Text Analysis 9 tf-idf

Junyan Yao

November 14, 2017

load the data

```
library(corpus)
library(Matrix)
library(tidytext)
library(ggplot2)
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
```

```
data<-read.csv("~/Documents/NYU/Fall 2017/Text Analysis Project/cpsv_text_project/chat_time_series.csv")
#data<- read.csv("C:/Users/jyao/Documents/Text Analysis/chat_time_series.csv") office comp
data<- data[,c(2,5,8)] #extract needed columns

#subset the data
chatdata<- data[which(data$type=="chat"),] #this is what we want to look at for now
problemdata<- data[which(data$type=="problem"),]


#load the outcome data
outcomedata<-read.csv("~/Documents/NYU/Fall 2017/Text Analysis Project/cpsv_text_project/group_outcomes.csv")
#outcomedata<-read.csv("C:/Users/jyao/Documents/Text Analysis/group_outcomes.csv") #office comp


subset1<- outcomedata[outcomedata$group_id>0,] #Get rid of all negative group_id
summary(subset1$delta) #we have 110 groups in this dataset
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.698000 -0.481000 -0.001394 -0.043190  0.405800  3.310000
```

```
performance<-ifelse(subset1$delta>0.4058,"high",ifelse(subset1$delta< -0.481,"low","in-b
etween"))
temp22<-cbind(subset1,performance) #label the performance for these groups

#try to get rid of the missing rows(some group id are missing in the outcome data)
merged_data<- merge(x=chatdata,y=temp22,by="group_id")

merged_data$text<-as.character(merged_data$content)

TermByGroup<- merged_data %>%
  unnest_tokens(word, text) %>%
  count(group_id,word, sort = TRUE) %>%
  ungroup()


tot<- TermByGroup %>%
  group_by(group_id) %>%
  summarize(total=sum(n))

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

```
## Joining, by = "group_id"
```

```
TermByGroup<- TermByGroup %>%
  bind_tf_idf(word,group_id, n)
head(TermByGroup)
```

```
## # A tibble: 6 × 7
##   group_id  word     n total         tf         idf       tf_idf
##      <int> <chr> <int> <int>      <dbl>       <dbl>        <dbl>
## 1      124   the    99  1648 0.06007282 0.056089467 0.0033694522
## 2       80   the    91  2314 0.03932584 0.056089467 0.0022057655
## 3       80     i    89  2314 0.03846154 0.009132484 0.0003512494
## 4       20   the    79  1260 0.06269841 0.056089467 0.0035167205
## 5      172   the    79  1676 0.04713604 0.056089467 0.0026438352
## 6      172    is    71  1676 0.04236277 0.037041272 0.0015691708
```

```
#sort it by TF_IDF value
temp<- TermByGroup %>%
  select(-total) %>%
  arrange(desc(tf_idf))

head(temp,30)
```

```
## # A tibble: 30 × 6
##    group_id       word     n         tf      idf    tf_idf
##       <int>      <chr> <int>      <dbl>    <dbl>     <dbl>
## 1        55        wiz     1 0.06250000 4.700480 0.2937800
## 2        86         tr     1 0.05263158 4.700480 0.2473937
## 3        94        try     3 0.16666667 1.062894 0.1771490
## 4        86 connection     1 0.05263158 3.314186 0.1744308
## 5        94  apologize     1 0.05555556 2.754570 0.1530317
## 6        55          s     1 0.06250000 2.397895 0.1498685
## 7        55       test     1 0.06250000 2.302585 0.1439116
## 8        55      tough     1 0.06250000 2.215574 0.1384734
## 9        86 everything     1 0.05263158 2.397895 0.1262050
## 10       94     please     1 0.05555556 2.135531 0.1186406
## # ... with 20 more rows
```

weight by performance

```
merged_data$text<- as.character(merged_data$content)
TermByGroup2<- merged_data %>%
  unnest_tokens(word, text) %>%
  count(performance,word, sort = TRUE) %>%
  ungroup()


tot<- TermByGroup2 %>%
  group_by(performance) %>%
  summarize(total=sum(n))

