708, replace = FALSE)

# create docvar with ID
  corp_debates$id_numeric <- 1:ndoc(corp_debates)

# 90% training set (documents not in id_test).
  corp_training <- corpus_subset(corp_debates, !id_numeric %in% id_test)

# Training Data Frame Matrix
  dfmat_training <- corp_training %>%
   dfm(remove = stopwords("english"),
     stem = F,
     remove_punct = T,
     remove_symbols = T,
     remove_numbers = T,
     include_docvars = T)

# 10% test set (documents in id_test) # Using pipe we can create the corpus and data frame matrix in on
  corp_test <- corpus_subset(corp_debates, id_numeric %in% id_test)

# Test Data Frame Matrix
  dfmat_test  <- corp_test %>%
   dfm(remove = stopwords("english"),
     stem = F,
     remove_punct = T,
     remove_symbols = T,
     remove_numbers = T,
     include_docvars = T)
```

Training Data Exploration

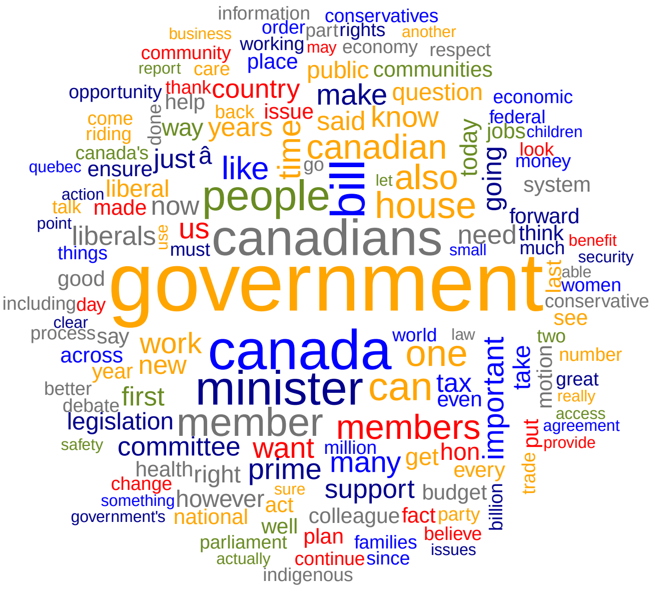
Frequency Analysis

Word Cloud

A word cloud or text cloud is an effective presentation of the frequency of words as they appear in the debates text. The bigger and bolder the word appears, the more often it's mentioned within the debate text. Not surprisingly, the words shown in the analysis predominately are House terms such as member, Canadians, committee etc. There are interesting terms showing in the cloud such as indigenous, economic, support and health. These words used repeatedly during the 42nd Parliament over a period of nearly 4 years is an insight to what is being said or mentioned in the course of debates in The House of Commons. The training dfm is trimmed and the frequency of words ranked for generation of the word cloud graphic.

```
# Explore training data - Wordclouds give a visual sense of word occurance
# Apply dfm_trim to issolate the top 20% of reoccuring words
    dfmat_debates_wrdcld <- dfm_trim(dfmat_training,
                            termfreq_type = "rank")

