Clinical_NLP

Gayatri Kudchadker

3/28/2020

Import libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidy
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v stringr 1.4.0
## v tidyr   1.0.0     v forcats 0.4.0
## v readr   1.3.1
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse_
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(bigrquery)
```

Read the dataset

```
con <- DBI::dbConnect(drv = bigquery(), project = "learnclinicaldatascience")
dnotes <- tbl(con, "course4_data.diabetes_notes") %>% collect()
```

```
## Using an auto-discovered, cached token.
## To suppress this message, modify your code or options to clearly consent to the use of a cached toke
## See gargle's "Non-interactive auth" vignette for more details:
## https://gargle.r-lib.org/articles/non-interactive-auth.html
## The bigrquery package is using a cached token for gkder264@gmail.com.
```

```
head(dnotes)
```

```
## # A tibble: 6 x 3
##   NOTE_ID NOTE_TYPE         TEXT
##     <int> <chr>             <chr>
## 1       1 History and Phys~ "CHIEF COMPLAINT:  Dog bite to his right lower leg.~
## 2       2 History and Phys~ "CHIEF COMPLAINT:    Left hip pain.\n\nHISTORY OF P~
## 3       3 Discharge Summary "CC: Dysarthria\n\nHX: This 52y/o RHF was transferr~
## 4       4 Operative Note    "PRE-OP DIAGNOSIS:  Osteoporosis, pathologic fractu~
## 5       5 Discharge Summary "CC: Left hemibody numbness.\n\nHX: This 44y/o RHF ~
## 6       6 Operative Note    "PREOPERATIVE DIAGNOSIS:  Left renal mass, left ren~
```

Steps: 1. Detect all notes with the term “diabetes”. Remove false positives 2. Search text with diabetes for direct presence of complications 3. Search Leftover texts after step 2 for symptoms related to complications

We want to find if the patient has diabetes thus we will remove sections like Family History, Allergies which do not tell us directly about the patient

```
diabetic <- dnotes %>% mutate(TEXT_WITH_DIABETES = case_when(str_detect(TEXT, regex("FAMILY\\s*HISTORY"
```

Filter dataset with notes having the term diabetes or diabetic

```
with_diab <- diabetic %>% filter(str_detect(string = TEXT_WITH_DIABETES, pattern = regex("diabet(es)?(i
```

Also extract texts without the word diabetes to analyze later

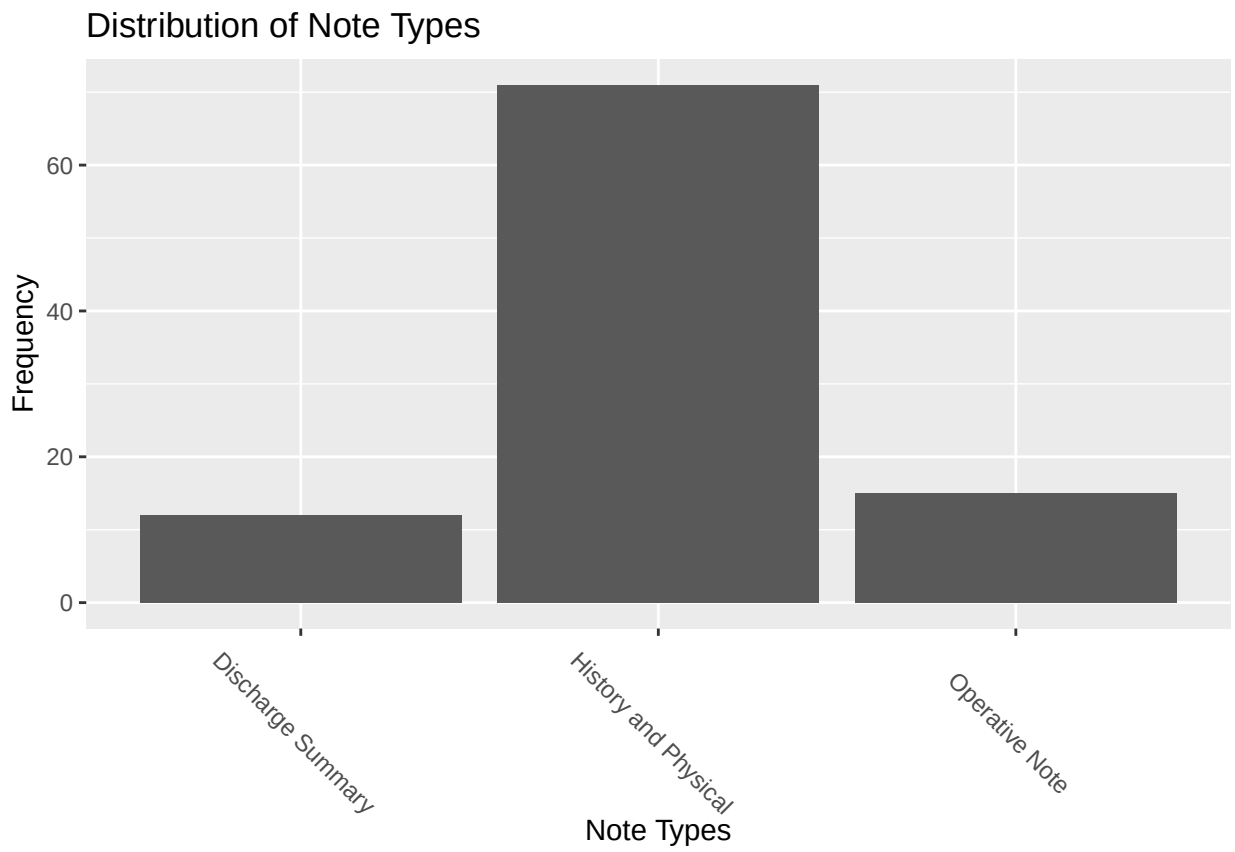
```
without_diab <- diabetic %>% filter(!(NOTE_ID %in% with_diab[["NOTE_ID"]]))
```

There are some text which contain the word “diabetes” but are associated with negative context like ’no history of diabetes". We will remove such text and add to the without_diab data table

