DSC510 - Programming Assignment 2

Alex Teboul

November 15, 2020

Problem 1: Text Mining and Natural Language Processing

Use these Head and Neck Cancer Medication Data to apply NLP/TM methods and investigate the Twitter corpus.

• Construct a VCorpus object using MEDICATION_SUMMARY.
• Clean the VCorpus object.
• Build document-term matrix (DTM).
• Compute the TF-IDF(term frequency - inverse document frequency).
• Use the DTM to construct a wordcloud.
• Explain what your wordcloud and frequencies are telling you about the data.

```
#Setup Working Directory
setwd("C:/Users/ateboul/Downloads")

#Read in Data
data <- read.csv("CaseStudy14_HeadNeck_Cancer_Medication.csv", header
= TRUE)

#libraries
library(tm)

## Warning: package 'tm' was built under R version 3.6.3

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 3.6.3

library(SnowballC)

## Warning: package 'SnowballC' was built under R version 3.6.3

library(rvest)

## Warning: package 'rvest' was built under R version 3.6.3

## Loading required package: xml2

## Warning: package 'xml2' was built under R version 3.6.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages
---------------------------------------------------------------------------
---------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tibble' was built under R version 3.6.3

## Warning: package 'tidyr' was built under R version 3.6.3

## Warning: package 'purrr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## -- Conflicts
---------------------------------------------------------------------------
------------------------------------- tidyverse_conflicts() --
## x ggplot2::annotate()     masks NLP::annotate()
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x purrr::pluck()          masks rvest::pluck()

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.6.3

## Loading required package: RColorBrewer
```

##• Construct a VCorpus object using MEDICATION_SUMMARY.

```
#Add an Index column to the dataset to distinguish that each row of
medical summary text comes from a different document
data <- tibble::rowid_to_column(data, "index")

#Explore the data
head(data)

##   index   PID ENC_ID seer_stage              MEDICATION_DESC
## 1     1 10000  46836          1                   ranitidine
## 2     2 10008  46886          1            heparin injection
## 3     3 10029  47034          4 ampicillin/sulbactam IVPB UH
## 4     4 10063  47240          1         fentaNYL injection UH
## 5     5 10071  47276          9                  simvastatin
## 6     6 10103  47511          1   dexamethasone (multiroute)
##
```

```
MEDICATION_SUMMARY
## 1                                                    (Zantac) 150 mg
tablet oral two times a day
## 2                                                      5,000 unit
subcutaneous three times a day
## 3
(Unasyn) 15 g IV every 6 hours
## 4 25 - 50 microgram IV every 5 minutes PRN severe pain\nMaximum
dose 200 mcg  Per PACU protocol
## 5                                                           (Zocor)
40 mg tablet oral at bedtime
## 6                                         2 mg IV/PO every 12 hours
May be administered IV or PO
##   DOSE      UNIT          FREQUENCY TOTAL_DOSE_COUNT
## 1  150        mg    two times a day                5
## 2 5000      unit  three times a day                3
## 3  1.5         g      every 6 hours               11
## 4   50 microgram    every 5 minutes                2
## 5   40        mg         at bedtime                1
## 6    2        mg     every 12 hours                2
```

```r
#head(data$MEDICATION_SUMMARY)

#Construct the VCorpus for medication summary (column 6 now)
medsumCorpus<-VCorpus(VectorSource(data[, 6]))
medsumCorpus
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 662
```

```r
#Check Corpus
head(medsumCorpus[[1]]$content)
```

```
## [1] "(Zantac) 150 mg tablet oral two times a day"
```

##• Clean the VCorpus object

```r
#Remove Words
medsumCorpus<-tm_map(medsumCorpus, removeWords, stopwords("english"))

#Remove Punctuation
medsumCorpus<-tm_map(medsumCorpus, removePunctuation)

#Remove Whitespace
medsumCorpus<-tm_map(medsumCorpus, stripWhitespace)

#Convert to Plain Text Document
```

```
medsumCorpus<-tm_map(medsumCorpus, PlainTextDocument)

#Stem the Documents
medsumCorpus<-tm_map(medsumCorpus, stemDocument)

#Check Corpus
medsumCorpus[[1]]$content

## [1] "Zantac 150 mg tablet oral two time day"
```

