Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

**Course: ALY6040 – Data Mining Application**

**Assignment:**

Module 5 – Technique Practice

**Submitted on:**

May 14th, 2023

**Submitted to:**  **Submitted by:**

Professor: BEHZAD AHMADI NIKSHITA RANGANATHAN

**ABSTRACT**

Text mining is the method of obtaining knowledge and insightful information from unstructured text data. It involves using various techniques such as natural language processing (NLP), machine learning, and data mining to analyze and extract relevant information from large collections of text.

One of the most popular visual techniques used in text mining is the creation of word clouds. A word cloud is a graphical representation of text data that displays the most frequently occurring words in a document or a collection of documents. The size of each word in the word cloud reflects how frequently it appears in the text.

Word clouds are useful for quickly understanding the most prominent themes in a document or a corpus of documents. They can help identify important keywords and topics that may be relevant to a particular field or domain. They can also be used to compare the language used in different documents, such as analyzing the differences in language used by political candidates during an election campaign.

There are numerous uses for text mining across several industries and fields. Text mining can help organizations extract valuable insights and knowledge from unstructured textual data, make better decisions, improve processes, and gain a competitive advantage in their respective industries.

**INTRODUCTION**

**About this Dataset:**

The dataset available given for this assignment contains the transcript of Martin Luther King Jr.'s historic "I Have a Dream" speech, delivered on August 28, 1963, during the March on Washington for Jobs and Freedom. The speech is widely considered one of the most significant and influential speeches of the 20th century and played a crucial role in the civil rights movement in the United States.

It is a plain text file that contains the full transcript of the speech divided into several sections. The text is unstructured, and it contains various elements such as paragraphs, quotes, and rhetorical devices that make it a rich source of textual data for analysis. The availability of the dataset online makes it easily accessible for researchers, educators, and students around the world to study and analyze.

The dataset can be used for various natural language processing (NLP) tasks. Text mining and analysis techniques can be applied to the dataset to extract valuable insights and knowledge. Sentiment analysis, for instance, can be used to determine the speech's overall emotional tone. Topic modeling can be applied to identify the key themes and topics addressed in the speech. The most frequently occurring words and phrases in the text can also be found by using keyword extraction methods.

This is an important resource for anyone interested in studying the Civil Rights Movement and its impact on American history and society.

**CODE WALK-THROUGH**

* **Installing and loading the packages:**

The required packages for text mining, stemming, creating word clouds, and color palettes are installed and loaded. In this code, we are using several packages to perform text mining and visualization tasks.

The "tm" package provides a framework for working with text documents, while the "SnowballC" package provides functions for stemming words to their root form. The "wordcloud" package is used to create visual representations of the most frequently occurring words in the text, while the "RColorBrewer" package provides a range of color palettes for use in the visualization.

* **Loading the text data**

The text data is loaded from an external URL. The readLines() function is used to read the text file, which is then stored in the text variable.

The purpose of this step is to load the raw text data into R so that it can be transformed and analyzed further.

* **Creating a corpus**

A corpus is a collection of text documents that serve as the foundation for analysis. A corpus can be created from any type of text data, including books, articles, websites, or social media posts.

The 'tm' package is used to load the text data as a corpus. The 'Corpus()' function is used to create a corpus of documents from the text. Each document in the corpus represents a line from the original text file.

* **Inspecting the Corpus**

The inspect() function is used to display the contents of the corpus. The output shows the number of documents, their names, and the number of text elements (words) in each document.

* **Data Cleaning and Preprocessing**

This is an important step in text mining. Several transformations are applied to the text data to preprocess it for analysis.

1. The toSpace() function is defined using the content\_transformer() function which replaces special characters such as ("/" or "@") with spaces.
2. The content\_transformer() function and tolower() function convert all text data to lowercase.
3. The removeNumbers() function removes all numbers from the text data.
4. The removeWords() function removes common English stop words (words that occur frequently but do not carry much meaning, such as "the", "and", "in", etc). Additionally, any custom stopwords specified by the user are removed using the same
5. ‘removePunctuation’ function is used to remove punctuations from the text.
6. ‘stripWhitespace’ function is used to eliminate extra white spaces.
7. Text stemming is commented out, which uses the 'stemDocument()' function to perform stemming. Stemming reduces words to their root form, which can help to identify common roots and group words together.

* **Creating Document-Term Matrix**

The pre-processed text data is then converted into a term-document matrix (TDM). A TDM is a matrix where each row represents a unique term in the corpus, and each column represents a document in the corpus. The values in the matrix represent the frequency of each term in each document.

* **Converting Matrix to Data Frame**

The matrix is then converted to a data frame to allow for easier sorting and visualization.

* **Generating Word Cloud**

The wordcloud() function is used to generate a word cloud from the data frame generated in the previous step. The function takes as input the words and their corresponding frequencies and generates a plot of the most common words in the text. It displays the most frequently occurring words with larger font sizes. The colors are generated using the brewer.pal function from the RColorBrewer package. The set.seed() function is used to ensure that the plot is reproducible.

