Basic Text-An funcs in Tidytext

Sudhir Voleti

Hi all,

First things first. Here is the setup code chunk. If any of the libraries are missing in your system, well, you know how to install them.

```
if (!require(tm)) {install.packages("tm")}
```

```
## Loading required package: tm

## Loading required package: NLP
```

```
if (!require(wordcloud)) {install.packages("wordcloud")}
```

```
## Loading required package: wordcloud

## Loading required package: RColorBrewer
```

```
if (!require(igraph)) {install.packages("igraph")}
```

```
## Loading required package: igraph

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
if (!require(ggraph)) {install.packages("ggraph")}
```

```
## Loading required package: ggraph

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```
if (!require(tidytext)) {install.packages("tidytext")}
```

```
## Loading required package: tidytext
```

```
if (!require(widyr)) {install.packages("widyr")}
```

```
## Loading required package: widyr
```

```
library(tm)
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------------
## v tibble  2.1.1     v purrr   0.3.2
## v tidyr   0.8.3     v dplyr   0.8.1
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.1     v forcats 0.4.0
```

```
## -- Conflicts ----------------------------------------------------------------------
## x ggplot2::annotate()    masks NLP::annotate()
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()       masks igraph::compose()
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()        masks stats::filter()
## x dplyr::groups()        masks igraph::groups()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()
```

```
library(tidytext)
library(wordcloud)
library(igraph)
library(ggraph)
```

## == Read in a real dataset (IBM) for basic text-an ==

Below is a transcript of IBM Q3's 2016 analyst call. What kind of context would you expect to see in an analyst call report?

Can we quickly text-an the same and figure out what the content is saying?

```
## reading in IBM analyst call data from my git
ibm = readLines('swiggy.txt')  #IBM Q3 2016 analyst call transcript
# ibm = readLines(file.choose())  # read from local file on disk
head(ibm, 5)    # view a few lines
```

```
## [1] "craved ice creame scorching afternoon ordered scoops delivery boy"
## [2] "hi team way possible include group conversation among friends list feedback best dishes offered
## [3] "start service morning breakfast aligarh uttar pradesh many famous shops kachauri samaosa etc"
## [4] "hunger takes time essence top restaurants go zero full flash"
## [5] "telling would compensations na ill stop using make atleast people stop using called app paid to
```

We saw how to build DTMs. Let us functionize that code in general terms so that we can repeatedly invoke the func where required.

```
dtm_build <- function(raw_corpus, tfidf=FALSE)
{                   # func opens

require(tidytext); require(tibble); require(tidyverse)

# converting raw corpus to tibble to tidy DF
textdf = data_frame(text = raw_corpus);    textdf

tidy_df = textdf %>%
                   mutate(doc = row_number()) %>%
                   unnest_tokens(word, text) %>%
                   anti_join(stop_words) %>%
                   group_by(doc) %>%
                   count(word, sort=TRUE)
tidy_df

# evaluating IDF wala DTM
if (tfidf == "TRUE") {
   textdf1 = tidy_df %>%
 group_by(doc) %>%
 count(word, sort=TRUE) %>% ungroup() %>%
 bind_tf_idf(doc, word, n) %>% # 'nn' is default colm name
 rename(value = tf_idf)} else { textdf1 = tidy_df %>% rename(value = n)  }

textdf1

dtm = textdf1 %>% cast_sparse(doc, word, value);    dtm[1:9, 1:9]

# order rows and colms putting max mass on the top-left corner of the DTM
colsum = apply(dtm, 2, sum)
col.order = order(colsum, decreasing=TRUE)
row.order = order(rownames(dtm) %>% as.numeric())

dtm1 = dtm[row.order, col.order];    dtm1[1:8,1:8]

return(dtm1)  }   # func ends

# testing func 2 on ibm data
system.time({ dtm_ibm_tf = dtm_build(ibm) })    # 0.02 secs
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
## Joining, by = "word"
```

```
##    user  system elapsed
##   0.656   0.127   0.784
```