    set.seed(1)
    textplot_wordcloud(dfmat_debates_wrdcld,
                    #min_size = 2,
                    max_words = 150,
                    random_color = T,
                    color = c("blue", "red", "orange", "darkblue", "gray45","olivedrab4" ))
```

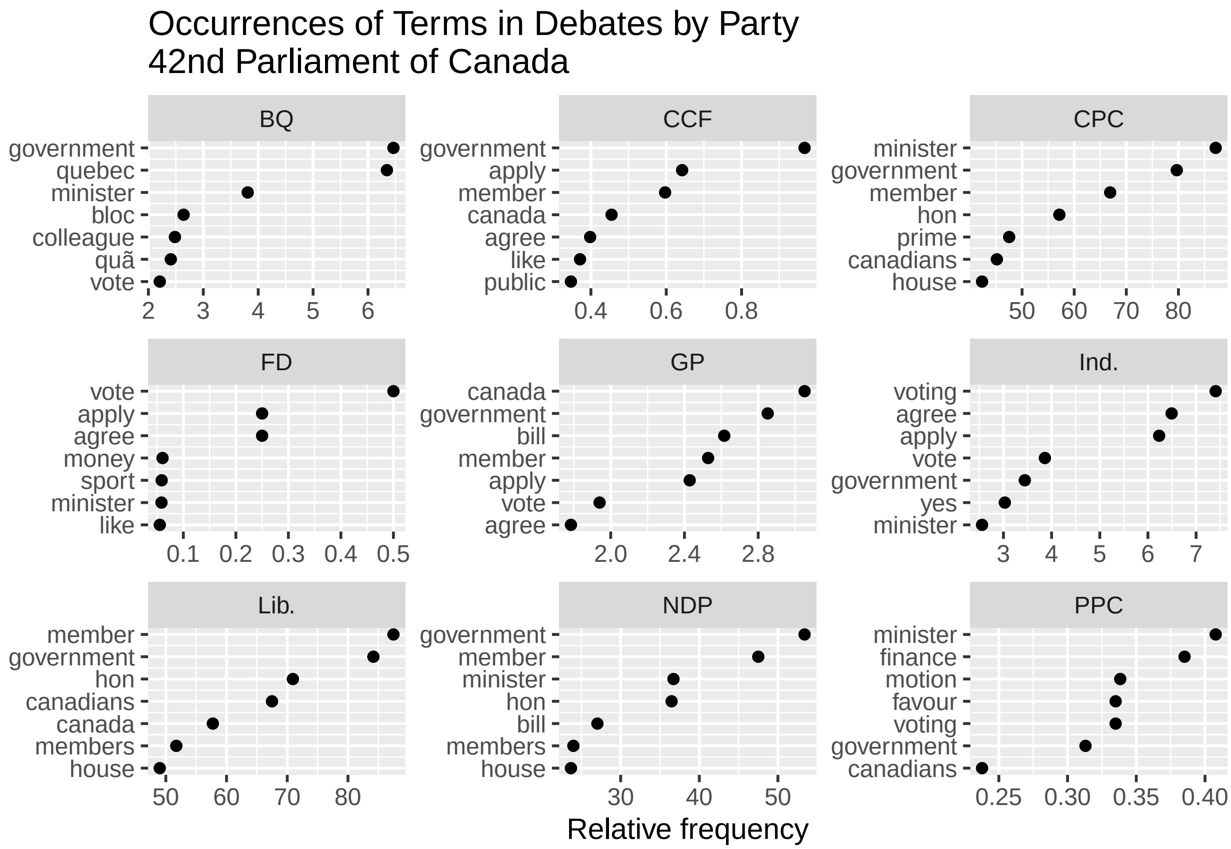


Terms By Party Affiliation

The inclusion of docvars() in the corpus enables grouping terms by any of the docvars() present in the corpus. The debates_corpus includes docvars() recording personSpeaking, personSpeakingAffilliation and the personSpeakingDate. Applying the personSpeakingAffiliation docvar the most frequent terms spoken in debates can be plotted by the party the person speaking is affiliated. The dfm_weght() function has an argument to measure the dfm results as the proportion of the feature counts of total feature counts. In other words, provide the relative frequency calculated as $\sum_{ij} / \sum_{j} tf_{ij}$

The resulting dfm, when used with textstat_frequency() argument groups set as personSpeakingAffiliation can be plotted to show the frequency of terms used over the course of the 42nd Parliament faceted by party affiliation.

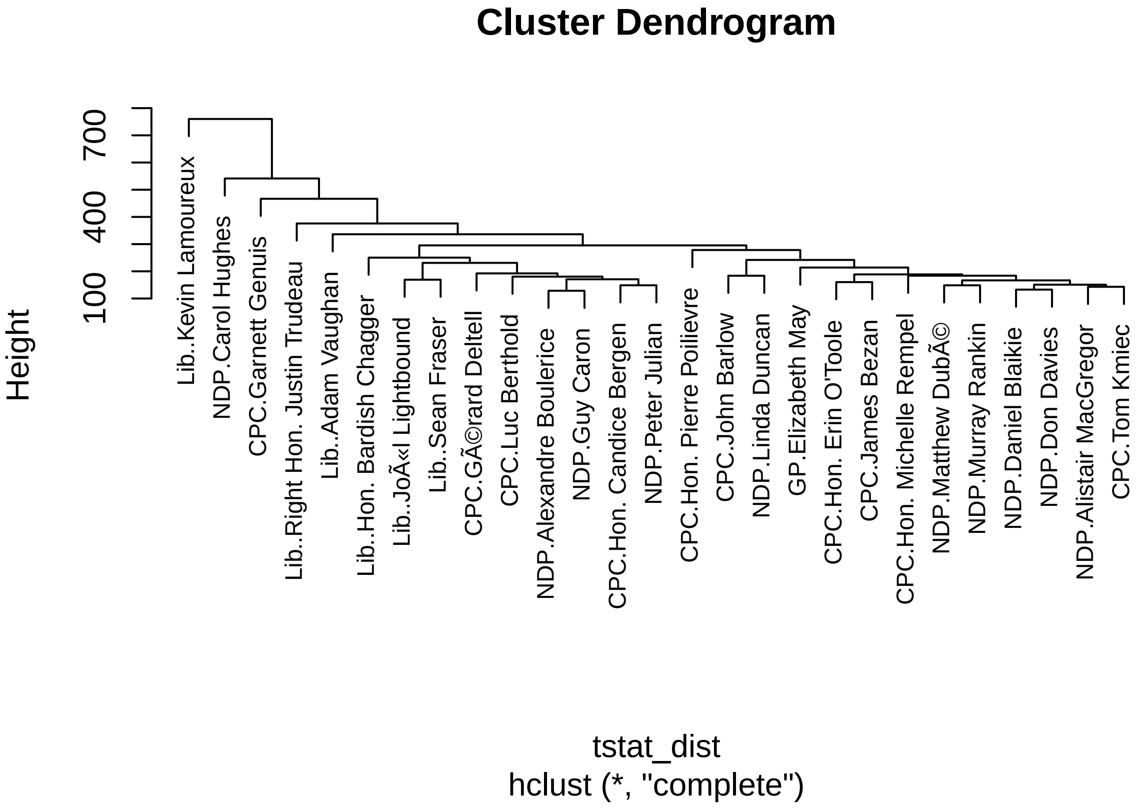
```r
# Words by party affiliation
dfm_debates_pty_prop <- dfmat_training %>% dfm_weight(scheme = "prop")
# Calculate relative frequency by party
freq_weight <- textstat_frequency(dfm_debates_pty_prop, n = 7,
                                  groups = "personSpeakingAffiliation")
ggplot(data = freq_weight, aes(x = nrow(freq_weight):1, y = frequency)) +