##• Build document-term matrix (DTM).

```
#Create Document-Term Matrix
dtm<-DocumentTermMatrix(medsumCorpus)
dtm

## <<DocumentTermMatrix (documents: 662, terms: 451)>>
## Non-/sparse entries: 5311/293251
## Sparsity           : 98%
## Maximal term length: 19
## Weighting          : term frequency (tf)

#Remove document meta data
#Relabel the Documents
dtm$dimnames$Docs<-as.character(1:662)
inspect(dtm)

## <<DocumentTermMatrix (documents: 662, terms: 451)>>
## Non-/sparse entries: 5311/293251
## Sparsity           : 98%
## Maximal term length: 19
## Weighting          : term frequency (tf)
## Sample             :
##      Terms
## Docs  day everi for glucos hour oral pain prn tablet time
##   132   1     0   8      8    0    0    0   0      0    1
##   195   0     0  32     33    0    0    0   0      0    0
##   196   0     0  32     33    0    0    0   0      0    0
##   227   1     0   8      8    0    0    0   0      0    1
##   269   1     0   8      8    0    0    0   0      0    1
##   277   1     0   8      8    0    0    0   0      0    1
##   335   1     0   8      8    0    0    0   0      0    1
##   336   1     0   8      8    0    0    0   0      0    1
##   39    1     0   8      8    0    0    0   0      0    1
##   67    1     0   8      8    0    0    0   0      0    1

#High frequency terms
#findFreqTerms(dtm, 5, 500)
```

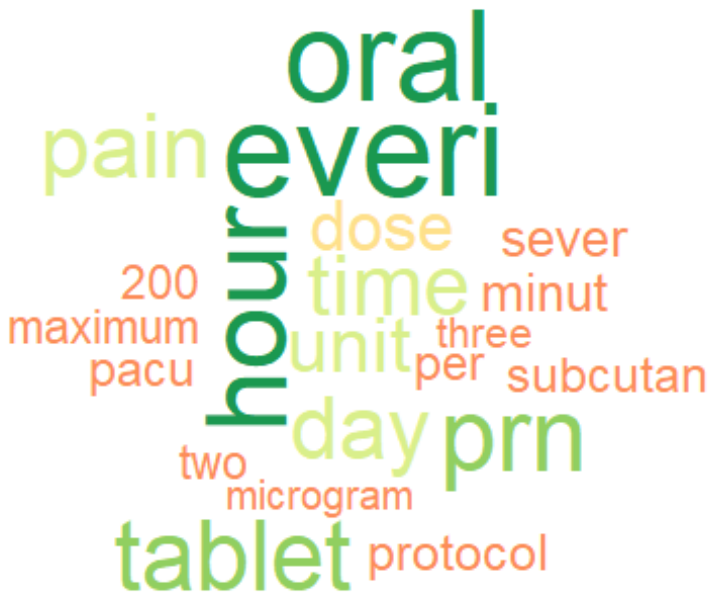
```
#Low frequency terms
#findFreqTerms(dtm, 1, 4)

#Remove Sparse Words
dtms<-removeSparseTerms(dtm, 0.90)
dtms

## <<DocumentTermMatrix (documents: 662, terms: 21)>>
## Non-/sparse entries: 3048/10854
## Sparsity           : 78%
## Maximal term length: 9
## Weighting          : term frequency (tf)

freq1<-sort(colSums(as.matrix(dtms)), decreasing=T)
freq1

##     everi       oral       hour        prn     tablet       pain
day
##       312        301        270        247        223        204
196
##      time       unit       dose      minut      sever   protocol
per
##       187        168        143        103         98         93
91
##  subcutan       pacu    maximum        two        200      three
microgram
##        88         87         86         86         82         71
67
```

##• Compute the TF-IDF(term frequency - inverse document frequency).

```
dtm.tfidf<-DocumentTermMatrix(medsumCorpus, control =
list(weighting=weightTfIdf))

## Warning in weighting(x): empty document(s): character(0)
character(0)
## character(0)

dtm.tfidf

## <<DocumentTermMatrix (documents: 662, terms: 451)>>
## Non-/sparse entries: 5311/293251
## Sparsity           : 98%
## Maximal term length: 19
## Weighting          : term frequency - inverse document frequency
(normalized) (tf-idf)
```

##• Use the DTM to construct a wordcloud.

```r
set.seed(123)
wordcloud(names(freq1), freq1,colors=brewer.pal(6, "RdYlGn"))
```



##• Explain what your wordcloud and frequencies are telling you about the data.

