RStudio CH1 DGA Text Analysis for Hana

* Manually cleaned data removing all unnecessary punctuation, whitespace, and numbers.
* Saved cleaned text as a .txt file
* Read the data into RStudio
  + library(readtext)
  + DGA\_CH1\_raw <- readtext("/Users/connornorton/Desktop/DGA\_CH\_1\_clean.txt")
* Add text to a Corpus
  + DGA\_CH1\_Corpus <- Corpus(VectorSource(DGA\_CH1\_raw))
* Preprocess the Corpus
  + DGA\_CH1\_Corpus <- tm\_map(DGA\_CH1\_Corpus, tolower)
  + DGA\_CH1\_Corpus <- tm\_map(DGA\_CH1\_Corpus, removePunctuation)
  + DGA\_CH1\_Corpus <- tm\_map(DGA\_CH1\_Corpus, removeNumbers)
  + Removing stopwords
    - library(stopwords)
    - stop\_words <- stopwords("en")
    - stop\_words <- stop\_words[!stop\_words %in% c("can", "cannot", "could", "may", "must", "should", "shall", "would", "are", "be", "able", "to", "need")]
    - DGA\_CH1\_Corpus <- tm\_map(DGA\_CH1\_Corpus, removeWords, stop\_words)
* Document Term Matrix
  + Create a Document Term Matrix from the text in the Corpus
    - DGA\_CH1\_DTM <- DocumentTermMatrix(DGA\_CH1\_Corpus)
  + Find modal verb frequency using *tm* library
    - library(tm)
    - modal\_verb\_freq\_CH1 <- colSums(as.matrix(DGA\_CH1\_DTM[, c("can", "cannot", "could", "may", "must", "should", "shall", "would")]))
      * function could cause error if each word isn’t in text, may have to remove some modal verbs
      * modal\_verb\_freq\_CH1 <- colSums(as.matrix(DGA\_CH1\_DTM[, c("can", "cannot", "could", "may", "should", "would")]))
    - Check for modal verb PHRASE frequency as well
      * library(stringr)
      * Convert data into a string
        + DGA\_CH1\_string <- as.String(DGA\_CH1\_Corpus)
      * freq\_beAbleTo\_CH1 <- str\_count(DGA\_CH1\_string, "are able to")
        + can skip this for CH1 because doesn’t exist
      * freq\_areAbleTo\_CH1 <- str\_count(DGA\_CH1\_string, "be able to")
      * freq\_needTo\_CH1 <- str\_count(DGA\_CH1\_string, "need to")
      * freq\_needsTo\_CH1 <- str\_count(DGA\_CH1\_string, "needs to")
    - Combine all modal verb frequencies into one vector
      * total\_modal\_verb\_freq\_CH1 <- append(modal\_verb\_freq\_CH1, freq\_areAbleTo\_CH1)
      * total\_modal\_verb\_freq\_CH1 <- append(total\_modal\_verb\_freq\_CH1, freq\_beAbleTo\_CH1)
      * total\_modal\_verb\_freq\_CH1 <- append(total\_modal\_verb\_freq\_CH1, freq\_needTo\_CH1)
      * total\_modal\_verb\_freq\_CH1 <- append(total\_modal\_verb\_freq\_CH1, freq\_needsTo\_CH1)
      * names(total\_modal\_verb\_freq\_CH1)[#] <- "are able to"
        + can skip this for CH1 because doesn’t exist
      * names(total\_modal\_verb\_freq\_CH1)[7] <- "be able to"
      * names(total\_modal\_verb\_freq\_CH1)[8] <- "need to"
      * names(total\_modal\_verb\_freq\_CH1)[9] <- "needs to"
    - Generate Table
      * table\_CH1\_modal\_verb\_freq <- data.frame(values = unlist(total\_modal\_verb\_freq\_CH1))

Table

Description automatically generated

* + Find all word frequency
    - library(tm)
    - all\_term\_freq\_CH1 <- colSums(as.matrix(DGA\_CH1\_DTM))
* Create individual word tokens from the Corpus
  + all\_CH1\_tokens <- text\_tokens(DGA\_CH1\_Corpus)
* Check the term stats of the Corpus
  + term\_stats(DGA\_CH1\_Corpus, subset = !term %in% stop\_words)

Table

Description automatically generated

* Generate word clouds
  + library(wordcloud)
  + Of all terms
    - All Black
      * Only the terms that appear < 3 times removed after custom stopwords are removed
        + wordcloud(names(all\_term\_freq\_CH1), all\_term\_freq\_CH1, min.freq = 3)
        + min freq = 3 because otherwise wordcloud would be too large

Text

Description automatically generated

* + - * Only the terms that appear < 6 times removed after custom stopwords are removed
        + wordcloud(names(all\_term\_freq\_CH1), all\_term\_freq\_CH1, min.freq = 6)

Text

Description automatically generated

* + **Of only modal verbs**
    - Basic (all black)
      * wordcloud(names(total\_modal\_verb\_freq\_CH1), total\_modal\_verb\_freq\_CH1, min.freq = 1)

A picture containing text

Description automatically generated Text

Description automatically generated

Text

Description automatically generated A picture containing text

Description automatically generated

* + - Color (RdBu)
      * wordcloud(names(total\_modal\_verb\_freq\_CH1), total\_modal\_verb\_freq\_CH1, min.freq = 1, colors = brewer.pal(6,"RdBu"))

**Logo, company name

Description automatically generated** **Logo, company name

Description automatically generated**

**A picture containing company name

Description automatically generated** **Logo, company name

Description automatically generated**