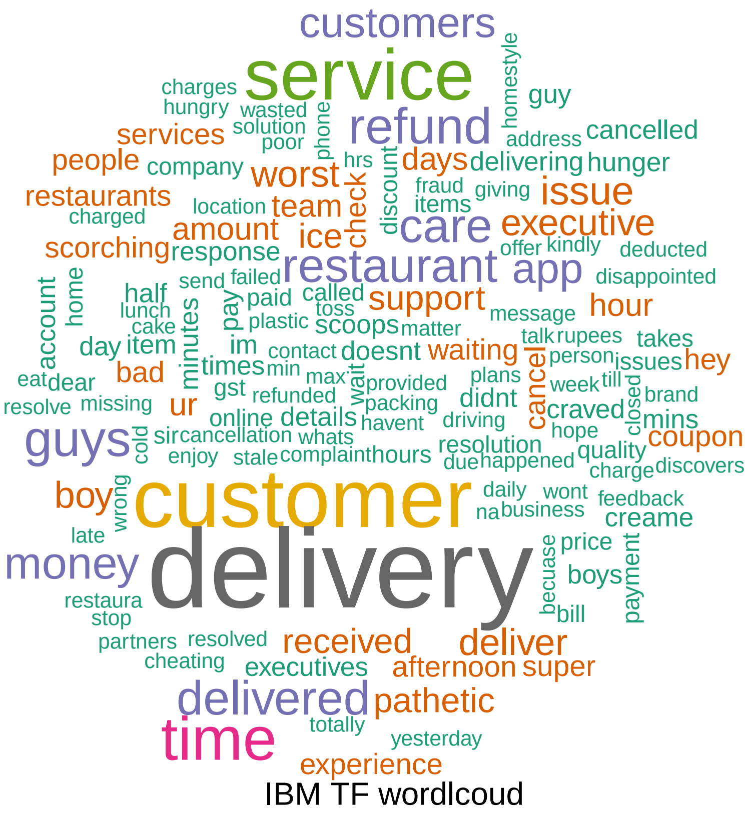
```
system.time({ dtm_ibm_idf = dtm_build(ibm, tfidf=TRUE) })  # 0.05 secs
```

```
## Joining, by = "word"
```

```
##    user  system elapsed
##   0.543   0.094   0.637
```

Func 3: wordcloud building

```
build_wordcloud <- function(dtm,
                               max.words1=150,     # max no. of words to accommodate
                               min.freq=5,       # min.freq of words to consider
                               plot.title="wordcloud"){          # write within double quotes

require(wordcloud)
if (ncol(dtm) > 20000){   # if dtm is overly large, break into chunks and solve

tst = round(ncol(dtm)/100)  # divide DTM's cols into 100 manageble parts
a = rep(tst,99)
b = cumsum(a);rm(a)
b = c(0,b,ncol(dtm))

ss.col = c(NULL)
for (i in 1:(length(b)-1)) {
 tempdtm = dtm[,(b[i]+1):(b[i+1])]
 s = colSums(as.matrix(tempdtm))
 ss.col = c(ss.col,s)
 print(i)      } # i loop ends

tsum = ss.col

} else { tsum = apply(dtm, 2, sum) }

tsum = tsum[order(tsum, decreasing = T)]       # terms in decreasing order of freq
head(tsum);    tail(tsum)

# windows()  # Opens a new plot window when active
wordcloud(names(tsum), tsum,     # words, their freqs
         scale = c(3.5, 0.5),     # range of word sizes
         min.freq,                     # min.freq of words to consider
         max.words = max.words1,       # max #words
         colors = brewer.pal(8, "Dark2"))    # Plot results in a word cloud
title(sub = plot.title)     # title for the wordcloud display

   } # func ends

# test-driving func 3 via IBM data
system.time({ build_wordcloud(dtm_ibm_tf, plot.title="IBM TF wordlcoud") })    # 0.4 secs
```



```
##    user  system elapsed
##   0.691   0.091   0.786
```