  geom_point() + facet_wrap(~ group, scales = "free") + coord_flip() +
  scale_x_continuous(breaks = nrow(freq_weight):1,labels = freq_weight$feature) +
  labs(x = NULL, y = "Relative frequency") +
  ggtitle("Occurrences of Terms in Debates by Party\n42nd Parliament of Canada")
```



Similarity Between People Speaking

The document feature matrix can also be used to inspect the closeness of terms used by those persons speaking in debates. The closeness of terms used in speaking is then able to be shown in clusters visually as a Cluster Dendrogram. Interesting groupings are revealed in the resulting plot. For example, Green Party leader Elizabeth May is clustered with New Democratic Party member Murray Rankin. Of particular interest is they are both close to Bardish Chagger who is a ranking member of the Liberal government as a House leader. To the other extreme Conservative Party of Canada member Garnett Genius is clustered with Liberal Kevin Lamoureux who we identified as a Parliamentary Secretary appointed by the Prime Minister. It should be noted the Dendrogram can be refined or expanded based on the arguments controlling the minimum number of characters needed to be considered a word and the minimum and maximum number of times the terms being cluster must occur across all documents in the dfm.

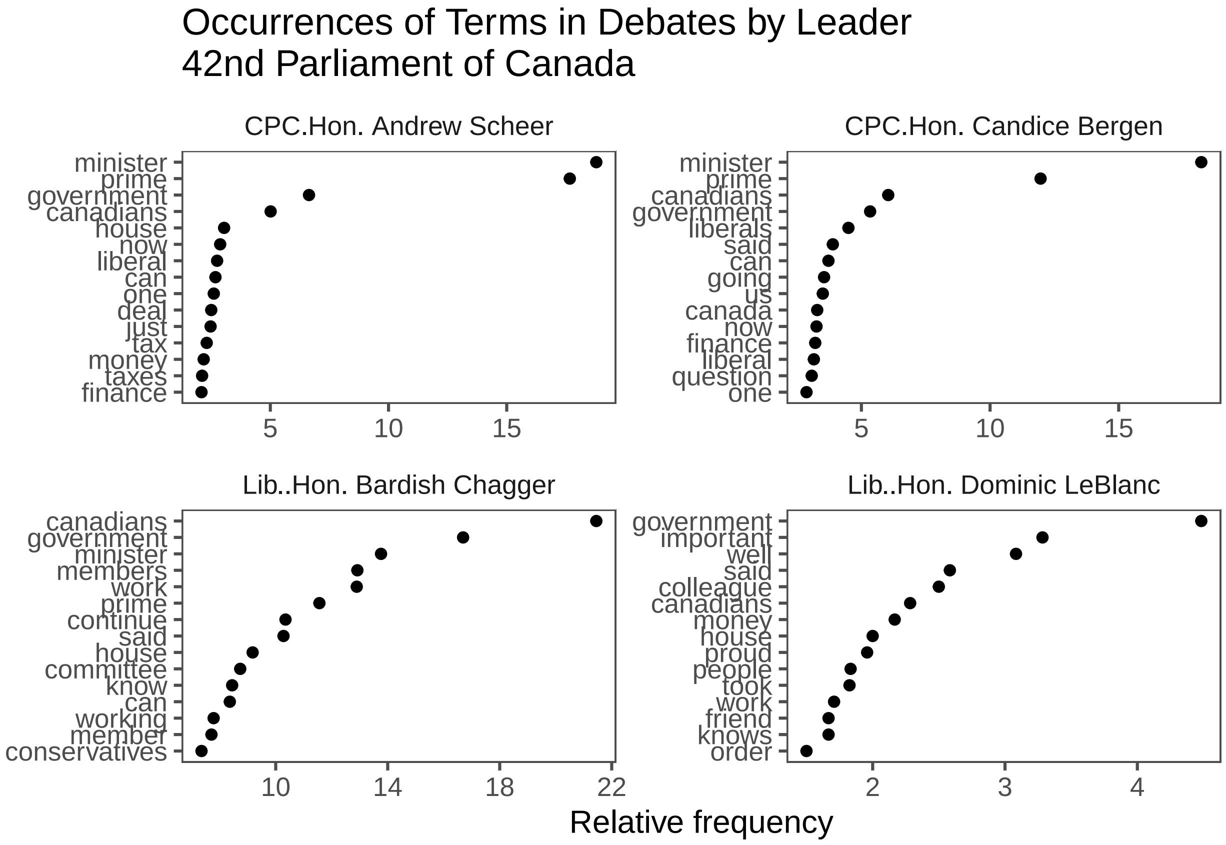
```
## Similarity between people speaking.  ## Cluster persons speaking by occurence 1K to 2k times
  dfmat_speaking_sim <- dfmat_training %>%