TermByGroup2<- left_join(TermByGroup2, tot)
```

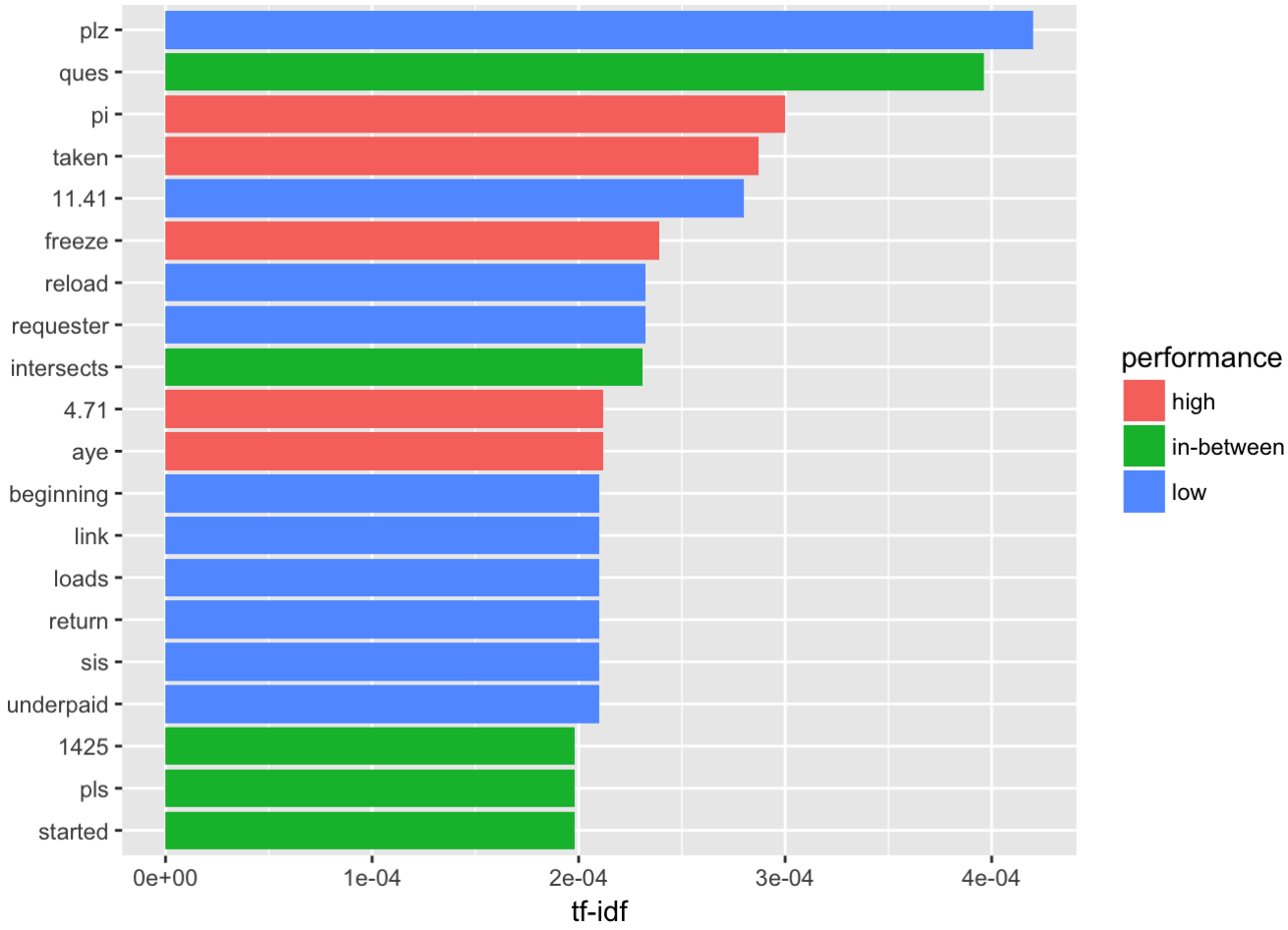
```
## Joining, by = "performance"
```

```
TermByGroup2<- TermByGroup2 %>%
  bind_tf_idf(word,performance, n)
#View(TermByGroup2)

temp2<- TermByGroup2 %>%
  select(-total) %>%
  arrange(desc(tf_idf))

