Topic - Analyzing patterns of English words used over time

**Abstract**

A considerable amount of historical data is now accessible as raw, digitized text. Familiar examples consists of – letters, personal notes, newspapers etc. There are many softwares that provide rich text analyzing tools for performing text mining and to analyze the trends. This project deals with analyzing the patterns of words used over time and also to find the association between frequently used terms(words). In this project we have made use of **R studio** for performing our intended work. R is a programming language used for statistical computation. It is freely available over the internet. The R language is greatly used among data miners and statisticians. The methodology used here involves using a tokenizer that parses text into elements such as words. We have gathered our data set from combining various State of the Union speeches.

**Keywords:** State of the Union speeches, tm package, tidyverse, frequent term associations, R language.

**Introduction**

The work embodied in this project consists of analyzing the patterns of words used over time and also finding frequent terms associations. The corresponding data set is gathered from the speeches given various Prime ministers of America. All of these speeches are kept inside a folder and from there it is loaded into R as a corpus for mining purposes. We have used *tidy verse package* and *tm package* of R language to perform this task. The first provides convenient tools for reading in and working with data sets, and the second contains the functions that allow us to split text data into words and sentences. The following sections will provide step by step implementation detail for above mentioned project.

**Data Selection**

We have gathered the State of the Union speech data from Kaggle repository. We are selecting from the year 1790 till 2018.We will use this data set to perform mining task and also find frequent term associations.

<https://www.kaggle.com/rtatman/state-of-the-union-corpus-1989-2017>

# Data Preprocessing

# For data to be suitable for mining purpose, it should be cleaned. We will first remove the various punctuation marks, white spaces, any numbers present. Remove common stop words like – “we”,” are”, “you”, “and” etc.

# Data Transformation

# Convert the data set into a particular case – Upper or lower (preferred). Stem the data set, meaning bring each of the words to their root form.

**Code:-**

# Package set up





# Loading the corpus

> input <- "E:/VIT/R projects/sotu\_text"

Once this is achieved, the following block of code can be used read in all of the texts:

> files <- dir(input, full.names = TRUE)

> text <- c()

> for (f in files) {

+ text <- c(text, paste(readLines(f), collapse = "\n")) + }

The for-loop variable ‘f’ iterates over all the text files in the *sotu\_text* folder and consequently reads all the texts.

Now we shall clean the data. By this we mean, we will remove whitespaces, convert all the text into upper or lower case, remove any numbers present in the data set, remove common stopwords like – “we”,”are”,”you” etc., remove any punctuations and lastly stem the texts present. This will reduce the words to their root form. For performing text stemming we require *SnowballC* package which is already loaded.

> # convert the text to lower case

> docs <- tm\_map(docs, content\_transformer(tolower))

> # remove numbers

> docs <- tm\_map(docs, removeNumbers)

> # Remove english common stopwords

> docs <- tm\_map(docs, removeWords, stopwords("english"))

> # remove punctuations

> docs <- tm\_map(docs, removePunctuation)

> # Text stemming

> docs <- tm\_map(docs, stemDocument)

# Exploratory Analysis

Now we will *tokenize\_words()* function of the *tokenizers* package. This will give us the length of each speech in total no. of words.

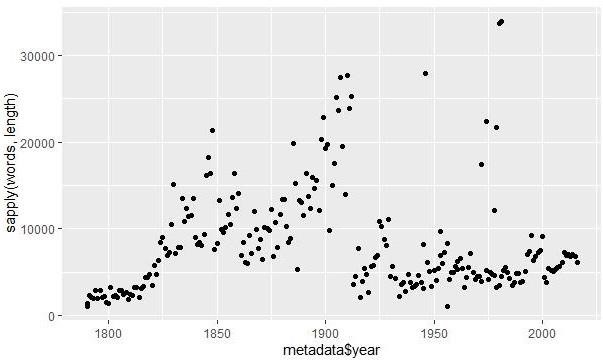
> words <- tokenize\_words(text)

As a first step in processing this text, we will use the *tokenize\_words()* function from the tokenizers package to split the text into individual words. *tokenize\_words()* returns a list object with one entry per document in the input. The *sapply()* function applies its second argument to every element of its first argument. As a result, we can calculate the length of every sentence in the paragraph with one line of code:

> sapply(words,length)

Now to measure the length of each of the speeches of various ministers we can draw a **scatter plot.** We can construct the same using the built in *qplot()* function in R by putting the year on the x-axis and the length in words on the y-axis.