    dfm(remove = stopwords('en'))
  dfmat_speaking_simlty <- dfm_group(dfmat_training,
                           groups = c('personSpeakingAffiliation','personSpeaking'))
  dfmat_speaking_simlty <- dfmat_speaking_simlty %>%
    dfm_select(min_nchar = 4) %>%  # minimum characters in term
    dfm_trim(min_termfreq = 1000) %>% #mimimum times the term occurs accross all documents
    dfm_trim(max_termfreq = 2000)  # maximum times the term occurs accross all documents
  dfmat_speaking_simlty <- dfmat_speaking_simlty[ntoken(dfmat_speaking_simlty) > 1999,] # min tkns
  tstat_dist <- as.dist(textstat_dist(dfmat_speaking_simlty))
  speaking_clust <- hclust(tstat_dist)
  plot(speaking_clust, cex=.75 )
```



Terms By House Leaders

Another application of the document feature matrix and the docvar personSpeakingAffiliation is quantifying the terms used by the House leaders in the various debates. Checking the wiki the Government House Leaders are documented as Dominic LeBlanc and Bardish Chagger. The Opposition House Leaders are documented as Andrew Scheer and Candice Bergen. To isolate the debates where these individuals are recorded as the person speaking the corpus is subset by personSpeaking. With a new corpus of only the House leaders a document feature matrix is created including the docvars providing the personSpeaking. The scheme argument applied to dfm_weight() is propmax. This scheme results in the proportion of features counts of the highest feature count in a document. Mathematically this expressed as $tf_{ii}/\max_i tf_{ii}$

