Northeastern University

ASSIGNMENT FRONT SHEET

Course Name: **ALY6040 Data Mining Applications**

Professor Name: **Nagadeepa Shanmuganathan**

Student Name: **Dong Quoc Tuong (Lukas)**

Student Class: **Fall 2019 CPS** **Term: Winter 2021**

Module 1: Introduction to tm Package in R

Completion Date: Jan 24th **Due Time:12:00am**

Statement of Authorship

I confirm that this work is my own. Additionally, I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person e.g. from a book, handout, another student. I am aware that it is a breach of Northeastern University's regulations to copy the work of another without clear acknowledgement and that attempting to do so renders me liable to disciplinary procedures. To this effect, I have uploaded my work onto Turnitin and have ensured that I have made any relevant corrections to my work prior to submission.

☒ **Tick here** to confirm that your paper version is identical to the version submitted through Turnitin

In this paper, we will analyze the Trump's twitter in order to understand more about the person. To start with, we will install the necessary packages, download the files and set the right directory for the system.

Start the analysis:

We create an empty corpus with VCorpus. Text corpus is defined as a language resource consisting of a large and structured set of texts to perform statistical analysis, hypothesis testing, occurrences, checking or validating linguistic rules ("NLP - Linguistic Resources," 2018)

```
                                                Length Class             Mode
Trump Black History Month Speech.txt            2      PlainTextDocument list
Trump CIA Speech.txt                            2      PlainTextDocument list
Trump Congressional Address.txt                 2      PlainTextDocument list
Trump CPAC Speech.txt                           2      PlainTextDocument list
Trump Florida Rally 2-18-17.txt                 2      PlainTextDocument list
Trump Immigration Speech 8-31-16.txt            2      PlainTextDocument list
Trump Inauguration Speech.txt                   2      PlainTextDocument list
Trump National Prayer Breakfast.txt             2      PlainTextDocument list
Trump Nomination Speech.txt                     2      PlainTextDocument list
Trump Police Chiefs Speech.txt                  2      PlainTextDocument list
Trump Response to Healthcare Bill Failure.txt 2      PlainTextDocument list
```

Next we loaded the details of any documents in the corpus. Looking at the first and second document, we can see that both documents have the same Meta data at 7 but the second one contains 3 times more characters compared to the first one, 12747 and 4068 respectively

```
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 4068
```

```
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 12747
```

Preprocessing

We remove anything that hinders our analysis process. This process includes numbers, capitalization, unnecessary figures (\\, @, etc.) common words – stop words in the English language (the, a, etc.) and punctuation. However that is not enough. Since Trump tends to repeat his messages multiple times in a speech and make vague assumption, we need to also use tm_map(docs, removeWords, c("syllogism", "tautology) to eliminate any words with the same meaning (ex: “ATM machine”, because “M” stands for “machine”. Another thing to remember while analyzing someone’s speech, especially Trump is his frequent use of words that are often associate with each other to have a specific meaning. “Fake” and “news”, if separated is totally misconstrued because they are always put together for “fake news” to be established as a contemporary phenomenon

Lastly, we move on to the stemming step and clean the white spaces. Stemming is to reduce inflected words to their word stem, base or root form—generally a written word form. So “Consulting”, “Consultant”, “Consultative” turns into “consult”.

Stage your data

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. (Yangchang Zhao, 2012) The spare entries / non spare entries ratio is 3/1. The longest words have 19 characters in them. Keeping in mind Trump’s preference of sound bites and short words to complicated ones, those long words could be the combination of 2 separate words like “politically-correct” (19) that we created above.

```
> dtm
<<DocumentTermMatrix (documents: 11, terms: 3698)>>
Non-/sparse entries: 8443/32235
Sparsity           : 79%
Maximal term length: 19
Weighting          : term frequency (tf)
```

Then we transpose the matrix and organize terms by frequency. Next, we start having a look at the most and least frequently occurring words. The resulting output is two rows of numbers. The top row reflects the frequency with which words appear while the bottom row indicates how many words appears that frequently. For example, here we are examining 20 least frequent words and we can see that 1655 terms appear once, 631 words appears twice, etc.

```
freq
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
1655  631  329  214  124  107   82   60   59   45   38   33   31   25   25   16
  17   18   19   20
   8   13   16   11
```