> qplot(metadata$year, sapply(words, length))



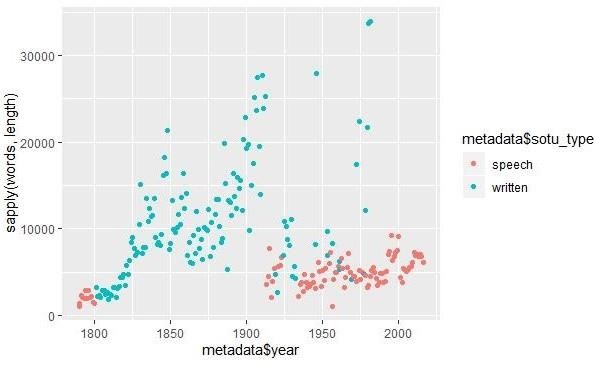
It seems that for the most part addresses steadily increased in length from 1790 to around 1850, and then increase again until the end of the 19th century. The length dramatically decreased around World War I, with a handful of fairly large outliers scattered throughout the 20th century.

Setting the color of the points to denote whether a speech is written or delivered orally explains a large part of the variation.

> qplot(metadata$year, sapply(words, length),

+ color = metadata$sotu\_type)

The above code yields the following plot:-



In this plot we can easily see whether the speeches were delivered orally or it was written.

We see that the rise in the 19th century occurred when the addresses switched to written documents, and the dramatic drop comes when Woodrow Wilson broke tradition and gave his State of the Union as a speech in Congress. The outliers we saw previously were all written addresses given after the end of World War II .

# Stylometric Analysis

Stylometry, the study of linguistic style, makes extensive use of computational methods to describe the style of an author’s writing. With our corpus, it is possible to detect changes in writing style over the course of the 19th and 20th centuries. A more formal stylometric analysis would usually entail the application of part of speech codes or complex, dimensionality reduction algorithms such as principal component analysis to study patterns over time of across authors.

The tokenizer package also supplies the function *tokenize\_sentences()* that splits a text into sentences rather than words.

The corpus can be split into sentences using the same function. In this case the result is a list with 236 items in it, each representing a specific document.

> sentences <- tokenize\_sentences(text)

Next, we want to split each of these sentences into words. The *tokenize\_words()* may be used, but not directly on the list object *sentences*. It would be possible to do this with a for loop again, but there is an easier way. The *sapply()* function provides a more straightforward approach. Here, we want to apply the word tokenizer individually to each document, and so this function works perfectly.

> sentence\_words <- sapply(sentences, tokenize\_words)

We now have a list (with each element representing a document) of lists (with each element representing the words in a given sentence). The output we need is a list object giving the length of each sentence in a given document.

> sentence\_length <- list()

> for (i in 1:nrow(metadata)) {

+ sentence\_length[[i]] <- sapply(sentence\_words[[i]], length) + }

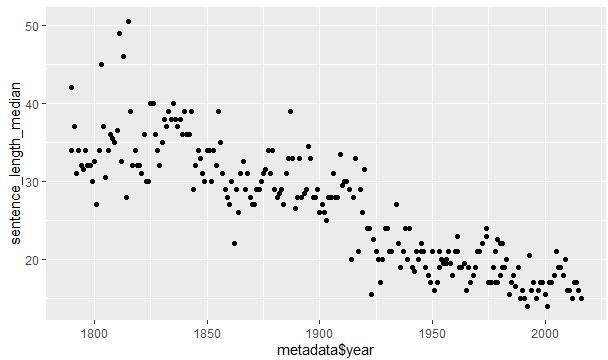
*sentence\_length* is a list object which will contain length of each sentence in a given document.