```
# Calculate relative frequency by ranking members
    corp_debates_leaders <- corpus_subset(corp_training, personSpeaking == c("Hon. Dominic LeBlanc", "H
    dfm_debates_leaders <- corp_debates_leaders %>% dfm(remove = stopwords('en'),
        remove_punct = T, remove_symbols = T, remove_numbers = T, include_docvars = T)
    dfm_debates_leaders_prop <- dfm_debates_leaders %>% dfm_weight(scheme = "propmax")
    freq_weight_leaders <- textstat_frequency(dfm_debates_leaders_prop,
                                             n = 15, groups = c("personSpeakingAffiliation", "personSpeaking"))
    ggplot(data = freq_weight_leaders, aes(x = nrow(freq_weight_leaders):1, y = frequency)) +
      geom_point() + facet_wrap(~ group, scales = "free") + coord_flip() +
      scale_x_continuous(breaks = nrow(freq_weight_leaders):1,
                         labels = freq_weight_leaders$feature) +
      labs(x = NULL, y = "Relative frequency")+ theme_few() +
      ggtitle("Occurrences of Terms in Debates by Leader\n42nd Parliament of Canada")
```



Key Words In Context

Proportionally there are terms that are common across all 4 House leaders in the 42nd Parliament. The kwic (key words in context) function will present the term with words before (pre token) and words after (post token). Including words before and after the token provides context of the terms use. For example the term government is frequently used by each of the leaders. Including the 5 words before and after the term gives context to how the person speaking applied the term at a specific point in a debate.