```
keyword <- "(?<![a-zA-Z])((no diabetes)|(is not diabetic)|(negative for diabetes)|(no history of diabet
neg_diab <- with_diab %>% filter(str_detect(TEXT_WITH_DIABETES, regex(keyword, ignore_case=TRUE)))
with_diab <- with_diab %>% filter(!str_detect(TEXT_WITH_DIABETES, regex(keyword, ignore_case=TRUE)))
without_diab <- rbind(without_diab, neg_diab)
```

88 transcriptions have the term diabetes in the text. Distribution of Note Types

```
ggplot(with_diab, aes(NOTE_TYPE)) + geom_bar() + labs(title = "Distribution of Note Types", x = "Note Ty
```



Remove unwanted tables to free space

```
rm(dnotes)
rm(diabetic)
rm(neg_diab)
```

For text with word diabetes, let us now find any direct mention of words “Neuropathy”, “Nephropathy”, “Retinopathy”. We will also extract a window around this such words to analyze the text further.

```
with_diab <- with_diab %>%
         mutate(DIABETES_TYPE = case_when(str_detect(TEXT_WITH_DIABETES, regex("Neuropath(y)?(ic)?", ig
                                          str_detect(TEXT_WITH_DIABETES, regex("Nephropath(y)?(ic)?", igno
                                          str_detect(TEXT_WITH_DIABETES, regex("Retinopath(y)?(ic)?", igno
                                          TRUE ~ ""))
```

We will separate text where diabetic type was found and the ones where it was not

```
with_type <- with_diab %>% filter(DIABETES_TYPE != "")
head(with_type)
```

```
## # A tibble: 6 x 5
##   NOTE_ID NOTE_TYPE   TEXT              TEXT_WITH_DIABETES         DIABETES_TYPE
##     <int> <chr>       <chr>             <chr>                      <chr>
## 1       6 Operative ~ "PREOPERATIVE DI~ "PREOPERATIVE DIAGNOSIS: ~ Nephropathy
## 2       7 Operative ~ "S - An 84-year-~ "S - An 84-year-old diabe~ Neuropathy
## 3      12 Operative ~ PREOPERATIVE DIA~ PREOPERATIVE DIAGNOSES1. ~ Nephropathy
## 4      24 Operative ~ "PREOPERATIVE DI~ "PREOPERATIVE DIAGNOSIS: ~ Neuropathy
## 5      27 History an~ "CHIEF COMPLAINT~ "CHIEF COMPLAINT:  Penile~ Neuropathy
## 6      30 History an~ "HISTORY OF PRES~ "HISTORY OF PRESENT ILLNE~ Neuropathy
```

```
without_type <- with_diab %>% filter(DIABETES_TYPE == "")
head(without_type)
```

