Simple Text Analytics: Wordcloud Demonstration

Anthony Castellani

This is a demonstration of a simple form of text analysis: the word cloud. This demonstration will require the use of R, as well as the R packages `dplyr`, `tm`, and `wordcloud`. Using these tools, there are actually three different forms of word cloud that will be demonstrated.

The first form is a basic word cloud that displays the words used in a segment of text such that the font size of any given word is proportional to the frequency of its usage (i.e., the more a word is used, the larger it will appear).

The second form is a comparison cloud. In this form, two or more different blocks of text are compared; text size is proportional in the same way as the basic format of word cloud. In this format, however, color is used to differentiate the words favored in the different blocks of text. Where a given word is used in more than one block of text in the comparison, the block of text where the frequency of that word is greatest will be the one to display that word in that block's color.

The third form is a commonality cloud. In this form, a single word cloud is displayed, but only those words that are common across all compared blocks of text are used.

Important note:

For the purposes of this demonstration, I have used the text of the acceptance speeches given by the Democratic (Hillary Clinton) and GOP (Donald Trump) Presidential Nominees at their respective conventions in 2016. These texts were used because they are timely and present an interesting opportunity to use exactly the tools that I am demonstrating here. I will not be making any political critiques of these candidates or their speeches; I present this in an entirely politically neutral context. I am trying to demonstrate a skill set, not a political preference. This demonstration should not be understood in any way to indicate support for any candidate.

Load the packages

```
library(dplyr)
library(tm)
library(wordcloud)
```

Note: I have separately loaded into R the two speeches. They are too lengthy to display here. In the following blocks of code, the speech delivered by Hillary Clinton is represented by the object `cs`, while the speech delivered by Donald Trump is represented by the object `ts`. They will need to be read in.

```
cs <- readRDS(file = "cs.rds")
ts <- readRDS(file = "ts.rds")
```

The two initial objects, `cs` and `ts`, can in fact be converted directly to a word cloud with no further modification by simply calling the character vector with the function `wordcloud()`. This will not provide the best outcome, however, as `cs` and `ts` both still contain all of the various letter cases and punctuation and numbers and whatnot that they had in the original speeches. In order to get the most out of the function, some modifications must be made. The first modification is a two-step process to convert the character vector into a corpus:

```
cs.vs <- VectorSource(cs)
cs.corp <- Corpus(cs.vs)

ts.vs <- VectorSource(ts)
ts.corp <- Corpus(ts.vs)
```

Next, several calls to `tm_map` will remove punctuation, transform all upper case letters to lower case, and remove numbers.

```
cs.corp <- tm_map(cs.corp, removePunctuation)
cs.corp <- tm_map(cs.corp, content_transformer(tolower))
cs.corp <- tm_map(cs.corp, removeNumbers)

ts.corp <- tm_map(ts.corp, removePunctuation)
ts.corp <- tm_map(ts.corp, content_transformer(tolower))
ts.corp <- tm_map(ts.corp, removeNumbers)
```

The objects are almost ready. The last cleaning step requires the removal of stop words, which are those small, common words that don't add much value to the final product (e.g., "all", "so", "for", etc.).

```
cs.corp <- tm_map(cs.corp, function(x)removeWords(x,stopwords()))
ts.corp <- tm_map(ts.corp, function(x)removeWords(x,stopwords()))
```

The corpus objects are now ready for display as word clouds.

Here is Hillary Clinton's word cloud (blue because of Clinton's association with the Democratic Party):

```
wordcloud(cs.corp, min.freq = 2, colors = "blue")
```

Here is Donald Trump's word cloud (red because of Trump's association with the GOP):

```
wordcloud(ts.corp, min.freq = 2, colors = "red")
```

In order to compare and contrast the two speeches in word cloud format, the two corpus objects must be transformed into different formats.

First, the two objects will be transformed into Term Document Matrices:

```
cs.tdm <- TermDocumentMatrix(cs.corp)
ts.tdm <- TermDocumentMatrix(ts.corp)
```

Next, the two TDMs will be transformed into data frames, where their row and column names will receive some modification:

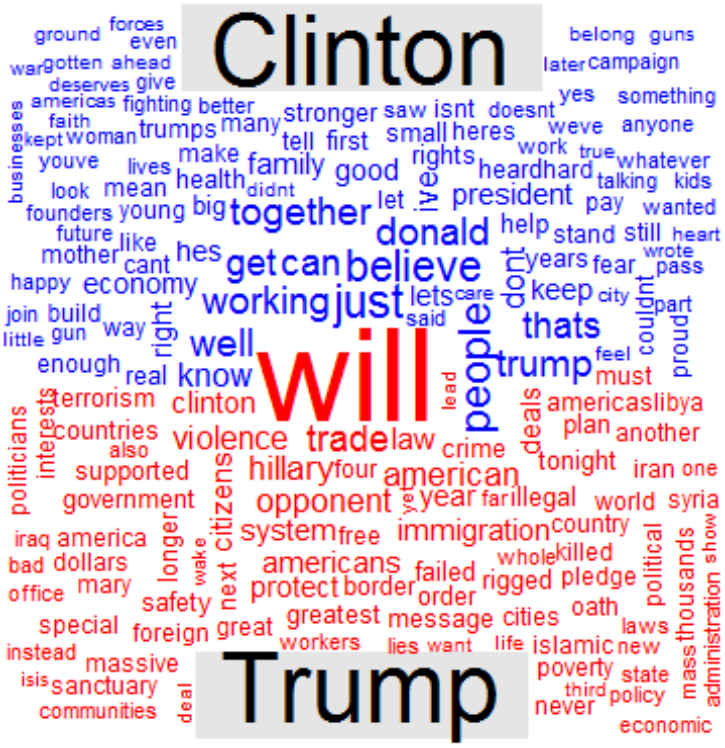
```
cs.mx <- as.data.frame(as.matrix(cs.tdm))
cs.mx$Terms <- rownames(cs.mx)
colnames(cs.mx) <- c("Clinton", "Terms")

ts.mx <- as.data.frame(as.matrix(ts.tdm))
ts.mx$Terms <- rownames(ts.mx)
colnames(ts.mx) <- c("Trump", "Terms")
```

The final step before plotting is to join the two data frames into one. `dplyr` will be used here:

```
cs.ts <- full_join(cs.mx, ts.mx, by = "Terms")
rownames(cs.ts) <- cs.ts$Terms
cs.ts$Terms <- NULL
cs.ts[is.na(cs.ts)] <- 0
```

The comparison cloud is now ready for display. Note that among all words used in the two speeches, those words that were used by Clinton more than were used by Trump show up on the blue side, while those words that were used by Trump more than were used by Clinton show up on the red side.

```
comparison.cloud(cs.ts, colors = c("blue", "red"))
```



Display of the commonality cloud requires no further modification. Note that this function produces a single word cloud. This single word cloud represents only those words that were used in both speeches.

```
commonality.cloud(cs.ts)
```

This concludes the demonstration of the use of a simple text analytics tool to compare and contrast two bodies of text.