```r
## Key Words in Context

# update docnames to name of speaker for convenience
    docnames(corp_debates_leaders) <- str_sub(corp_debates_leaders$personSpeaking,
                                                    start = 6)

    toks_debates_leaders <- tokens(corp_debates_leaders,
                                  what = "word",
                                  include_docvars = T)

# key words in context
    kwic_debates_leaders <- as.data.frame(kwic(toks_debates_leaders,
                                pattern = c("government"),
                                window = 5))

  kwic_debates_leaders %>%
    mutate(member = str_sub(docname, start=1, end=10))  %>%
    group_by(member) %>%
    slice(tail(row_number(),3))%>%
    select(member, pre, keyword, post) %>%
    kable(caption = "kwic for term 'government'", booktabs = T) %>%
    kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                  full_width = F, font_size = 10, latex_options = "hold_position")
```

Table 5: kwic for term 'government'

member	pre	keyword	post
Andrew Sch	every possible solution the federal	government	could propose to end this
Andrew Sch	project , as the Liberal	government	continues to add new hurdles
Andrew Sch	, I asked if the	government	had secured a supplier to
Bardish Ch	position to carry out official	government	duties . The Prime Minister
Bardish Ch	, whether on personal or	government	business . The Prime Minister
Bardish Ch	of the time provided for	Government	Orders on the day allotted
Candice Be	justice minister tell us what	government	business she discussed with lawyers
Candice Be	answer from a so-called feminist	government	. Why could that minister
Candice Be	another question for the Liberal	government	. We know that Mastercard
Dominic Le	the	government	is proud of enhancing the
Dominic Le	care must be taken with	government	spending and the management of
Dominic Le	funds , and that this	government	will not allow anyone ,

Classification Of Documents

Classifier Training

The Naive Bayes classifier classifies based on probabilities of events. As the debate corpus has an associated docvar() of personSpeakingAffiliation (i.e. the political party they are a member) the function can be used test if the debate texts can classified by Party Affiliation. The algorithm for the classification test will be accessed via Quanteda's textmodel_nb(). The results of the classifier are then used in the creation of a Confusion Matrix that will lend it self to plotting a heat map of results of the classification test.

The results of the classification are summarized as follows and reports the classification success on the training set is about 62%.

Table 6: Training Document Classification by Party'

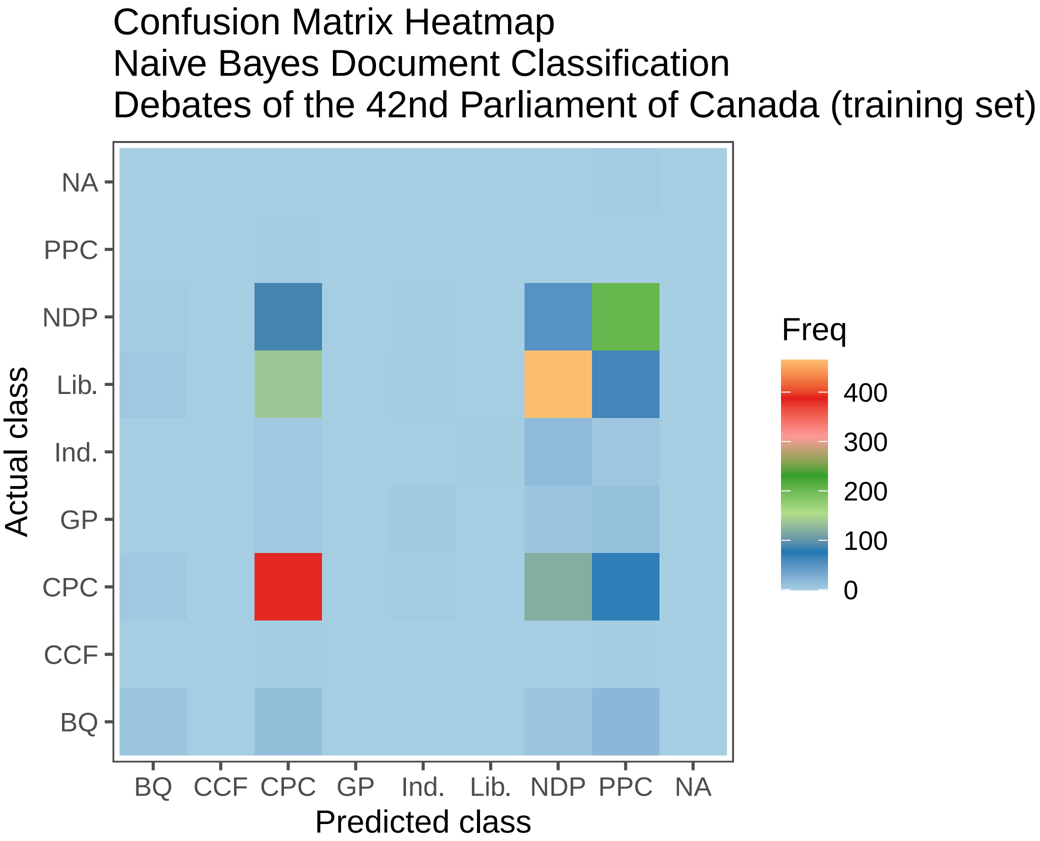
	BQ	CCF	CPC	FD	GP	Ind.	Lib.	NDP	PPC
BQ	8	0	13	0	0	0	7	19	0
CCF	0	0	2	0	0	0	0	1	0
CPC	4	0	383	0	2	0	120	72	0
FD	0	0	0	0	0	0	0	2	0
GP	0	0	4	0	3	0	7	12	0
Ind.	0	0	4	0	0	1	16	6	0
Lib.	4	0	138	0	2	0	465	64	0
NDP	2	0	87	0	1	0	53	205	0
PPC	0	0	1	0	0	0	0	0	0

Table 7: Training Results'

	x
Accuracy	0.624
Kappa	0.442
AccuracyLower	0.600
AccuracyUpper	0.647
AccuracyNull	0.391
AccuracyPValue	0.000
McnemarPValue	NaN

The table reveals the classifier matched with greater frequency the texts of debates from the major parties and to a much lesser degree the texts of the minor parties. An interesting result is the PPC result who is a single member in the House who was a ranking member within the CPC prior to creating the PPC. We see the texts from this single individual all classified as CPC. Also of interest are the results from the BQ who are members from a party exclusively from the Province of Quebec that will often align with the major parties on topics of mutual interest.

An alternative presentation of the results table is a heat map. It is clearly evident where the successes in classifying occurred as distinct colours where the predicted class intersects with the actual class from the document feature matrix applied in the Naive Bayes test.



Classifier Validation

With an accuracy of 85.5% the validation of the classifier with the test data provides a improved result over the training set's accuracy of 62.4%.

Table 8: Training Document Classification by Party'

	BQ	CCF	CPC	FD	GP	Ind.	Lib.	NDP	PPC
BQ	17	0	12	0	0	0	7	11	0
CCF	0	0	2	0	0	0	0	1	0
CPC	0	0	520	0	0	0	47	14	0
FD	0	0	1	0	0	0	0	1	0
GP	0	0	5	0	9	0	7	5	0
Ind.	0	0	10	0	0	2	12	3	0
Lib.	1	0	44	0	0	0	610	18	0
NDP	0	0	20	0	0	0	26	302	0
PPC	0	0	1	0	0	0	0	0	0

Table 9: Training Results'

	x
Accuracy	0.855
Kappa	0.784
AccuracyLower	0.837
AccuracyUpper	0.871
AccuracyNull	0.415
AccuracyPValue	0.000
McnemarPValue	NaN

Sentiment Analysis

Is a text analysis of debates complete without a sentiment analysis of the texts analyzed? Is there a tendacy to be overly positive or negative in the course of debating in the House? The table of sentiment analysis results highlights positive attributes are measured more frequently than negative ones in the words used in the debates over the course of the 42nd Parliament in Canada's House of Commons.