And now, test driving the IDF one…

```
system.time({ build_wordcloud(dtm_ibm_idf, plot.title="IBM IDF wordlcoud", min.freq=2) })    # 0.09 se
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : svavgzt could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : automation could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : vizag could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : sorryi could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : provoking could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : xhhcvtvee could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : abschattet could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : kharekhar could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : shodddy could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hdhdhd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : fghhs could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : gfhdh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdjdh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : piggy could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : mam could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : awww could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hrjjr could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhsheh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ahahahahahahahahahaha could not be fit on page.
## It will not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : swiggi could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : chill could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : lintas could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : rajmachawal could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : trippin could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ghdhhd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hushe could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : venkatesh could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : zbhctve could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : skdjctrv could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : thankyou could not be fit on page. It will not be
## plotted.
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : raspans could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ghdye could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : quitzomato could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : greedy could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : palais could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : sjhwyywtts could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhshdh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdhrh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : references could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : svvssghs could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdhdh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : dggff could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : staple could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : reminder could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : sankarannas could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hjsje could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : repl could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : texted could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : anzuraten could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : wooooo could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : sooo could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ah could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdje could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : bchxhhd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ignoring could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ghdhdh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : watching could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hdjdj could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : schafsfell could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : dhfgf could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : condone could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : rescue could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : planned could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : koisa could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : dday could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : newapplaunch could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : brownie could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : sricharan could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : morons could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : abduct could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : inputs could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : tyshd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : cheaper could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : crooks could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ghrjjd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : whatsover could not be fit on page. It will not
## be plotted.
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ghxjdh could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : curry could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : royal could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : alaoeur could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : rocks could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hellos could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : ground could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : whorst could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hdjjd could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhshd could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : lowe could not be fit on page. It will not be
## plotted.
```

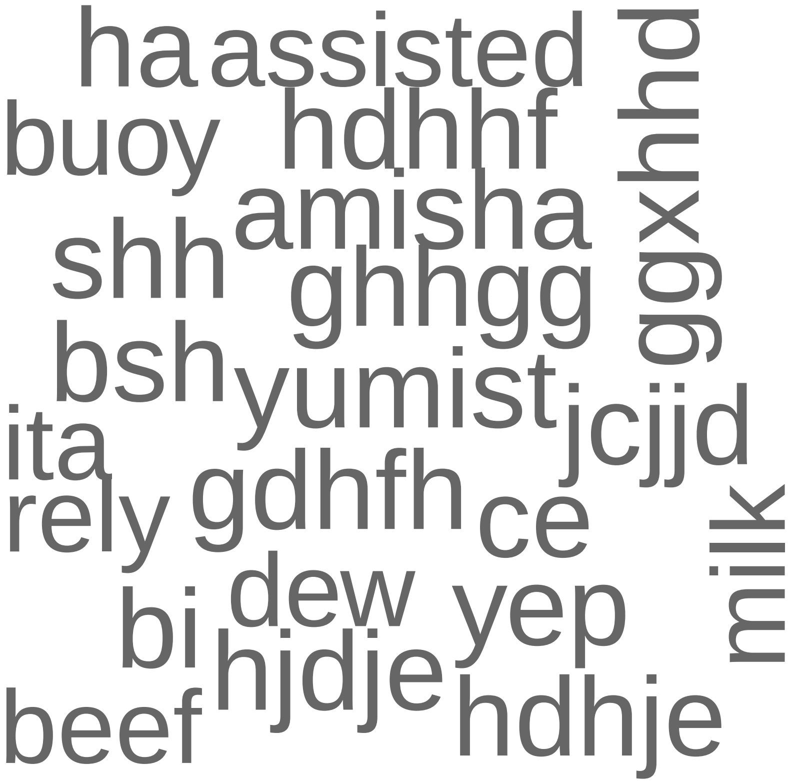
```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : jejje could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : clearing could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : suman could not be fit on page. It will not be
## plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : angewandt could not be fit on page. It will not
## be plotted.
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : swigg could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : actionit could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : chffd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhxjxh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : dissatisfied could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : raipur could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : promotion could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhhsh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : savage could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdjd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : satuafactory could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : godbig could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : aaa could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : art could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : foot could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : swiggyappraisals could not be fit on page. It
## will not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : corruptswiggy could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : gydhey could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : huy could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : bhhdhd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : gaststtte could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : safedrivelonglife could not be fit on page. It
## will not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : robbery could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : inspiration could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : okkk could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : pull could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : remaining could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : slower could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hdj could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : swiggyfood could not be fit on page. It will not
## be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hdghdh could not be fit on page. It will not be
## plotted.
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdheh could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhdhhd could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : nonsenseswiggy could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : brother could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : broman could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : hhfhf could not be fit on page. It will not be
## plotted.
```