Here are the 20 most frequent words

```
freq
 79  83  88  89  98 100 101 102 105 107 111 122 127 139 140 163 174 265 278 428
  2   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

Then we create the vocabulary table with the frequency of appearances for each word. For example, "also" appears 54 times similarly to "years"

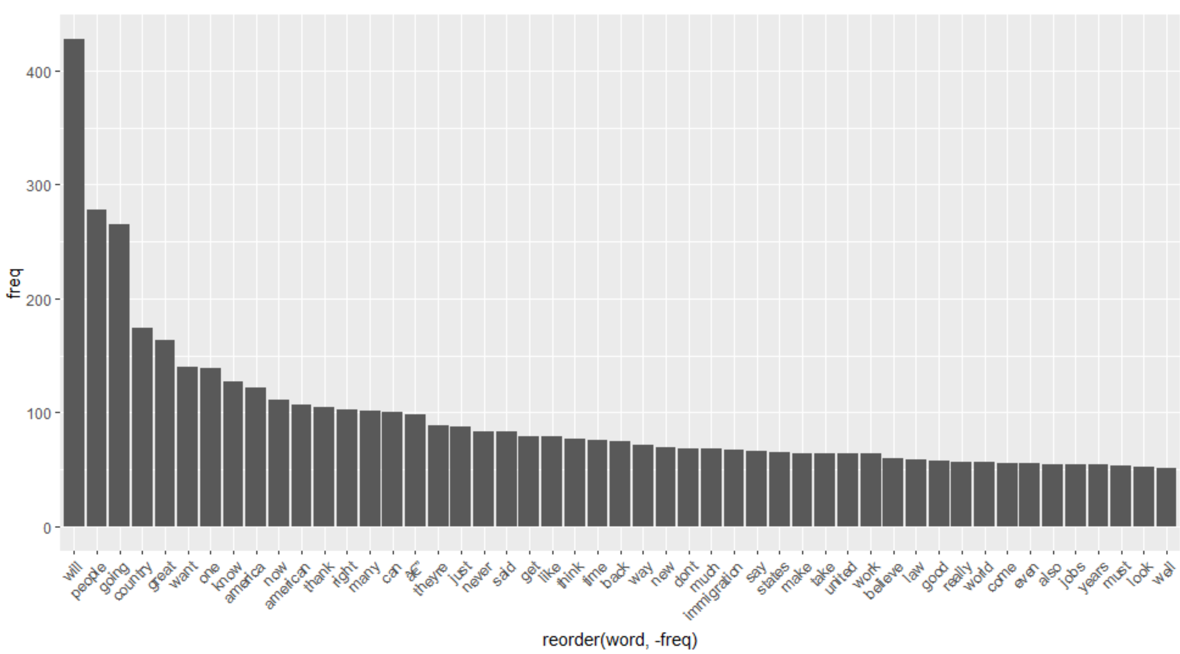
```
         also      always     america    american     another        back         bad
           54          24         122         107          22          75          35
      believe         big        came         can        care        come     country
           60          45          20         100          37          55         174
          day   different        done enforcement        even        ever       every
           36          16          24          43          55          42          49
          get     getting        give       going        good       great       group
           79          23          25         265          58         163          20
       happen         job        just        know        last         law         let
           36          38          88         127          44          59          40
         life        like      little        long        look         lot        love
           27          79          24          36          52          44          45
         made        many        much        must      nation        need       never
           32         101          68          53          48          32          83
          new         now      office         one      people   president         put
           69         111          24         139         278          44          35
       really    remember       right        safe        said         say         see
           57          27         102          35          83          66          48
         seen   something     special      states        take        tell       thank
           34          25          26          65          64          50         105
       things       think        time       today    together     totally       truly
           40          77          76          33          34          18          16
   understand      united        want         way        well        will        work
           29          64         140          71          51         428          64
        world        year       years
           56          47          54
```

Next we sort the words according to the frequency of appearances in decreasing order and create a data frame from our next step. Here is the final outcome