The output *sentence\_length* may be visualized over time. We first need to summarize all of the sentence lengths within a document to a single number. The **median** function, which finds the 50th percentile of its inputs, is a good choice for summarizing these as it will not be overly effected by a parsing error that may mistakenly create an artificially long sentence.

> sentence\_length\_median <- sapply(sentence\_length, median)

We now plot this variable against the speech year using, once again, the *qplo()* function. This will fetch us the median length of the various speeches.

> qplot(metadata$year, sentence\_length\_median)



The plot shows a strong general trend in shorter sentences over the two centuries of our corpus. Recall that a few addresses in the later half of the 20th century were long, written addresses much like those of the 19th century. It is particularly interesting that these do not show up in terms of the median sentence length. This points out at least one way in which the State of the Union addresses have been changed and adapted over time.

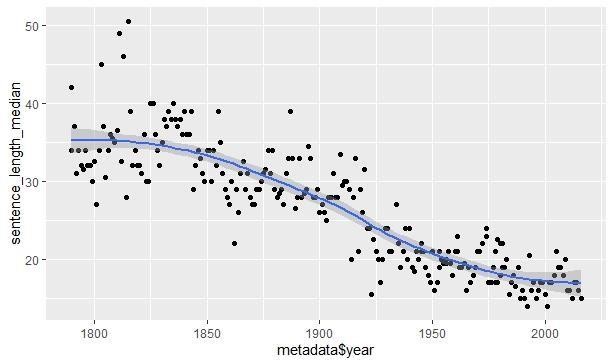
To make the pattern even more explicit, it is possible to add a smoothing line over the plot with the function *geom\_smooth().*

> qplot(metadata$year, sentence\_length\_median) +

+ geom\_smooth()

`geom\_smooth()` using method = 'loess' and formula 'y ~ x'

Smoothing lines are a great addition to many plots. They have a dual purpose of picking out the general



trend of time series data, while also highlighting any outlying data points.

# Most frequent word used

We can build the ***TermDocumentMatrix.*** This will give us the word and corresponding its frequency, i.e. , the number of times that word has appeared.

> dtm <- TermDocumentMatrix(docs)

> m <- as.matrix(dtm)

> v <- sort(rowSums(m),decreasing=TRUE)

> d <- data.frame(word = names(v),freq=v)

> head(d, 10)

|  |  |  |
| --- | --- | --- |
|  | word | freq |
| will | will | 11221 |
| state | state | 9382 |
| govern | govern | 8643 |
| year | year | 7365 |
| nation | nation | 6761 |
| congress | congress | 5769 |
| unit | unit | 5325 |
| can | can | 4768 |
| countri | countri | 4737 |
| peopl | peopl |  |

Now we can will generate the **word cloud** which is a visual representation of textual data.

wordcloud(words = d$word, freq = d$freq, min.freq = 1,

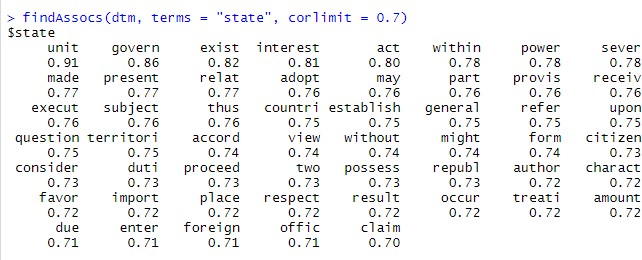
+ max.words = 200, random.order = FALSE, rot.per = 0.35,

+ colors = brewer.pal(8, “Dark2”))



The above word cloud shows that the words - “will”,”govern” and “state” more frequently used Hence their text size is bigger than others.

Now to find the associations between frequent words we can perform correlation using *findAssocs()* function.



# Conclusion

In this project we analyzed the pattern of words used over time and also found out associations between most frequent terms. The data, as mentioned, was collected from speeches delivered by various Prime ministers of America during their tenure right from 1790 to 2019 from the Kaggle repository.