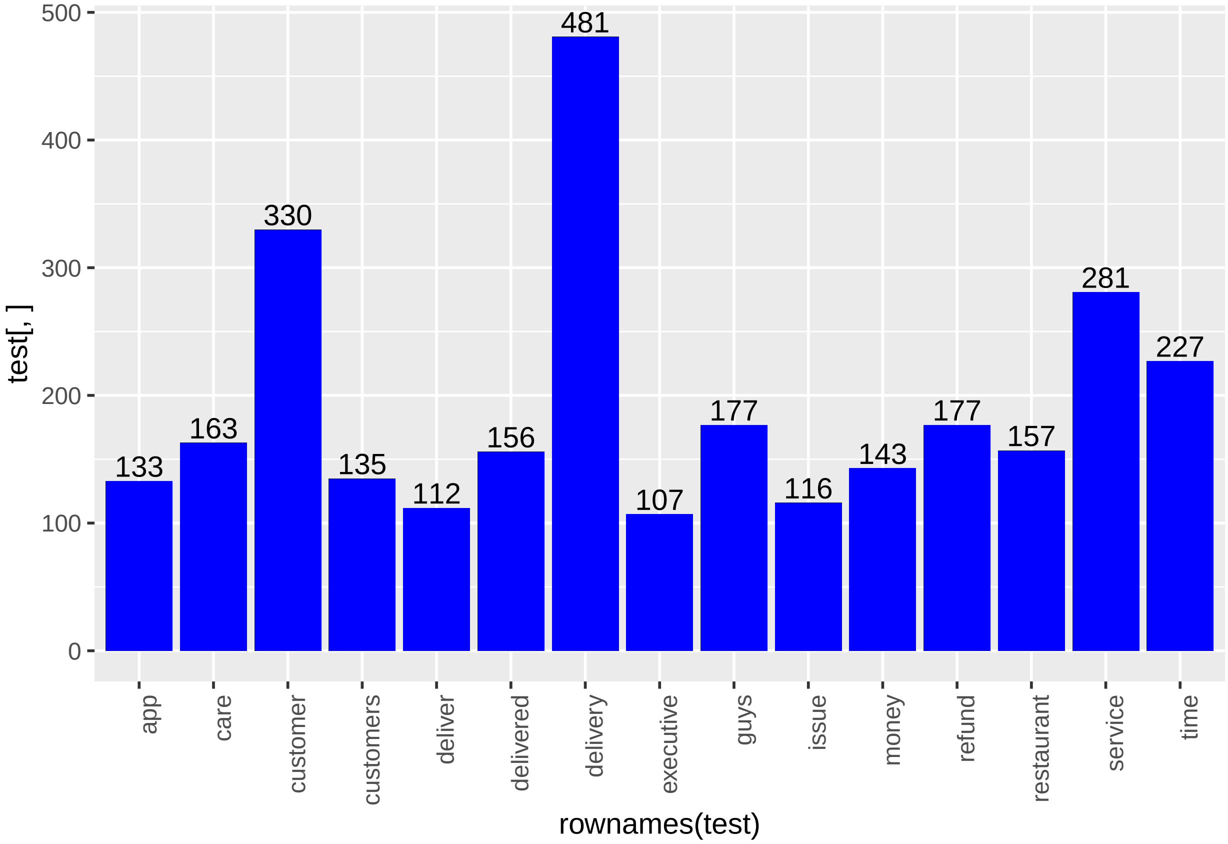
IBM IDF wordlcoud

```
##    user  system elapsed
##   0.966   0.068   1.127
```