```
# Tokens
    toks_training <- tokens(corp_training,
                            remove_punct = T,
                            remove_symbols = T,
                            remove_numbers = T,
                            include_docvars = T)
    toks_training_lsd <- tokens_lookup(toks_training,
                                       dictionary =  data_dictionary_LSD2015[1:2])
 dfmat_training_ptystmnt <-  dfm(toks_training_lsd, tolower = T, groups = "personSpeakingAffiliation")

 senment_prty <- dfmat_training_ptystmnt %>%
   convert(to = "data.frame") %>%
   group_by(document) %>%
   mutate(pos = sum(positive /(negative + positive)))

 dfmat_training_prsnstmnt <-  dfm(toks_training_lsd, tolower = T, groups = "personSpeakingAffiliation")

 head(dfmat_training_prsnstmnt)
```

```
## Document-feature matrix of: 6 documents, 2 features (0.0% sparse) and 1 docvar.
##       features
## docs   negative positive
##   BQ       2740     4287
##   CCF       299      529
##   CPC     49265    78853
##   FD          8       33
##   GP       2028     2699
##   Ind.     2094     4645
```

Results

Debates are, by nature a series words to make, defend and challenge ideas and thoughts. Analysis of the text of the debates of 42nd Parliament of Canada reveals the words used in the course of debate are often used in related contexts regardless of party affiliation of the debater.

Demonstrated in hierarchical clustering, there are patterns of closeness within party affiliation and the words used during debates. It is also observed in the confusion matrix the affiliation of major parties of the member speaking in the debate can be classified to a certain extent.

Finally the analysis demonstrates even less frequent participants in debates from minor parties have quantifiable closeness in the terms they use with the major parties.

Conclusion

This report demonstrates text analytics can be applied in areas as diverse as debates in a political arena as dynamic as Canada's House of Commons. Quantificiation of the features, tokens and corpus extract interesting insights into the words used in the course of debating the progression of the Canada's laws and public policy.

This report should not be considered a determining insight into the work of the elected officials participating in the debates. The House of Commons is just one area faciliatating the development and implementation of policy. Much of government policy development is done in committee where members have focused opportunity to apply their wants, wishes and needs in formatting policy.

Analysis of committee proceedings would be a next logical step in assessing the effectivness of text analysis in quantifying debate and spoken words as influencers to the development of public policy.

Appendix

The following is included as supporting detail in how the data from ourparliament.ca was collected. On average the script takes about 15 seconds to load, read, parse and write the data from the webpage to the tibble. Steps are included in the script to provide periods of rest for the server so as to not overwhelm the server with page requests. The collection of 150,000 rows of data required, roughly, 28 hours to complete. The script can be augmented to select different ranges of data by adjusting the url as highlighted in the report section “Isolate Records for 42nd Parliament”.

```
### Data Collection Script, Choose Your Own

library(exploratory)
library(dplyr)
library(stringr)
library(readr)

# Components to build the series of URLs that will be needed to load the individual pages from where th
  url.mncmpt <- "https://www.ourcommons.ca/PublicationSearch/en/?View=L&Item=&ParlSes=All&oob=&Topic=&P
  url.pubtype <- "&PubType=37"

# Itterate through url.mncmpt and url.pubtype to build the list of URLs that will provide the data to b

# empty container to store the links that will be created
  links <- NULL

# Loop through a range of numbers to use as page numbers in the URL
  for (i in 1:5000){
    links <- rbind(links,paste0(url.mncmpt,i,url.pubtype))
    }

# backup the links table should we need to use the orignial as some point in the script
  links.backup <- links

# Intialize a tibble to store the information that will be collected from ourparliament.ca
  debates <- tibble()

# record when the data collection started
  starttime <- Sys.time()

# A loop to use as the range of rows from links that will be scrapped for data
for (d in 1:5) # update d to identify range of records to read from 'links'
  {
  # Page number of URL, helpful if the script terminates to identify where we ended
    urlPageNumber <- d

    print(paste("reading content from url at row",d,"from links table"), quote = F)
    print(paste(nrow(links)-d,"rows to go until done"), quote = F)

    pagecontent <- read_html(links[d])
    Sys.sleep(4.5) # give R time to read the full web page before extracting data

    print(paste("writing 30 results from row",d, "to debates tibble"))