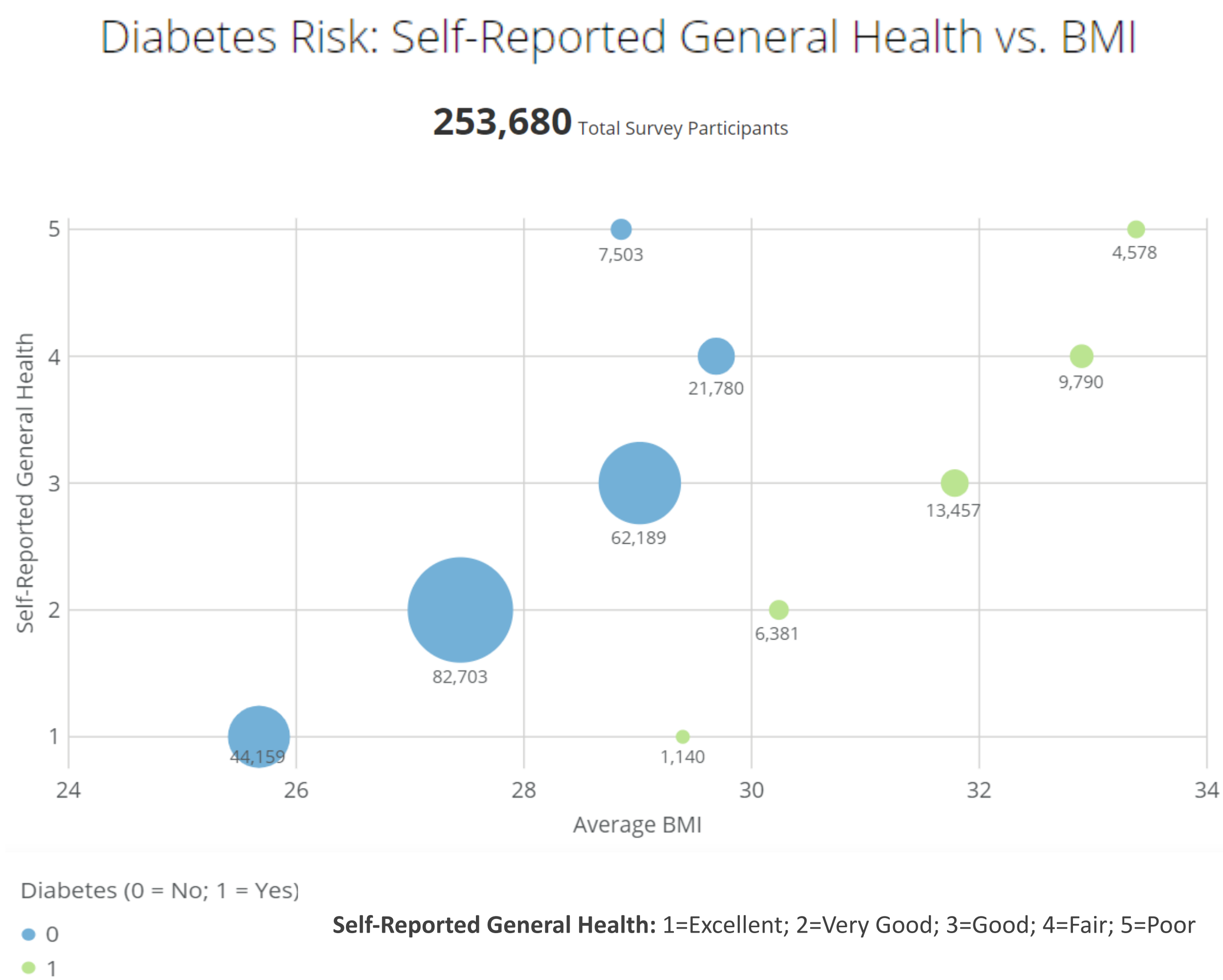
- The word cloud shows that doctors are recording medication information with respect to medication timing (words like “hour”, “everi”/every, “day”, “time”, “minut”/minute), with respect to amount (“200”, “three”, “two”, “maximum”, “microgram”), and with respect to type of medication (“oral”, “tablet”, “subcutan”/subcutaneous, “dose”, “unit”, “prn”). So the wordcloud contains a enough frequent text to give the audience an general understanding of the type of text found in the medication summary of the data table for head & neck cancer medication. Doctors are recording medication timing, amount of medication, and type of medication or delivery in the MEDICATION_SUMMARY field of the head and neck cancer data table.
- Interestingly the “prn” word is also important because it means ‘pro re nata’ or essentially take as needed. Many of the notes actually have the word pain or words describing pain following “prn”. So it’s really just saying that medication should be taken if the patient is in pain - an important point that is not recorded elsewhere in the data.

