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('zomato.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] "worst experience able cancely last hour didnt assign delivery boy serious"
## [2] ""
## [3] "wanted two delivery charges bill hidden taxesrs"
## [4] "cant believe thisyou people selling cold drink mrp exact double pricei ordered bottles price rs
## [5] "happened mate lost cat caught tongue"
```

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
##   1.275   0.426   1.786
```

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

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

```
##    user  system elapsed
##   0.989   0.358   1.352
```

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
```

```
## Warning in wordcloud(names(tsum), tsum, scale = c(3.5, 0.5), min.freq,
## max.words = max.words1, : money 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, : support 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, : service 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, : delivery 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, : person 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, : charged 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, : chicken 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, : leave 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, : response 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, : time 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, : cancelled could not be fit on page. It will not
## be plotted.
```

IBM TF wordlcoud

```
##    user  system elapsed
##   1.200   0.410   1.613
```

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, : islamabad 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, : obligation 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, : investigate 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, : jlokhandecom 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, : superrrrrrrr 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, : asshole 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, : damage 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, : deep 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, : shrewsbury 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, : bhakarwadi 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, : musketeers 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, : panda 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, : namoone 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, : hahahahahaha 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, : nothingits 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, : iceland 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, : misal 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, : sujeetscom 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, : caring 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, : nos 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, : tunday 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, : bhaturaas 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, : zomatoisscam 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, : faraa 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, : discussing 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, : corrected 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, : identify 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, : dabbulochesay 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, : yepp 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, : hebbars 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, : weak 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, : stopzomatostartswiggy 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, : ashwini 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, : manuintouchcom 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, : lapse 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, : recipe 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, : salivating 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, : ramdarweshfoodcom 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, : ordee 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, : consequences 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, : eh 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, : hscom 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, : assam 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, : thts 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, : fun 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, : sticker 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, : yayy 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, : epic 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, : chiken 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, : likes 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, : rajasthani 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, : happn 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, : awaited 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, : threesome 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, : profmohantycom 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, : pakistan 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, : rofl 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, : dmed 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, : buttha 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, : bastards 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, : chattisgarh 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, : finger 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, : absurd 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, : khoya 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, : testing 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, : bikanervala 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, : dieting 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, : dosmh 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, : zomatobad 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, : stole 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, : badshaaah 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, : highlights 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, : suggesting 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, : idliboobs 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, : amethi 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, : revealed 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, : someday 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, : bhia 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, : thiz 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, : nunber 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, : balme 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, : somalia 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, : chamcham 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, : intestines 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, : oreder 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, : stokes 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, : apoomalpecom 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, : suparb 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, : dubai 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, : peesy 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, : quarry 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, : shaken 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, : dmlink 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, : proofofincompetence 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, : exchanged 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, : keema 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, : hmm 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, : lifeline 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, : responce 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, : savarkar 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, : acceptanceorder 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, : karwaabhishekcom 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, : ist 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, : yeahhh 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, : heavenly 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, : breed 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, : bears 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, : barra 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, : hahahahaha 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, : buddy 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, : ashutoshkvatcom 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, : yum 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, : othershame 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, : speachless 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, : farah 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, : madarchod 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, : wars 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, : africa 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, : damid 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, : chhattisgarh 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, : yep 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, : ducks 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, : dmd 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, : fyr 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, : matcha could not be fit on page. It will not be
## plotted.
```