```
## # A tibble: 6 x 5
##   NOTE_ID NOTE_TYPE   TEXT              TEXT_WITH_DIABETES         DIABETES_TYPE
##     <int> <chr>       <chr>             <chr>                      <chr>
## 1       2 History an~ "CHIEF COMPLAINT~ "CHIEF COMPLAINT:    Left~ ""
## 2       4 Operative ~ "PRE-OP DIAGNOSI~ "PRE-OP DIAGNOSIS:  Osteo~ ""
## 3       8 History an~ "HISTORY OF PRES~ "HISTORY OF PRESENT ILLNE~ ""
## 4      10 Operative ~ "PREOPERATIVE DI~ "PREOPERATIVE DIAGNOSIS: ~ ""
## 5      13 Operative ~ PREOPERATIVE DIA~ PREOPERATIVE DIAGNOSES1. ~ ""
## 6      14 History an~ "CHIEF COMPLAINT~ "CHIEF COMPLAINT:  Bladde~ ""
```

Let us create a function to extract a text window dataframe :- diabetic dataset keyword :- word surrounding which the text should be extracted half_window_size :- length of the text to be extracted from both ends of the word

```
extract_text_window <- function(dataframe, keyword, half_window_size) {
  dataframe %>%
  group_by(NOTE_ID) %>%
  mutate(WORDS = TEXT_WITH_DIABETES) %>%
  separate_rows(WORDS, sep = "[ \\n]+") %>%
  mutate(INDEX = seq(from = 1, to = n(), by = 1.0),
      WINDOW_START = case_when(INDEX - half_window_size < 1 ~ 1, TRUE ~ INDEX - half_window_size),
      WINDOW_END = case_when(INDEX + half_window_size > max(INDEX) ~ max(INDEX), TRUE ~ INDEX + half_wi
      WINDOW = word(string = TEXT_WITH_DIABETES, start = WINDOW_START, end = WINDOW_END, sep = "[ \\n]+
  ungroup() %>%
  filter(str_detect(string = WORDS, pattern = regex(keyword, ignore_case = TRUE)))
}
```

Let us first extract the winow around the diabetic types

```
keyword <- "(Neuropathy)|(Nephropathy)|(Retinopathy)"
with_type <- extract_text_window(with_type, keyword, 10)
head(with_type)
```

```
## # A tibble: 6 x 10
##   NOTE_ID NOTE_TYPE TEXT  TEXT_WITH_DIABE~ DIABETES_TYPE WORDS INDEX
##     <int> <chr>     <chr> <chr>            <chr>         <chr> <dbl>
## 1       6 Operativ~ "PRE~ "PREOPERATIVE D~ Nephropathy   neph~    48
## 2       7 Operativ~ "S -~ "S - An 84-year~ Neuropathy    Neur~   224
## 3      12 Operativ~ PREO~ PREOPERATIVE DI~ Nephropathy   Neph~     6
## 4      24 Operativ~ "PRE~ "PREOPERATIVE D~ Neuropathy    neur~    49
## 5      27 History ~ "CHI~ "CHIEF COMPLAIN~ Neuropathy    neur~    51
## 6      30 History ~ "HIS~ "HISTORY OF PRE~ Neuropathy    neur~    32
## # ... with 3 more variables: WINDOW_START <dbl>, WINDOW_END <dbl>, WINDOW <chr>
```

After examining the context words, we see that NOTE_ID 136 says there is no evidence of diabetic Retinopathy. So we remove that note.

```
with_type <- with_type %>% filter(NOTE_ID != 136) %>% select(-c(WORDS, INDEX, WINDOW_START, WINDOW_END,

without_type <- without_type %>% select(-c(DIABETES_TYPE, TEXT))
```

Let us find symptoms in the text

```
keyword <- "((optic )?nerve(s)? damage)|(damage(d)? nerve(s)?)|(renal disease)|(kidney disease)|(renal
without_type <- without_type %>% mutate(SYMPTOMS = str_extract_all(TEXT_WITH_DIABETES, regex(keyword, ig

keys1 <- c("nerve damage")
keys2 <- c("kidney disease", "renal disease", "renal failure", "kidney failure", "chronic renal", "chro
keys3 <- c("retinal damage", "blindness", "optic nerve damage")

types <- c()
for (i in 1:nrow(without_type)){
  sym <- without_type[[i, "SYMPTOMS"]]
  if(length(sym) > 0){
    if(all(tolower(unique(sym)) %in% keys1)){
      types <- c(types, "Neuropathy")
    }
    else if(all(tolower(unique(sym)) %in% keys2)){
      types <- c(types, "Nephropathy")
    }
    else if(all(tolower(unique(sym)) %in% keys3)){
      types <- c(types, "Retinopathy")
    }
    else{
      types <- c(types, "")
    }
  }
  else{
      types <- c(types, "")
    }
}