```
           word freq
will       will  428
people   people  278
going     going  265
country country  174
great     great  163
want       want  140
```

Plot Word Frequency , Calculate terms correlations , Create word clouds

To plot the word frequency histogram, we will use the ggplot2 package. In order for the histogram to not get too clustered or all over the place , we will make sure to only include words that appear more than 50 times in the corpus and get some alignment parameters for our code. As we can see that Trump's most favorite words are "will", "people, "going", "country", "great", "one" indicating the fact that he likes to make a lot of promises that target the common folks to make them feel united for the country or something great.



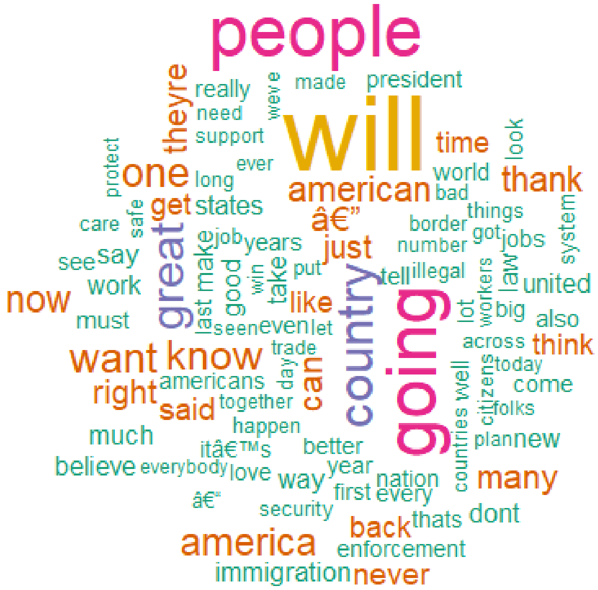
The findings leads us to a question to what words often get associated to the words “American” and ”Country” as we can see they are one of the few words Trump likes to use constantly. If the words always appear together then the correlation is 1. In this case we are specifying the correlation limit of 0.85 but we can change that in the future. Looking into this allows us to see whether or not we need to update “Combining words” section that we established above.

```
$country
  nothing    cities countries      jobs      come   biggest    donors    second 
     0.95      0.94      0.94      0.92      0.91      0.90      0.90      0.90 
    begin    border      plan    crimes     globe     meant thousands     means 
     0.88      0.88      0.88      0.87      0.87      0.87      0.87      0.86 
  workers      also   despite      take 
     0.86      0.85      0.85      0.85 

