Assignment 15

Allison Tessman

2024-04-12

Question:

Find two text datasets and do the follows.

* Print out a sample of the Document Term Matrix using term frequency
* Print out a sample of the Document Term Matrix using tf-idf
* Cluster the words in the dataset using k-means.
* Plot the words frequency of the words in each clusters
* Plot the frequency of the part of speech of all the words in the dataset.

Text Data Set #1:

#document term matrix- term freq  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidytext)  
library(tm)

## Loading required package: NLP  
##   
## Attaching package: 'NLP'  
##   
## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(wordcloud)

## Loading required package: RColorBrewer

df <- read\_csv("~/Applied Analystics SAS Prog/mymath475/CNNtext.csv")

## Rows: 11490 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): id, article, highlights  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

df <- df %>%   
 select(id, highlights) %>%   
 rename(document = id,   
 texts = highlights)  
  
# create the DTM  
df\_tm <- df %>%   
 unnest\_tokens(output = word, input = texts) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(tibble(word = c(letters, LETTERS, "oh", 'just', as.character(c(1:100)))))

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`

word\_freq <- df\_tm %>%   
 group\_by(document) %>% count(word, sort = TRUE)  
  
df\_dtm <- word\_freq %>%   
 cast\_dtm(document = document, term = word, n)  
  
tm::inspect(df\_dtm)

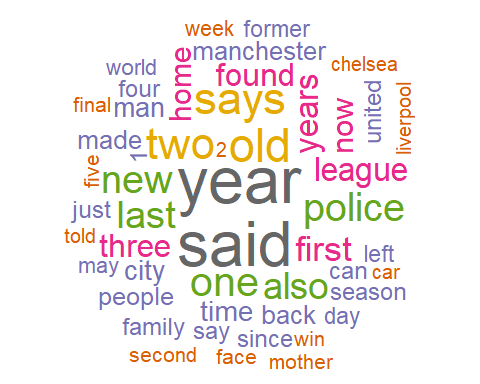
## <<DocumentTermMatrix (documents: 11490, terms: 37298)>>  
## Non-/sparse entries: 341639/428212381  
## Sparsity : 100%  
## Maximal term length: 33  
## Weighting : term frequency (tf)  
## Sample :  
## Terms  
## Docs also last new old one police said  
## 3815f10f7a7df0067d9ef27fb9e68d1803a615d0 0 0 0 0 0 0 0  
## 5a214c5117ec8b32d076653147711a9d2426fac2 0 5 0 0 0 0 0  
## 613e86fb4098c7bb4058aa17f5db4b5e4a1a3506 2 0 0 0 3 0 0  
## 6ddf76d9a7fcdbb060b394970e9558faa819eccc 0 0 2 0 0 0 0  
## 7cee65d7accdda886e2d9d2a2b7d02e3af72e41d 1 0 0 2 0 0 0  
## 8c9120d228fecd6791d7e5b5641549964500f18e 0 0 1 0 0 0 1  
## 95fc90b781eda59646706905b675ce5219725be1 0 0 1 0 0 1 0  
## cbed2a778a5e17d729c6e5ee5691b88710620dd7 1 0 2 0 0 0 0  
## d8fbffc5ba638a17bab565436ae8b3f197d7eb53 2 0 0 0 1 0 0  
## fa5594de0a40cf58a2aab30056f1860ed1177c9b 0 0 4 0 0 0 0  
## Terms  
## Docs says two year  
## 3815f10f7a7df0067d9ef27fb9e68d1803a615d0 0 0 0  
## 5a214c5117ec8b32d076653147711a9d2426fac2 0 3 0  
## 613e86fb4098c7bb4058aa17f5db4b5e4a1a3506 0 0 0  
## 6ddf76d9a7fcdbb060b394970e9558faa819eccc 0 0 2  
## 7cee65d7accdda886e2d9d2a2b7d02e3af72e41d 0 0 1  
## 8c9120d228fecd6791d7e5b5641549964500f18e 1 1 1  
## 95fc90b781eda59646706905b675ce5219725be1 0 1 1  
## cbed2a778a5e17d729c6e5ee5691b88710620dd7 0 1 1  
## d8fbffc5ba638a17bab565436ae8b3f197d7eb53 4 0 1  
## fa5594de0a40cf58a2aab30056f1860ed1177c9b 0 0 1

