R Notebook

load packages

```
require(dplyr)
```

```
## Loading required package: 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(Matrix)
library(corpus)
library(tidytext)
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP
```

Load problem data

```
setwd("/Users/YaoJunyan/Documents/cpsx-text analysis")
prob_data<-read.csv("problem_data.csv",stringsAsFactors=FALSE)
head(prob_data,10)
```

```
##     X
## 1   1
## 2   2
## 3   3
## 4   4
## 5   7
## 6   8
## 7  11
## 8  12
## 9  13
## 10 14
##
                                                                       value
## 1

## 2                                            \n      The diagram is part of a scale d
rawing of a house. What is the length, in feet, of the side labeled x?\n
## 3  \n      You and your partner can each make ONE selection from the following list o
f hints. Use this information to provide your answer in the box below.\n
## 4
                Value of A  Value of B  Value of C  Value of D  Value of E
## 5
                      What is the length, in feet, of the side labeled  x ?
## 6
                                                             x   =\n    \n
## 7

## 8                                                \n    The diagram is part of a scale
drawing of a house. What is the length, in feet, of the side labeled x?\n
## 9                          \n    You can make TWO selections from the following list
of hints. Use this information to provide your answer in the box below.\n
## 10
                Value of A  Value of B  Value of C  Value of D  Value of E
##                  L1
## 1  008EG_COL_H1.xml
## 2  008EG_COL_H1.xml
## 3  008EG_COL_H1.xml
## 4  008EG_COL_H1.xml
## 5  008EG_COL_H2.xml
## 6  008EG_COL_H2.xml
## 7  008EG_IND_H1.xml
## 8  008EG_IND_H1.xml
## 9  008EG_IND_H1.xml
## 10 008EG_IND_H1.xml
```

Load the chunk seperated chat data

```
chat_data<- read.csv("/Users/YaoJunyan/Documents/cpsx-text analysis/chunk_seperated fil
e.csv",stringsAsFactors=FALSE)
head(chat_data,10)
```

```
##     X group_id user_id       time    type       module correct
## 1   1        1    6181 1473272857    chat         <NA>      NA
## 2   2        1    6181 1473272892    chat         <NA>      NA
## 3   3        1    5913 1473272900    chat         <NA>      NA
## 4   4        1    5913 1473272912    chat         <NA>      NA
## 5   5        1    6181 1473272912    chat         <NA>      NA
## 6   6        1    5913 1473272920    chat         <NA>      NA
## 7   7        1    5913 1473272924 problem 008EG_COL_H1       1
## 8   8        1    6181 1473272926    chat         <NA>      NA
## 9   9        1    6181 1473272931 problem 008EG_COL_H1       1
## 10 10        1    5913 1473272937    chat         <NA>      NA
##                                                       content obs chunk_id
## 1                                    So how should we do this?   1        1
## 2  So I guess one of us should pick c and one should pick a?   2        1
## 3                                                          Yes   3        1
## 4                                                   Ill pick a   4        1
## 5                                                  I'll take a   5        1
## 6                                                       c then   6        1
## 7                                                     choice_2   7        1
## 8                                                          lol   8        1
## 9                                                     choice_0   9        1
## 10                                                         lol  10        2
##    obsnn obs_grp
## 1     NA       1
## 2     NA       2
## 3     NA       3
## 4     NA       4
## 5     NA       5
## 6     NA       6
## 7      1       7
## 8     NA       8
## 9      1       9
## 10    NA      10
```

merge two data file using the module name

```
prob_data$module_name<- gsub(".xml","",prob_data$L1) #remove ".xml"

module_name<- unique(chat_data$module)
module_name<- module_name[!is.na(module_name)]
chunk_id<- unique(chat_data$chunk_id)
chunk_id<- chunk_id[!is.na(chunk_id)]
df<- data.frame(chunk_id,module_name)

prob_data<- prob_data[!is.na(prob_data$value),]
prob_data<-aggregate(value ~ module_name , data = prob_data, toString) #concatenate all
 rows in one module

joined_data<- left_join(chat_data,df, by=c("chunk_id","chunk_id"))
joined_data<- left_join(joined_data,prob_data,by=c("module_name","module_name"))
```

```
## Warning: Column `module_name` joining factor and character vector, coercing
## into character vector
```

```
#create a column to combine the group id and module name, so we can tokenlize words by t
his index

joined_data$ind<- paste0("G",joined_data$group_id,"Q",joined_data$module_name)
```

STEMMING (don’t think this looks good)

```
#joined_data$stem_content<- wordStem(joined_data$content,language = "porter")
```

Tokenlize chat data by questions and group Wijk

```
TermByGroupQuestion<- joined_data %>%
  unnest_tokens(word, content) %>%
  count(ind,word,sort=TRUE) %>%
  filter(!word %in% stop_words$word) %>%  #remove stop_words
  ungroup
```