head(temp2,20)
```

```
## # A tibble: 20 × 6
##    performance       word     n           tf       idf       tf_idf
##         <fctr>      <chr> <int>        <dbl>     <dbl>        <dbl>
## 1          low        plz     6 0.0003824336 1.0986123 0.0004201462
## 2   in-between       ques    12 0.0003607178 1.0986123 0.0003962890
## 3         high         pi    17 0.0007402891 0.4054651 0.0003001614
## 4         high      taken     6 0.0002612785 1.0986123 0.0002870438
## 5          low      11.41     4 0.0002549557 1.0986123 0.0002800975
## 6         high     freeze     5 0.0002177321 1.0986123 0.0002392032
## 7          low     reload     9 0.0005736503 0.4054651 0.0002325952
## 8          low  requester     9 0.0005736503 0.4054651 0.0002325952
## 9   in-between intersects     7 0.0002104187 1.0986123 0.0002311686
## 10        high       4.71    12 0.0005225570 0.4054651 0.0002118786
## 11        high        aye    12 0.0005225570 0.4054651 0.0002118786
## 12         low  beginning     3 0.0001912168 1.0986123 0.0002100731
## 13         low       link     3 0.0001912168 1.0986123 0.0002100731
## 14         low      loads     3 0.0001912168 1.0986123 0.0002100731
## 15         low     return     3 0.0001912168 1.0986123 0.0002100731
## 16         low        sis     3 0.0001912168 1.0986123 0.0002100731
## 17         low  underpaid     3 0.0001912168 1.0986123 0.0002100731
## 18  in-between       1425     6 0.0001803589 1.0986123 0.0001981445
## 19  in-between        pls     6 0.0001803589 1.0986123 0.0001981445
## 20  in-between    started     6 0.0001803589 1.0986123 0.0001981445
```

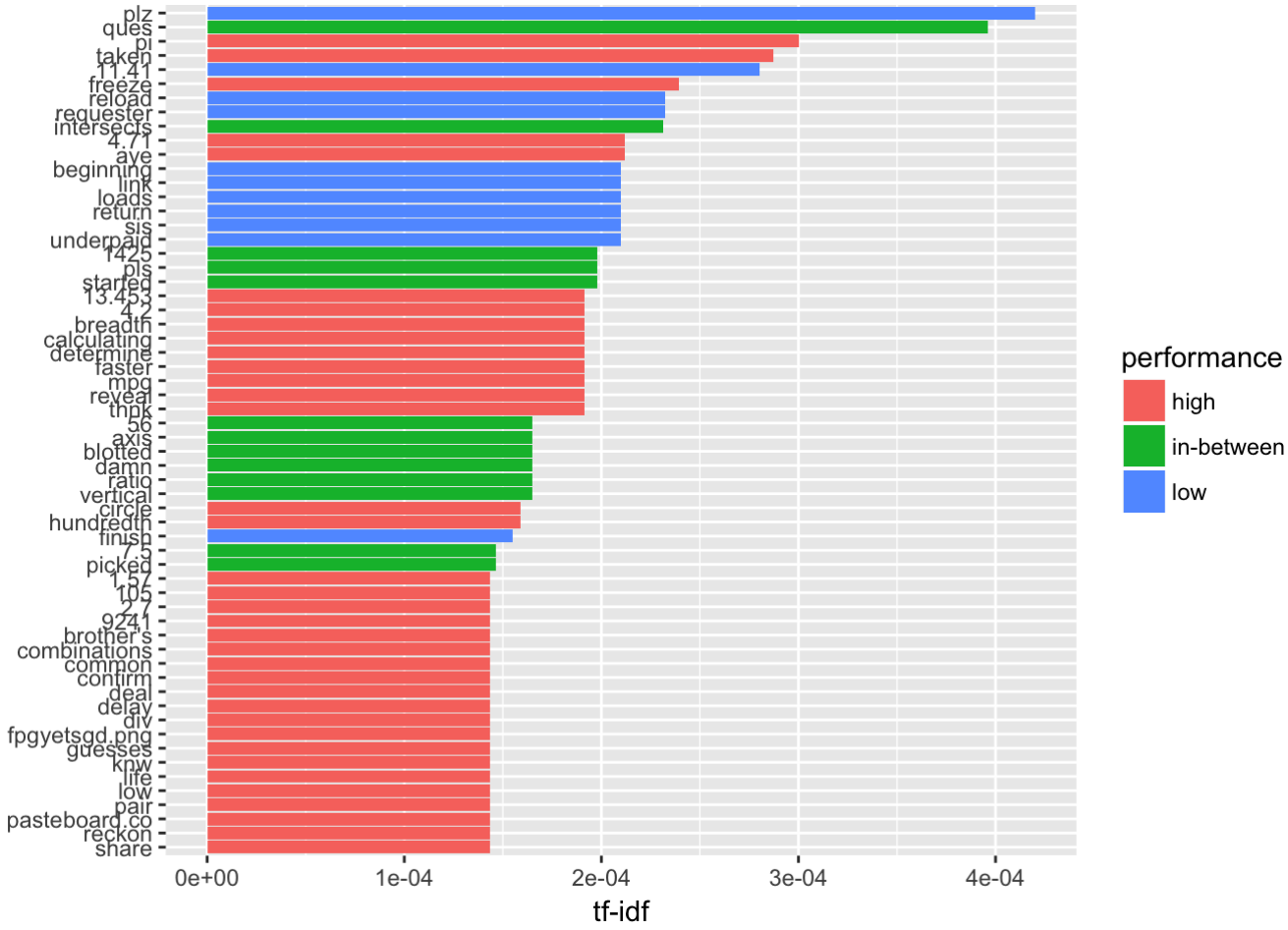
```
#visulize the high tf-idf
plot_idf<- TermByGroup2 %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels = rev(unique(word))))