* **Finding Frequent Terms and Word Associations**

The findFreqTerms() function is used to find terms (words) that occur frequently in the text. The lowfreq parameter is used to specify the minimum frequency of the terms to be returned. In this case, terms that occur at least 4 times are returned.

The findAssocs() function is used to find terms that are associated with a given term. The terms parameter is used to specify the target term, and the corlimit parameter is used to specify the minimum correlation between the target term and the associated terms to be returned. In this case, terms that have a correlation of at least 0.3 with the term "freedom" are returned.

* **Visualizing Most Frequent Words**

Finally, we create a bar plot using the "barplot" function to display the top 10 most frequent words in the text.

**ANALYSIS**

A close up of words

Description automatically generated with low confidence

***Figure 1 – Wordcloud***

The analysis of the data reveals that the speech contains several words such as "freedom", "together", “nation”, “America” and "dream". These words reflect the main themes of the speech, which are equality, togetherness, and freedom in America. The plot also shows that the word "white" and “black” appears frequently in the text, indicating the issue of racial discrimination.

The speech uses more positive words compared to negative words, which could be one of the reasons why it had a significant impact over the years.

A picture containing text, screenshot, font, line

Description automatically generated

***Figure 2 – findAssocs() function***

We can see that the words "let" and "ring" have a strong correlation with "freedom" (correlation coefficient of 0.89 and 0.86, respectively). This indicates that the speaker has used the words "let" and "ring" in close proximity to the word "freedom" to emphasize the importance of these concepts in his speech.

A screenshot of a computer code

Description automatically generated with low confidence

***Figure 3 – Frequency of words***

From the frequency table of words, we can see that the word "will" has the highest frequency of 17 times, followed by "freedom" - 13 times. This indicates that the speech has a strong emphasis on the importance of "freedom" and the determination needed to achieve it.

A picture containing text, screenshot, diagram, font

Description automatically generated

***Figure 4 – Barplot***

The most frequent words are "will", "freedom", "ring", "day", "dream", "let", "every", "able", "one", and "together". These words are important to the text and the overall message conveyed by Martin Luther in his speech.

**INTERPRETATION AND RECOMMENDATIONS**

The frequency count and word cloud indicate that "freedom" is the central theme of the speech.

Based on this analysis, I would recommend that organizations and individuals continue to focus on promoting civil rights, freedom, and equality for all. They should also focus on eliminating discrimination and promoting diversity and inclusiveness in all their communications. Incorporating visuals and powerful language can help to reinforce these messages and inspire action.

Another recommendation is to incorporate additional variables, such as demographic data on the audience or historical events occurring at the time of the speech. This may provide additional insights into the speech.

For further analysis, it may be useful to explore other speeches and writings of Martin Luther King Jr. to gain a deeper understanding of his message and ideas. It may also be useful to compare this speech with speeches by other leaders to identify common themes and differences.

**REFERENCES**

Holtz, Y. (n.d.). *The R Graph Gallery – Help and inspiration for R charts*. The R Graph Gallery. <https://r-graph-gallery.com/>

*Home - RDocumentation*. (n.d.). https://www.rdocumentation.org/

|  |
| --- |
| **APPENDIX** |

# Install

install.packages("tm") # for text mining

install.packages("SnowballC") # for text stemming

install.packages("wordcloud") # word-cloud generator

install.packages("RColorBrewer") # color palettes

# Load

library("tm")

library("SnowballC")

library("wordcloud")

library("RColorBrewer")

#text <- readLines(file.choose())

filePath <- "http://www.sthda.com/sthda/RDoc/example-files/martin-luther-king-i-have-a-dream-speech.txt"

text <- readLines(filePath)

# Load the data as a corpus

docs <- Corpus(VectorSource(text))

inspect(docs)

toSpace <- content\_transformer(function (x , pattern ) gsub(pattern, " ", x))

docs <- tm\_map(docs, toSpace, "/")

docs <- tm\_map(docs, toSpace, "@")

docs <- tm\_map(docs, toSpace, "\\|")

# 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 your own stop word

# specify your stopwords as a character vector

docs <- tm\_map(docs, removeWords, c("blabla1", "blabla2"))

# Remove punctuations

docs <- tm\_map(docs, removePunctuation)

# Eliminate extra white spaces

docs <- tm\_map(docs, stripWhitespace)

# Text stemming

docs <- tm\_map(docs, stemDocument)

dtm <- TermDocumentMatrix(docs)

m <- as.matrix(dtm)

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

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

head(d, 10)

set.seed(1234)

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"))

findFreqTerms(dtm, lowfreq = 4)

findAssocs(dtm, terms = "freedom", corlimit = 0.3)

head(d, 10)

barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,

col ="lightblue", main ="Most frequent words",

ylab = "Word frequencies")