without_type$DIABETES_TYPE <- types
```

```
head(without_type)
```

```
## # A tibble: 6 x 5
##   NOTE_ID NOTE_TYPE     TEXT_WITH_DIABETES                SYMPTOMS DIABETES_TYPE
##     <int> <chr>         <chr>                             <list>   <chr>
## 1       2 History and ~ "CHIEF COMPLAINT:    Left hip pa~ <chr [0~ ""
## 2       4 Operative No~ "PRE-OP DIAGNOSIS:  Osteoporosis~ <chr [0~ ""
## 3       8 History and ~ "HISTORY OF PRESENT ILLNESS:  Th~ <chr [0~ ""
## 4      10 Operative No~ "PREOPERATIVE DIAGNOSIS:   Hemat~ <chr [0~ ""
## 5      13 Operative No~ PREOPERATIVE DIAGNOSES1.  End-st~ <chr [3~ Nephropathy
## 6      14 History and ~ "CHIEF COMPLAINT:  Bladder cance~ <chr [0~ ""
```

Filter out the rows where symptoms have been found. Combine it with with_type table

```
sym_df <- without_type %>% filter(DIABETES_TYPE != "") %>% select(-c(SYMPTOMS))
final_df <- rbind(with_type, sym_df)
final_df <- final_df %>% distinct()
head(final_df)
```

```
## # A tibble: 6 x 4
##   NOTE_ID NOTE_TYPE      TEXT_WITH_DIABETES                       DIABETES_TYPE
##     <int> <chr>          <chr>                                    <chr>
## 1       6 Operative Note "PREOPERATIVE DIAGNOSIS:  Left renal mas~ Nephropathy
## 2       7 Operative Note "S - An 84-year-old diabetic female, 5'7~ Neuropathy
## 3      12 Operative Note PREOPERATIVE DIAGNOSES1.  End-stage rena~ Nephropathy
## 4      24 Operative Note "PREOPERATIVE DIAGNOSIS:  Gangrene osteo~ Neuropathy
## 5      27 History and P~ "CHIEF COMPLAINT:  Penile discharge, inf~ Neuropathy
## 6      30 History and P~ "HISTORY OF PRESENT ILLNESS: The patient~ Neuropathy
```

```
rm(sym_df)
rm(with_type)
```

Results

Now let us compare the accuracy of the results with the gold standard data

```
gold <- tbl(con, "course4_data.diabetes_goldstandard") %>% collect()
head(gold)
```

```
## # A tibble: 6 x 5
##   NOTE_ID ANY_DIABETIC_COMP~ DIABETIC_NEUROPA~ DIABETIC_NEPHRO~ DIABETIC_RETINO~
##     <int>              <int>             <int>            <int>            <int>
## 1       1                  0                 0                0                0
## 2       2                  0                 0                0                0
## 3       3                  0                 0                0                0
## 4       4                  0                 0                0                0
## 5       5                  0                 0                0                0
## 6       6                  1                 0                1                0
```

```
true_y <- gold %>% filter(ANY_DIABETIC_COMPLICATION == 1) %>% select(NOTE_ID) %>% unlist()
predicted_y <- final_df %>% select(NOTE_ID) %>% unlist()
sprintf("Number of Actual Diabetic Texts: %i", length(true_y))
```

```
## [1] "Number of Actual Diabetic Texts: 27"
```

```
sprintf("Number of Predicted Diabetic Texts: %i", length(predicted_y))
```

```
## [1] "Number of Predicted Diabetic Texts: 29"
```

```
sprintf("Total Correctly Classified Diabetic Texts: %i", sum(true_y %in% predicted_y))
```

```
## [1] "Total Correctly Classified Diabetic Texts: 22"
```

```
print("Incorrectly Identified Texts:")
```

```
## [1] "Incorrectly Identified Texts:"
```

```
predicted_y[!(predicted_y %in% true_y)]
```

```
## NOTE_ID20 NOTE_ID21 NOTE_ID22 NOTE_ID23 NOTE_ID25 NOTE_ID27 NOTE_ID28
##        15        41        55        69        94       112       123
```

```
print("Missed Texts:")
```

```
## [1] "Missed Texts:"
```

```
true_y[!(true_y %in% predicted_y)]
```

```
##  NOTE_ID5  NOTE_ID6  NOTE_ID7 NOTE_ID14 NOTE_ID23
##        14        16        18        51       118
```

Neuropathy