IBM IDF wordlcoud

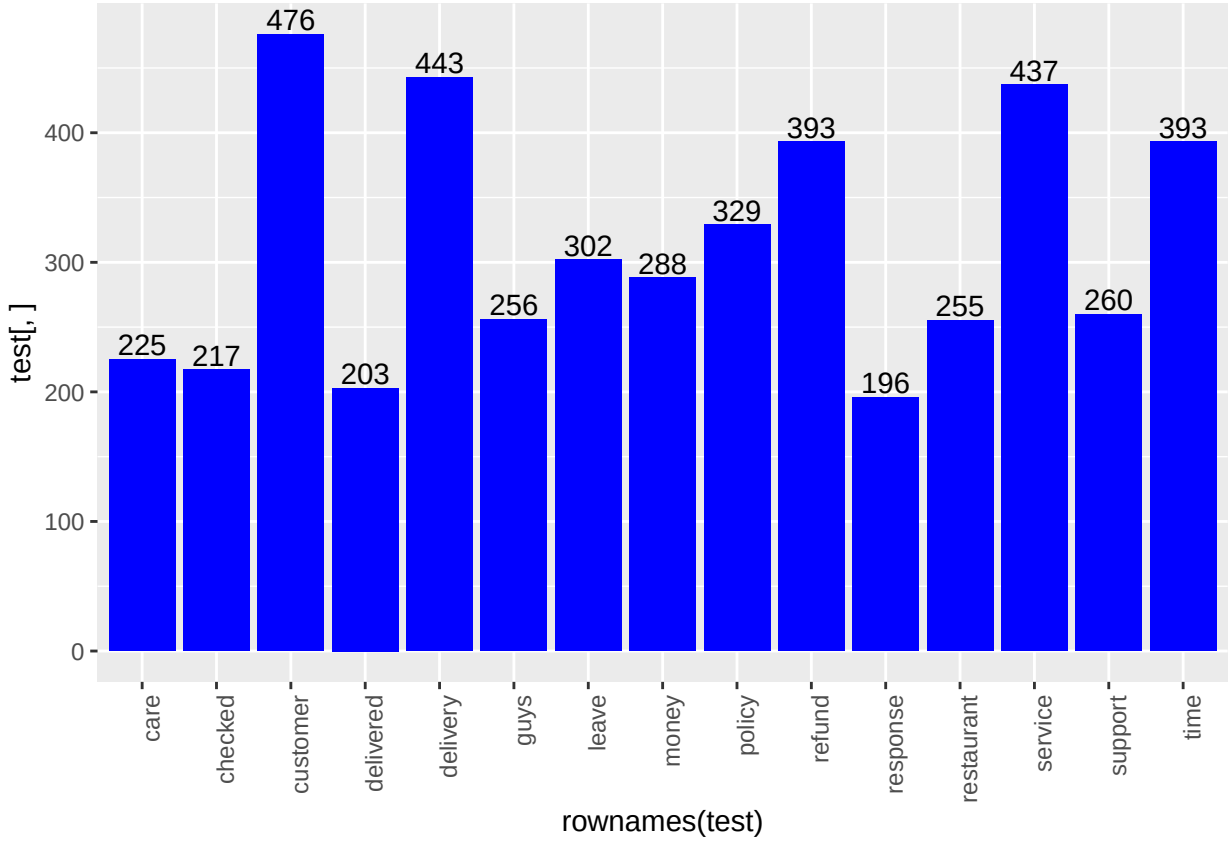
```
##    user  system elapsed
##   1.353   0.397   1.754
```

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
##   0.900   0.361   1.289
```

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

Func 5: Co-occurrence graphs (COGs)

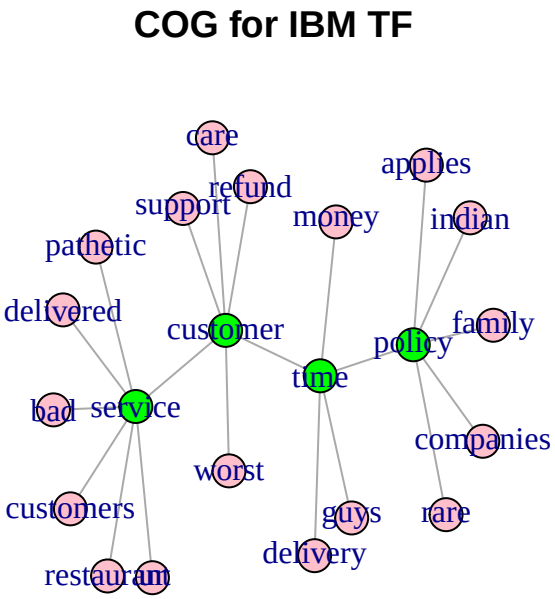
COGs as the name suggests connects those tokens together that most co-occur within documents, using a network graph wherein the nodes are tokens of interest.

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
```

```
## Warning in vattrs[[name]][index] <- value: number of items to replace is
## not a multiple of replacement length
```




```
##    user  system elapsed
## 129.511   0.940 131.019
```

```
# 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('zomato.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 i
## This warning is displayed once per session.
```

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

```
## Selecting by wordfr
```

```
##    user  system elapsed
##   0.395   0.012   0.412
```

```
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