#document term matrix- using tf-idf  
df\_dtm <- weightTfIdf(df\_dtm)  
tm::inspect(df\_dtm)

## <<DocumentTermMatrix (documents: 11490, terms: 37298)>>  
## Non-/sparse entries: 341639/428212381  
## Sparsity : 100%  
## Maximal term length: 33  
## Weighting : term frequency - inverse document frequency (normalized) (tf-idf)  
## Sample :  
## Terms  
## Docs last league new old one police said  
## 47b39b8aa58ac18300727e7c481080cad3296658 0 0 0 0 0 0 0  
## 61a6105642804ffdb55a3b78f74380f8d099cde6 0 0 0 0 0 0 0  
## 84fec0986380d300b19cc2d4b3fadcb70fec38e4 0 0 0 0 0 0 0  
## a524048031be4aaecda0827a912c309b40ba7ae1 0 0 0 0 0 0 0  
## d94481ac0e95089d6b4b0877d58e0b6602a5894d 0 0 0 0 0 0 0  
## def00a311a1907c432f1347eb48394025dd1c733 0 0 0 0 0 0 0  
## dffb812ad66b9cc284fb8f3a8e50f34fc065ec0c 0 0 0 0 0 0 0  
## e41f76bc65fb1980dff8d2abc99dc564c8411429 0 0 0 0 0 0 0  
## e5139cfcb26808a1f29c18fa0f10c7b2e47bd999 0 0 0 0 0 0 0  
## f35c44125649a9ebc18d1807d9d26753a53394f7 0 0 0 0 0 0 0  
## Terms  
## Docs says two year  
## 47b39b8aa58ac18300727e7c481080cad3296658 0.1298443 0 0  
## 61a6105642804ffdb55a3b78f74380f8d099cde6 0.0000000 0 0  
## 84fec0986380d300b19cc2d4b3fadcb70fec38e4 0.0000000 0 0  
## a524048031be4aaecda0827a912c309b40ba7ae1 0.0000000 0 0  
## d94481ac0e95089d6b4b0877d58e0b6602a5894d 0.0000000 0 0  
## def00a311a1907c432f1347eb48394025dd1c733 0.0000000 0 0  
## dffb812ad66b9cc284fb8f3a8e50f34fc065ec0c 0.0000000 0 0  
## e41f76bc65fb1980dff8d2abc99dc564c8411429 0.0000000 0 0  
## e5139cfcb26808a1f29c18fa0f10c7b2e47bd999 0.0000000 0 0  
## f35c44125649a9ebc18d1807d9d26753a53394f7 0.0000000 0 0

#cluster with k means  
# df\_dtm <- removeSparseTerms(df\_dtm, 0.5)  
kmeans.data <- as.matrix(t(df\_dtm))  
kfit <- kmeans(kmeans.data, 3)  
  
# plot word cloud  
df %>%  
 unnest\_tokens(input = texts, output = word) %>%   
 anti\_join(get\_stopwords()) %>%   
 count(word, sort = TRUE) %>%   
 with(wordcloud(word, n, random.order = FALSE,   
 max.words = 50, colors=brewer.pal(8,"Dark2")))