Problem 2: Create a Visualization and Tell a Story about the Data

Using your final project dataset, create a visualization discussed in the basic and advanced portions of this course.

Make tell the story behind your visualization.

- Remember the results are more than the statistics and mathematics to calculate it.
- The data story is referring to the application of the dataset.
- Remember all axes should have titles and the figure should be numbered and have a main title.



- The chart above shows the Self-Reported General Health versus Average BMI of BRFSS 2015 survey participants. Bubble size is based on the total number of survey participants that make up that bubble. The blue color represents participants that do not have diabetes and a small number that have prediabetes. The green color is for those that have diabetes. Note that 1 is Excellent health and 5 is Poor health on the y-axis.

- There is a clear separation between the non-diabetics and diabetics in terms of the average BMI of individuals at a given self-reported general health score. For the same self-reported general health score, people with diabetes are more likely to have a higher BMI. This comes as no surprise as BMI and diabetes risk are highly correlated and discussed in the literature.
- Interestingly, as average BMI of survey participants goes up, their self-reported general health status becomes more negative. In the data, 5 is Poor health and 1 is Excellent health. Individuals at higher BMIs tend to respond to the question "would you say that in general your health is:" in a more negative manner. This is similar among diabetics and non-diabetics.
- BMI and Self-Reported General Health are two of the important features selected by the entropy feature selection method of the Random Forest method used in my final project.

Problem 3: Create an Infographic about your final project results.

Note: You can create the infographic in Excel, PowerPoint, or a similar type of program. As learned from class, infographics can be a great way of disseminating information about your study.

Create **<u>1 infographic</u>** about your results or an important fact from the literature review the audience should know about the study.

(see next page)

Modeling Diabetes Risk: BRFSS 2015

1 in 10 Americans have diabetes...

...or 34.2 million people[1]

High Blood Pressure

Do you have high blood pressure?

General Health

Rate your health:

○Excellent ○Very Good ○Good ○Fair ○Poor

High Cholesterol

Do you have high cholesterol?

Body Mass Index (BMI)

What is your BMI?

Age

What is your age?

Using these 5 Questions....

- **Diabetes can be predicted with**
  - 74% (+/- 0.01) Accuracy
  - 0.82 (+/- 0.01) AUC
  - 0.78 (+/- 0.01) Recall
  - 0.71 (+/- 0.02) Precision
- **Models Tested**
  - Neural Networks
  - Random Forests
  - AdaBoost
  - Gradient Boosting
- **Model Specs**
  - 75,323 responses used in models
  - Undersampling used to create this balanced 50-50 dataset
  - 21 total variables assessed

1 Centers for Disease Control and Prevention. National Diabetes Statistics Report, 2020. Atlanta, GA: Centers for Disease Control and Prevention, U.S. Dept of Health and Human Services; 2020. https://www.cdc.gov/diabetes/pdfs/data/statistics/national-diabetes-statistics-report.pdf