Func 4: Simple Bar.charts of top tokens

Self-explanatory. And simple. But just for completeness sake, making a func out of it.

```
plot.barchart <- function(dtm, num_tokens=15, fill_color="Blue")
{
a0 = apply(dtm, 2, sum)
a1 = order(a0, decreasing = TRUE)
tsum = a0[a1]

# plot barchart for top tokens
test = as.data.frame(round(tsum[1:num_tokens],0))


# windows()  # New plot window
require(ggplot2)
p = ggplot(test, aes(x = rownames(test), y = test[,])) +
      geom_bar(stat = "identity", fill = fill_color) +
      geom_text(aes(label = test[,]), vjust= -0.20) +
      theme(axis.text.x = element_text(angle = 90, hjust = 1))

plot(p) }  # func ends

# testing above func
system.time({ plot.barchart(dtm_ibm_tf) })    # 0.1 secs
```



```
##    user  system elapsed
##   0.474   0.029   0.506
```

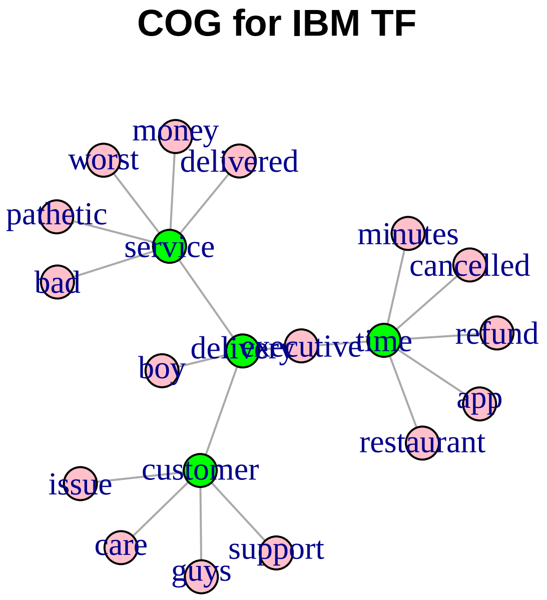
```
# system.time({ plot.barchart(dtm_ibm_idf, num_tokens=12, fill_color="Red") })    # 0.11 secs
```

Func 5: Co-occurrence graphs (COGs)

COGs as the name suggests connects those tokens together that most co-occur within documents, using a network graph wherein the nodes are tokens of interest.