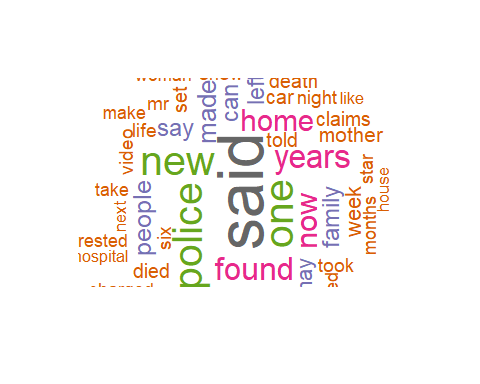
## Joining with `by = join\_by(word)`



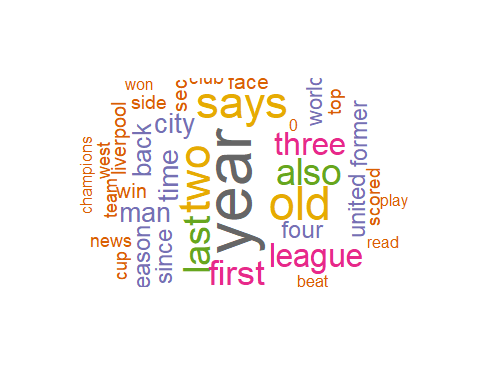
# Plot cluster   
k = 1  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))



# Plot cluster   
k = 2  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))



# Plot cluster   
k = 3  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))

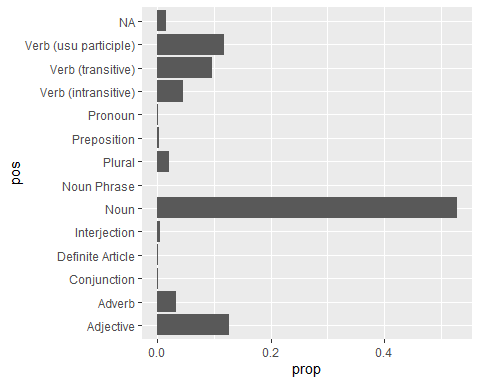


#plot freq of part of speech of all words  
library(tidyverse)  
library(tidytext)  
library(tm)  
library(wordcloud)  
df <- read\_csv("~/Applied Analystics SAS Prog/mymath475/CNNtext.csv")

## Rows: 11490 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): id, article, highlights  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

df <- df %>%   
 select(id, highlights) %>%   
 rename(document = id,   
 texts = highlights)  
  
stop\_word2 = tibble(word = c(letters, LETTERS, "oh", 'just'))  
  
df %>%   
 unnest\_tokens(input = texts, output = word) %>%   
 count(word, sort = TRUE) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(stop\_word2) %>%   
 inner\_join(parts\_of\_speech) %>% # join POS  
 count(pos) %>% # count  
 mutate(prop=n/sum(n)) %>%   
 slice\_max(prop, n = 15) %>%  
 ggplot()+geom\_col(aes(y = pos, x = prop), position = 'dodge')

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`



Text Data Set #2

#document term matrix- term freq  
library(tidyverse)  
library(tidytext)  
library(tm)  
library(wordcloud)  
df <- read\_csv("https://bryantstats.github.io/math475/assignments/netflix\_titles.csv")

## Rows: 7787 Columns: 12  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (11): show\_id, type, title, director, cast, country, date\_added, rating,...  
## dbl (1): release\_year  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

df = df %>%   
 select(type, description) %>%   
 rename(texts = description,  
 document = type)  
  
# create the DTM  
df\_tm <- df %>%   
 unnest\_tokens(output = word, input = texts) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(tibble(word = c(letters, LETTERS, "oh", 'just', as.character(c(1:100)))))

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`

word\_freq <- df\_tm %>%   
 group\_by(document) %>% count(word, sort = TRUE)  
  
df\_dtm <- word\_freq %>%   
 cast\_dtm(document = document, term = word, n)  
  
tm::inspect(df\_dtm)

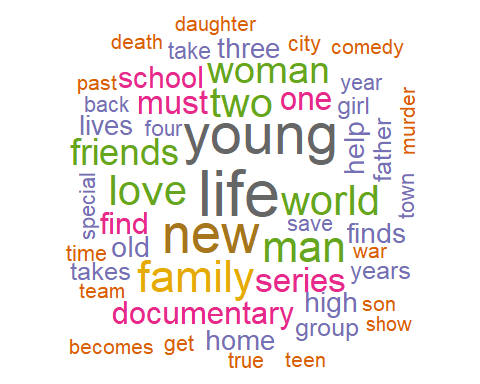
## <<DocumentTermMatrix (documents: 2, terms: 18770)>>  
## Non-/sparse entries: 25207/12333  
## Sparsity : 33%  
## Maximal term length: 18  
## Weighting : term frequency (tf)  
## Sample :  
## Terms  
## Docs family friends life love man new two woman world young  
## Movie 337 217 504 291 388 410 309 313 239 493  
## TV Show 160 166 242 156 76 203 134 89 214 162