$american
 restore     task     fair   budget    cycle      new promises  dollars  finally 
    0.97     0.93     0.92     0.91     0.89     0.89     0.89     0.88     0.88 
millions national     tens  foreign   middle  justice  program    break  joining 
    0.88     0.88     0.88     0.87     0.87     0.86     0.86     0.85     0.85 
  united 
    0.85 
```

From the result, we can deduct that Trump likes to use the “country” with “nothing” as the base and then it is the “American” to “restore” the country as his answer.

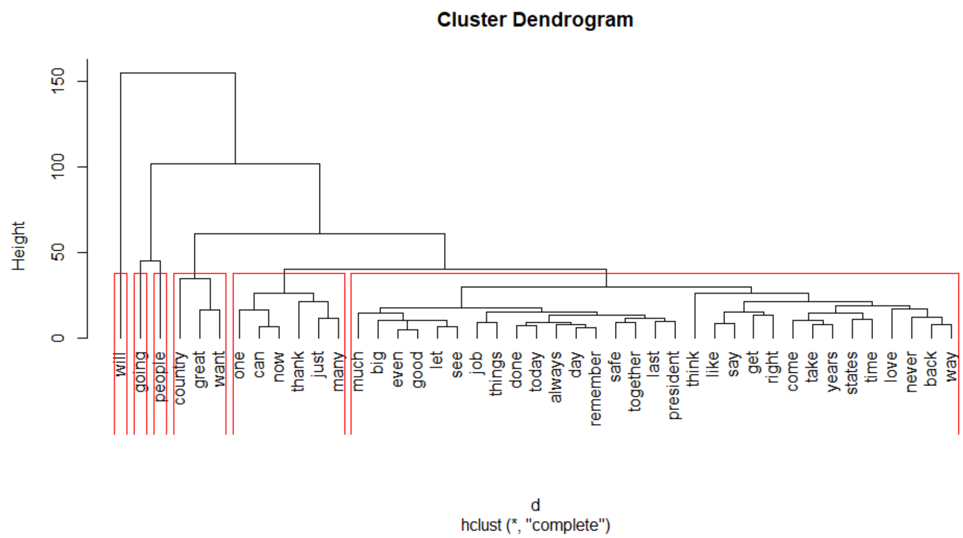
Wordcloud library was created to specifying in creating word cloud. After preparing the data max 15% and getting colored according to the frequencies. We can see that “will” is the most used words (1st in yellow) followed by “people” and “going” (2nd and 3rd in pink). Then we have “country” and “great” (4th and 5th in purple) the next top ten in orange and the rest in green. The clear and concise visualization allows the readers to digest information faster



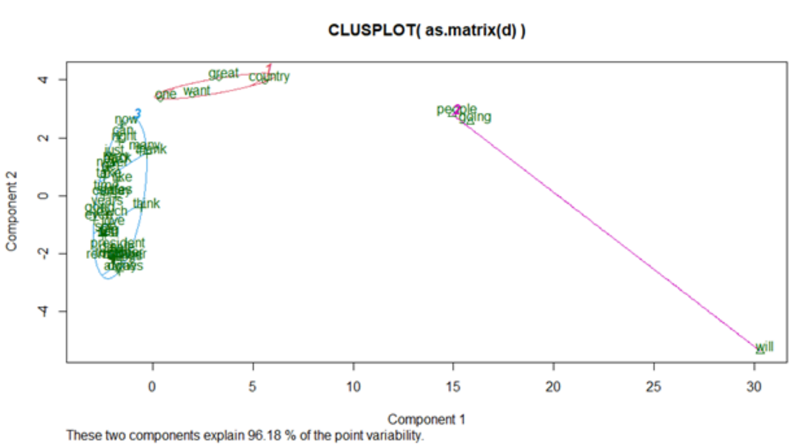
Hierarchal Clustering vs. K-means clustering

There are two main clusterisation algorithms that we can dell on: Hierarchical clustering and K-means clustering. Hierarchical clustering is the method of cluster analysis that aims to build a hierarchy of clusters regardless of having a fixed clustered beforehand. Hierarchical methods can be either divisive or agglomerative. K-means, on the other hand, uses pre-specified number of clusters (centroids) to plot the map. The number of centroids is determined by using the elbow methods. (Valentina Alto, 2019)

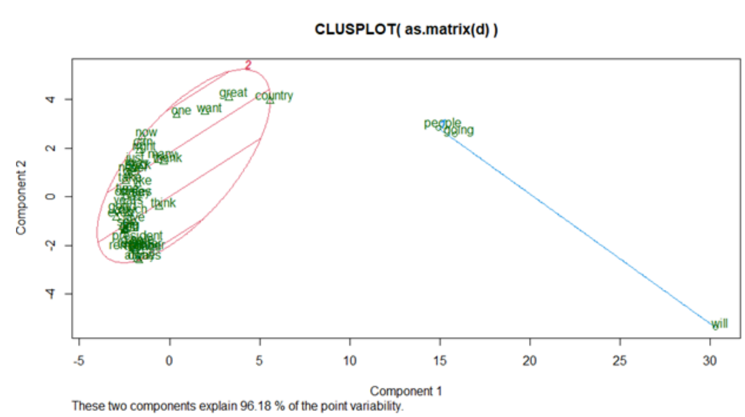
For hierarchal clustering, we make the matrix that is only 15% empty space and then introduce the “cluster” library. Then we calculate the distance between words with Euclidian method and assign the number of clusters that you are using as 6. Last but not least, we make sure to draw a dendrogram with red borders around the 5 clusters. The horizontal axis indicates the clusters whereas the vertical scale represents the distance or dissimilarity. Each joining (fusion) of two clusters is pained by the splitting of a vertical line into the vertical line. We can give an example like “like” and “right” are equally distant from “just”.



We do the same with K-means clustering where we prepare the data max 15% empty space and use the Euclidian distance. From what we have seen above, there are actually two groups of clusters due to one having a significant number of words and thus decide to assign k as 2 or 3 for this. After running the code, we can see that k=3 is the much better choice compared to 2 due to the fact that the first 2 clusters contain words that appear too frequently compared to the second cluster’s ones. “Will” seems to be the outliner, worthy of having its own cluster but it is best to group them with the other popular words that are “people“, “going”



K= 3



K= 2

References

NLP - Linguistic Resources. (2018). Retrieved January 24, 2021, from https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_linguistic_resources.htm

Valentina Alto. (2019, July 8). Unsupervised Learning: K-means vs Hierarchical Clustering. Retrieved January 24, 2021, from https://towardsdatascience.com/unsupervised-learning-k-means-vs-hierarchical-clustering-5fe2da7c9554

Yangchang Zhao. (2012). Document Matrix - an overview. Retrieved January 24, 2021, from https://www.sciencedirect.com/topics/mathematics/document-matrix