This is admittedly a slightly long-winded func. Also introduces network visualization concepts. If you're unfamiliar with this, pls execute the func's content line-by-line to see what each line does.

```
distill.cog = function(dtm, # input dtm
                       title="COG", # title for the graph
                       central.nodes=4,    # no. of central nodes
                       max.connexns = 5){  # max no. of connections

# first convert dtm to an adjacency matrix
dtm1 = as.matrix(dtm)   # need it as a regular matrix for matrix ops like %*% to apply
adj.mat = t(dtm1) %*% dtm1    # making a square symmatric term-term matrix
diag(adj.mat) = 0     # no self-references. So diag is 0.
a0 = order(apply(adj.mat, 2, sum), decreasing = T)   # order cols by descending colSum
mat1 = as.matrix(adj.mat[a0[1:50], a0[1:50]])

 # now invoke network plotting lib igraph
 library(igraph)

 a = colSums(mat1) # collect colsums into a vector obj a
 b = order(-a)     # nice syntax for ordering vector in decr order

 mat2 = mat1[b, b]      # order both rows and columns along vector b
 diag(mat2) =  0

 ## +++ go row by row and find top k adjacencies +++ ##

 wc = NULL

 for (i1 in 1:central.nodes){
   thresh1 = mat2[i1,][order(-mat2[i1, ])[max.connexns]]
   mat2[i1, mat2[i1,] < thresh1] = 0   # neat. didn't need 2 use () in the subset here.
   mat2[i1, mat2[i1,] > 0 ] = 1
   word = names(mat2[i1, mat2[i1,] > 0])
   mat2[(i1+1):nrow(mat2), match(word,colnames(mat2))] = 0
   wc = c(wc, word)
 } # i1 loop ends


 mat3 = mat2[match(wc, colnames(mat2)), match(wc, colnames(mat2))]
 ord = colnames(mat2)[which(!is.na(match(colnames(mat2), colnames(mat3))))]  # removed any NAs from th
 mat4 = mat3[match(ord, colnames(mat3)), match(ord, colnames(mat3))]

 # building and plotting a network object
 graph <- graph.adjacency(mat4, mode = "undirected", weighted=T)    # Create Network object
 graph = simplify(graph)
 V(graph)$color[1:central.nodes] = "green"
 V(graph)$color[(central.nodes+1):length(V(graph))] = "pink"

 graph = delete.vertices(graph, V(graph)[ degree(graph) == 0 ]) # delete singletons?

 plot(graph,
      layout = layout.kamada.kawai,
      main = title)

 } # distill.cog func ends

# testing COG on ibm data
system.time({ distill.cog(dtm_ibm_tf, "COG for IBM TF") })    # 0.27 secs
```



COG for IBM TF

```
##    user  system elapsed
##  49.124   0.461  49.683
```

```
# system.time({ distill.cog(dtm_ibm_idf, "COG for IBM IDF", 5, 5) })    # 0.57 secs
```

Func 6 - wordcloud + COG combo

Both the 2 major display aids we saw thus far - cog and wordcloud - have their pros and cons. Can we somehow combine them and get the best of both worlds, so to say? Read on.