#document term matrix- using tf-idf  
df\_dtm <- weightTfIdf(df\_dtm)  
tm::inspect(df\_dtm)

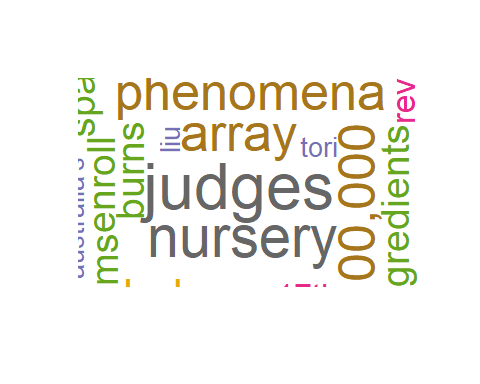
## <<DocumentTermMatrix (documents: 2, terms: 18770)>>  
## Non-/sparse entries: 12333/25207  
## Sparsity : 67%  
## Maximal term length: 18  
## Weighting : term frequency - inverse document frequency (normalized) (tf-idf)  
## Sample :  
## Terms  
## Docs 100,000 bride concert eve judges  
## Movie 0.0000000000 0.0002413005 0.0002667005 0.0002286005 0.0000000000  
## TV Show 0.0001974835 0.0000000000 0.0000000000 0.0000000000 0.0002539074  
## Terms  
## Docs nursery performance riffs stuck union  
## Movie 0.0000000000 0.0002921006 0.0002286005 0.0002286005 0.0002159004  
## TV Show 0.0002256954 0.0000000000 0.0000000000 0.0000000000 0.0000000000

#cluster with k means  
# df\_dtm <- removeSparseTerms(df\_dtm, 0.5)  
kmeans.data <- as.matrix(t(df\_dtm))  
kfit <- kmeans(kmeans.data, 3)  
  
# plot word cloud  
df %>%  
 unnest\_tokens(input = texts, output = word) %>%   
 anti\_join(get\_stopwords()) %>%   
 count(word, sort = TRUE) %>%   
 with(wordcloud(word, n, random.order = FALSE,   
 max.words = 50, colors=brewer.pal(8,"Dark2")))

## Joining with `by = join\_by(word)`



# Plot cluster   
k = 1  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))



# Plot cluster   
k = 2  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))



# Plot cluster   
k = 3  
  
cluster = names(kfit$cluster)[kfit$cluster==k]  
cluster = as\_tibble(cluster) %>%   
 rename(word = value)  
  
cluster = word\_freq <- df\_tm %>%   
 count(word, sort = TRUE) %>%   
 inner\_join(cluster, by = 'word')  
  
library(wordcloud)  
set.seed(2024)  
cluster %>% with(wordcloud(word, n, max.words = 50, random.order = FALSE, rot.per = 0.35,   
 colors = brewer.pal(8, "Dark2")))



#plot freq of part of speech of all words  
library(tidyverse)  
library(tidytext)  
library(tm)  
library(wordcloud)  
df <- read\_csv("~/Applied Analystics SAS Prog/mymath475/CNNtext.csv")

## Rows: 11490 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): id, article, highlights  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

df <- df %>%   
 select(id, highlights) %>%   
 rename(document = id,   
 texts = highlights)  
  
stop\_word2 = tibble(word = c(letters, LETTERS, "oh", 'just'))  
  
df %>%   
 unnest\_tokens(input = texts, output = word) %>%   
 count(word, sort = TRUE) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(stop\_word2) %>%   
 inner\_join(parts\_of\_speech) %>% # join POS  
 count(pos) %>% # count  
 mutate(prop=n/sum(n)) %>%   
 slice\_max(prop, n = 15) %>%  
 ggplot()+geom\_col(aes(y = pos, x = prop), position = 'dodge')

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`