Tokenlize question data by question id

```
TermbyQuestion <- joined_data %>%
  unnest_tokens(word, value) %>%
  count(module_name,word, sort=TRUE) %>%
  filter(!word %in% stop_words$word) %>%
  ungroup
```

calculate TF-IDF

```
tot<- TermByGroupQuestion %>%
  group_by(ind) %>%
  summarize(total=sum(n))

TermByGroupQuestion<- left_join(TermByGroupQuestion, tot)
```

```
## Joining, by = "ind"
```

```
TermByGroupQuestion[,5] <- TermByGroupQuestion[,3]/TermByGroupQuestion[,4]
colnames(TermByGroupQuestion) <- c(colnames(TermByGroupQuestion)[1:4],"tf")



TermByModule<- joined_data %>%
  unnest_tokens(word, content) %>%
  count(module_name,word,sort=TRUE) %>%
  filter(!word %in% stop_words$word) %>%  #remove stop_words
  ungroup

TermByGroupQuestion$module_name<- unlist(strsplit(TermByGroupQuestion$ind,"Q"))[seq(2,2*
dim(TermByGroupQuestion)[1],2)]



idf <- rep(0,dim(TermByGroupQuestion)[1])
for (i in c(1:dim(TermByGroupQuestion)[1])){
  # no. of documents()
  wd <- as.character(TermByGroupQuestion[i,2])
  md <- as.character(TermByGroupQuestion[i,6])

  ## correcting for question words
  nd <- dim(TermByGroupQuestion[TermByGroupQuestion[,2]==wd & TermByGroupQuestion[,6]==m
d,])[1] + ifelse(dim(TermbyQuestion[TermbyQuestion[,1]==md & TermbyQuestion[,2]==wd,])[1
] > 0,length(unique(joined_data$group_id)),0)
  N <- dim(TermByGroupQuestion[TermByGroupQuestion[,6]==md,])[1] + ifelse(dim(TermbyQues
tion[TermbyQuestion[,1]==md & TermbyQuestion[,2]==wd,])[1] > 0,length(unique(joined_data
$group_id)),0)

  idf[i] <- -log(nd/N)
}
```

```
## corrected tf-idf
TermByGroupQuestion$idf <- idf
TermByGroupQuestion$tfidf <- TermByGroupQuestion$tf * TermByGroupQuestion$idf

TermByGroupQuestion_v1<-TermByGroupQuestion[order(TermByGroupQuestion$tfidf,decreasing =
 TRUE),]

## Remove numbers and Remove choices
TermByGroupQuestion_v1<-TermByGroupQuestion[is.na(as.numeric(TermByGroupQuestion$wor
d)),]
```

```
## Warning in `[.tbl_df`(TermByGroupQuestion,
## is.na(as.numeric(TermByGroupQuestion$word)), : NAs introduced by coercion
```

```
TermByGroupQuestion_v1<-TermByGroupQuestion_v1[!grepl("choice_",TermByGroupQuestion_v1$w
ord),]

#order it by the TF-IDF value
TermByGroupQuestion_v1<-TermByGroupQuestion_v1[order(TermByGroupQuestion_v1$tfidf,decrea
sing = TRUE),]

head(TermByGroupQuestion_v1,50)
```

```
## # A tibble: 50 x 8
##    ind                word         n total    tf module_name    idf tfidf
##    <chr>              <chr>    <int> <int> <dbl> <chr>        <dbl> <dbl>
##  1 G54Q008EG_COL_H1  patience     1     1 1.00  008EG_COL_H1 1.97  1.97
##  2 G120Q008EG_COL_H1 testing      1     1 1.00  008EG_COL_H1 1.96  1.96
##  3 G63Q008EG_COL_H1  day          1     1 1.00  008EG_COL_H1 1.93  1.93
##  4 G48Q023ED_COL_J2  triangle     1     1 1.00  023ED_COL_J2 1.93  1.93
##  5 G84Q023ED_COL_J2  helped       1     1 1.00  023ED_COL_J2 1.93  1.93
##  6 G101Q030EN_COL_M1 means        1     1 1.00  030EN_COL_M1 0.990 0.990
##  7 G62Q030EN_COL_M1  cuz          1     1 1.00  030EN_COL_M1 0.990 0.990
##  8 G76Q008EG_COL_H1  gl           1     2 0.500 008EG_COL_H1 1.97  0.984
##  9 G117Q008EG_COL_H1 glad         1     2 0.500 008EG_COL_H1 1.96  0.980
## 10 G76Q008EG_COL_H1  glad         1     2 0.500 008EG_COL_H1 1.96  0.980
## # ... with 40 more rows
```