  # ID of Hansard publication
```

```
    hansardID <-
     pagecontent %>%
      html_nodes(xpath = '//*[contains(concat( " ", @class, " " ), concat( " ", "simple-modal-video", "
      html_attr("data-desc")

    Sys.sleep(.5)

  # Person speaking during debate
      personSpeaking <-
        pagecontent %>%
        html_nodes(".PersonSpeakingName") %>%
        html_text(trim = T)

    Sys.sleep(.5)  # a built in rest period to catch up

   # DateTime of person speaking
      timePersonSpeaking <-
        pagecontent %>%
        html_nodes("[class='pi-time']") %>%
        html_text(trim = T)

    Sys.sleep(.5)  # a built in rest period to catch up

   # Provinice and political affiliation of person speaking
      affiliation <-
        pagecontent %>%
        html_nodes("[class='pi-content']") %>%
        html_text(trim = T)

    Sys.sleep(.5)  # a built in rest period to catch up

   # Text of the person speaking
    speakingContent <-
      pagecontent %>%
      html_nodes("[class='pi-text']") %>%
      html_text(trim = T)

    Sys.sleep(.5)  # a built in rest period to catch up

  # Add content collected as new row to tibble 'Debates'
    debates3 <- rbind(debates3, tibble(
      urlPageNumber = urlPageNumber,
      hansardID = hansardID,
      personSpeaking = personSpeaking,
      timepersonspeaking = timePersonSpeaking,
      affiliation = affiliation,
      speakingcontent = speakingContent
    ))
# clear the console for a neat tidy work progress area
    cat("\014")

# Be kind to the server, adding in some randomness to when loading the pages for scraping
    Sys.sleep(sample(seq(2,5,0.5), 1))
```

```r
# time the current row was written to the debates tibble
    endtime <- Sys.time()

# calculate the difference between when the loop started and when the most recent records were written
    elapsetime <- (endtime-starttime)

# elapsetime is the processing time so far.  Print how long the script has been running to the console
    print(elapsetime)

    }

# Write the records collected to a csv file for archiving.
  write_csv(debates,"debatestemp.csv")

  debates <- tmp %>%
    mutate(speakingcontent = str_clean(speakingcontent)) %>%
    mutate(speakingcontent = str_replace_all(speakingcontent,".More","")) %>%
    mutate(speakingcontent = str_replace_all(speakingcontent,"Mr. Speaker, ","")) %>%
    mutate(speakingcontent = str_replace_all(speakingcontent,"Madam Speaker, ","")) %>%
    mutate(speakingContent = str_replace_all(speakingcontent,".LessCollapse","")) %>%
    mutate(hansardDate = str_sub(hansardID,start = 15, end = 24)) %>%
    mutate(hansardId = trimws(str_sub(hansardID, start = 1, end = 13))) %>%
    mutate(personSpeakingRiding = sub(".*\\((.*)\\).*", "\\1", .$personSpeaking, perl=TRUE)) %>% # sub(
    mutate(personSpeaking = sub("\\ \\(.*", "", .$personSpeaking)) %>%
    mutate(personSpeakingDate = str_sub(timepersonspeaking, start = 1, end = 10)) %>%
    mutate(personSpeakingTime = str_sub(timepersonspeaking, start = 12, end = 16)) %>%
    mutate(personSpeakingJurisdiction = sub(".*\\((.*)\\).*", "\\1", .$affiliation, perl=TRUE)) %>%
    mutate(personSpeakingJurisdiction = str_sub(personSpeakingJurisdiction, start = 1, end = 2)) %>%
    mutate(personSpeakingAffiliation = sub("\\ \\(.*", "", .$affiliation)) %>%
    select(-c("urlPageNumber","affiliation", "hansardID", "timepersonspeaking","speakingcontent"))
    debates <- debates[c(4,3,1,8,5,6,7,9,2)] # arrange columns of tibble to a logical order

head(debates, 3) # have a look at the mutations & inspect the data format
write_csv(debates, "parlamentCanadaDebates.csv") # archive the tibble to .csv

debates %>%
  group_by(personSpeakingAffiliation) %>%
  count() # see there are some anomolies


test <- debates %>%
  filter(str_detect(personSpeakingAffiliation, c("Charles", "Marc"))) # election of speaker




write_delim(debates,"CYO.parliamentCanadaDebates", delim = "|")
CYO.d42pc <- sample_frac(debates, .15)
write_delim(CYO.d42pc,"CYO.d42pc.txt", delim = "|")
```