plot_idf %>%
  top_n(20) %>%
  ggplot(aes(word, tf_idf, fill=performance)) +
  geom_col()+
  labs(x=NULL, y="tf-idf") +
  coord_flip()
```

```
## Selecting by tf_idf
```




```
#visulize the high tf-idf
plot_idf<- TermByGroup2 %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels = rev(unique(word))))

plot_idf %>%
  top_n(50) %>%
  ggplot(aes(word, tf_idf, fill=performance)) +
  geom_col()+
  labs(x=NULL, y="tf-idf") +
  coord_flip()
```

```
## Selecting by tf_idf
```



create TF_IDF Matrix

```
new<-unique(merged_data[,c(1,7)])
TermByGroup3<- left_join(TermByGroup,new, by="group_id")
TF_IDF_Matrix<- matrix(0, nrow=3408, ncol=110)
rownames(TF_IDF_Matrix)<- unique(TermByGroup3$word)
colnames(TF_IDF_Matrix)<- unique(TermByGroup3$group_id)
unique_groups <- unique(TermByGroup3$group_id)
for(i in c(1:length(unique_groups))){
  tf_idfs <- subset(TermByGroup3,TermByGroup3$group_id == unique_groups[i],select=c("wor
d","tf_idf"))
  word_rows <- which(rownames(TF_IDF_Matrix) %in% tf_idfs$word,arr.ind = TRUE)
  col_num <-  which(colnames(TF_IDF_Matrix) %in% unique_groups[i],arr.ind = TRUE)
  TF_IDF_Matrix[word_rows,col_num] <- tf_idfs$tf_idf
}

high_groups_cols <- which(colnames(TF_IDF_Matrix) %in% new$group_id[new$performance=="hi
gh"],arr.ind = TRUE)
low_groups_cols <- which(colnames(TF_IDF_Matrix) %in% new$group_id[new$performance=="lo
w"],arr.ind = TRUE)
```

chi-squared test

```
pvalue<- rep(NA, 3408)
for (i in 1:length(pvalue)){
  if(length(unique(TF_IDF_Matrix[i,low_groups_cols])) > 2 & length(unique(TF_IDF_Matrix[
i,high_groups_cols]))>2){
    pvalue[i]<-chisq.test(TF_IDF_Matrix[i,low_groups_cols],TF_IDF_Matrix[i,high_groups_c
ols],correct = TRUE)$p.value
  }else{
    pvalue[i] <- NA
  }
}

names(pvalue) <- rownames(TF_IDF_Matrix)
pvalue <- sort(pvalue)

sig<- pvalue[pvalue<0.05]

head(sig,50)
```

```
##        minute            32          sold        though         begin
##  2.910962e-10  2.910962e-10  1.135738e-09  2.377689e-09  2.820845e-09
##            68         their            n2 collaboration         sound
##  1.305456e-08  1.305456e-08  1.385281e-08  1.385281e-08  2.052908e-08
##         those           box          also        equals         check
##  2.460499e-08  1.117055e-07  1.123523e-07  1.151551e-07  1.151551e-07
##        change          fast       algebra           hmm         sense
##  1.151551e-07  1.151551e-07  1.196952e-07  1.543613e-07  1.543613e-07
##        little       cookies           6.5      negative           man
##  1.543613e-07  1.138194e-06  1.144827e-06  1.144827e-06  1.144827e-06
##        divide           990         slope           3.5      shipment
##  1.144827e-06  1.186892e-06  1.383903e-06  2.260011e-06  2.263363e-06
##        chosen          want           yet            ab     batteries
##  2.433572e-06  6.422970e-06  8.276189e-06  9.329749e-06  1.057568e-05
##         above        stupid          lose          else          such
##  1.223414e-05  1.223414e-05  1.223414e-05  1.223414e-05  1.223414e-05
##       whisker      educated           all            x2      selected
##  1.223414e-05  1.223414e-05  2.185688e-05  2.615476e-05  2.615476e-05
##           2.5        person        figure            us         makes
##  2.615476e-05  2.615476e-05  2.953041e-05  4.365505e-05  4.365505e-05
```