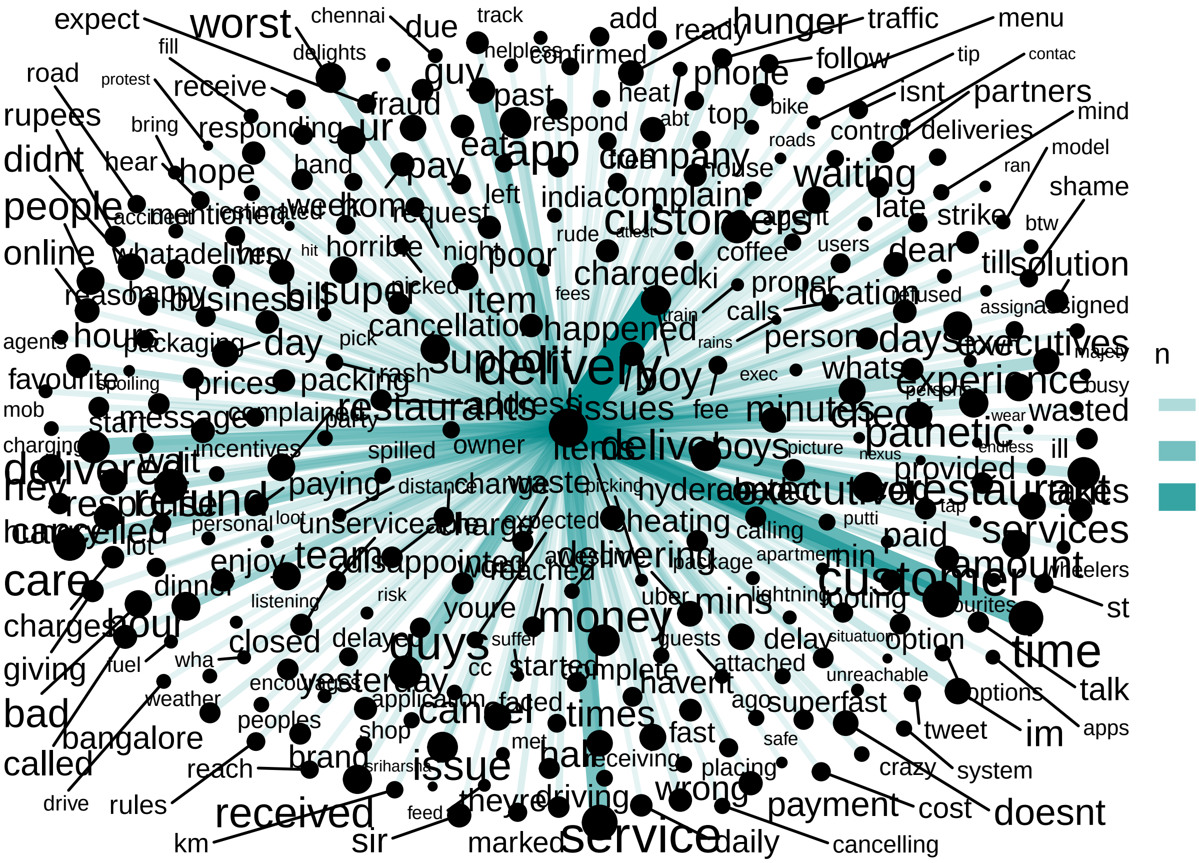
```
build_cog_ggraph <- function(corpus,   # text colmn only
                            max_edges = 150,
                            drop.stop_words=TRUE,
                            new.stopwords=NULL){

 # invoke libraries
 library(tidyverse)
 library(tidytext)
 library(widyr)
 library(ggraph)

 # build df from corpus
 corpus_df = data.frame(docID = seq(1:length(corpus)), text = corpus, stringsAsFactors=FALSE)

 # eval stopwords condn
 if (drop.stop_words == TRUE) {stop.words = unique(c(stop_words$word, new.stopwords)) %>%
   as_tibble() %>% rename(word=value)} else {stop.words = stop_words[2,]}

 # build word-pairs
 tokens <- corpus_df %>%

   # tokenize, drop stop_words etc
   unnest_tokens(word, text) %>% anti_join(stop.words)

   # pairwise_count() counts #token-pairs co-occuring in docs
 word_pairs = tokens %>% pairwise_count(word, docID, sort = TRUE, upper = FALSE)# %>% # head()

 word_counts = tokens %>% count( word,sort = T) %>% dplyr::rename( wordfr = n)

 word_pairs = word_pairs %>% left_join(word_counts, by = c("item1" = "word"))

 row_thresh = min(nrow(word_pairs), max_edges)

 # now plot
 set.seed(1234)
 # windows()
 plot_d <- word_pairs %>%
   filter(n >= 3) %>%
   top_n(row_thresh) %>%   igraph::graph_from_data_frame()

 dfwordcloud = data_frame(vertices = names(V(plot_d))) %>% left_join(word_counts, by = c("vertices"= "

 plot_obj = plot_d %>%   # graph object built!

   ggraph(layout = "fr") +
   geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "cyan4")  +
   # geom_node_point(size = 5) +
   geom_node_point(size = log(dfwordcloud$wordfr)) +
   geom_node_text(aes(label = name), repel = TRUE,
                  point.padding = unit(0.2, "lines"),
                  size = 1 + log(dfwordcloud$wordfr)) +
   theme_void()

 return(plot_obj)    # must return func output

}  # func ends

# quick example for above using amazon nokia corpus
nokia = readLines('swiggy.txt')
system.time({ b0=build_cog_ggraph(nokia) })    # 0.36 secs
```

```
## Warning: Calling `as_tibble()` on a vector is discouraged, because the behavior is likely to change
## This warning is displayed once per session.
```

```
## Joining, by = "word"
```

```
## Selecting by wordfr
##    user  system elapsed
##   0.281   0.008   0.293
```

```
b0
```



Clearly, the most frequently occuring token above has taken epi-central node status. What if we dropped it? What new patterns might emerge? Points to ponder…

Well, that's it for now. I'm sure I have run out of time. If so, will pickup in the next session from where we leave off.

Sudhir