```
true_neu <- gold %>% filter(DIABETIC_NEUROPATHY == 1) %>% select(NOTE_ID) %>% unlist()
predicted_neu <- final_df %>% filter(DIABETES_TYPE == "Neuropathy") %>% select(NOTE_ID) %>% unlist()
sprintf("Number of Actual Diabetic Texts: %i", length(true_neu))
```

```
## [1] "Number of Actual Diabetic Texts: 15"
```

```
sprintf("Number of Predicted Diabetic Texts: %i", length(predicted_neu))
```

```
## [1] "Number of Predicted Diabetic Texts: 14"
```

```
sprintf("Total Correctly Classified Neuropathy Texts: %i", sum(true_neu %in% predicted_neu))
```

```
## [1] "Total Correctly Classified Neuropathy Texts: 12"
```

```
print("Incorrectly Identified Texts:")
```

```
## [1] "Incorrectly Identified Texts:"
```

```
predicted_neu[!(predicted_neu %in% true_neu)]
```

```
## NOTE_ID10 NOTE_ID14
##        86        94
```

```
print("Missed Texts:")
```

```
## [1] "Missed Texts:"
```

```
true_neu[!(true_neu %in% predicted_neu)]
```

```
##  NOTE_ID2  NOTE_ID3 NOTE_ID13
##        14        18       118
```

Nephropathy

```
true_neph <- gold %>% filter(DIABETIC_NEPHROPATHY == 1) %>% select(NOTE_ID) %>% unlist()
predicted_neph <- final_df %>% filter(DIABETES_TYPE == "Nephropathy") %>% select(NOTE_ID) %>% unlist()
sprintf("Number of Actual Nephropathy Texts: %i", length(true_neph))
```

```
## [1] "Number of Actual Nephropathy Texts: 10"
```

```
sprintf("Number of Predicted Nephropathy Texts: %i", length(predicted_neph))
```

```
## [1] "Number of Predicted Nephropathy Texts: 14"
```

```
sprintf("Total Correctly Classified Nephropathy Texts: %i", sum(true_neph %in% predicted_neph))
```

```
## [1] "Total Correctly Classified Nephropathy Texts: 8"
```

```
print("Incorrectly Identified Texts:")
```

```
## [1] "Incorrectly Identified Texts:"
```

```
predicted_neph[!(predicted_neph %in% true_neph)]
```

```
##  NOTE_ID7  NOTE_ID8  NOTE_ID9 NOTE_ID10 NOTE_ID13 NOTE_ID14
##        15        41        55        69       112       123
```

```
print("Missed Texts:")
```

```
## [1] "Missed Texts:"
```

```
true_neph[!(true_neu %in% predicted_neph)]
```

```
##  NOTE_ID1  NOTE_ID2  NOTE_ID3  NOTE_ID4  NOTE_ID5  NOTE_ID6  NOTE_ID7  NOTE_ID8
##         6        12        13        27        42        51        62        85
##  NOTE_ID9 NOTE_ID10      <NA>      <NA>      <NA>      <NA>      <NA>
##       108       140        NA        NA        NA        NA        NA
```

Retinopathy

```
true_ret <- gold %>% filter(DIABETIC_RETINOPATHY == 1) %>% select(NOTE_ID) %>% unlist()
predicted_ret <- final_df %>% filter(DIABETES_TYPE == "Retinopathy") %>% select(NOTE_ID) %>% unlist()
sprintf("Number of Actual Retinopathy Texts: %i", length(true_ret))
```

```
## [1] "Number of Actual Retinopathy Texts: 3"
```

```
sprintf("Number of Predicted Retinopathy Texts: %i", length(predicted_ret))
```

```
## [1] "Number of Predicted Retinopathy Texts: 1"
```

```
sprintf("Total Correctly Classified Retinopathy Texts: %i", sum(true_ret %in% predicted_ret))
```

```
## [1] "Total Correctly Classified Retinopathy Texts: 1"
```

```
print("Incorrectly Identified Texts:")
```

```
## [1] "Incorrectly Identified Texts:"
```

```
predicted_ret[!(predicted_ret %in% true_ret)]
```

```
## named integer(0)
```

```r
print("Missed Texts:")
```

```
## [1] "Missed Texts:"
```

```r
true_ret[!(true_ret %in% predicted_ret)]
```

```
## NOTE_ID1 NOTE_ID2
##       16       86
```