# Text mining in practice - Bag of Words - Intro to word networks

Digital Humanities DSC 105 Spring 2023

Marcus Birkenkrahe

March 30, 2023

## README

Time-stamp: *<2023-03-30 Thu 08:51>*

- This lecture closely follows the DataCamp lesson "Text Mining with Bag-of-Words in R" by Ted Kwartler, chapter 2, lesson 3, "Other word clouds and word nets" (Link).
- Download and open the practice file `9_wordnets_practice.org` from GitHub to code along.
- In this lecture & practice:
  1. Finding and visualizing common words
  2. Creating a polarized tag cloud and pyramid plots
  3. Visualize word networks
  4. Visualizing word clusters as dendograms

## Get the corpus data and the R packages

- If you didn't do it yet: download `corpora.R` from GitHub (bit.ly/tm-corpora)
- Run the file on the shell (`M-x eshell`) as a batch job:

```
R CMD BATCH corpora.R
ls -al .RData
```

- Load the `.RData` file in your current R session and check that packages and user-defined objects were loaded:

```
load_packages <- function() {
  library(tm)
  library(qdap)
  library(SnowballC)
  library(wordcloud)
  search()
}
load_packages()
load("c:/Users/birkenkrahe/Downloads/.RData")
search()
ls()
```

```
 [1] ".GlobalEnv"               "package:plotrix"
 [3] "package:dplyr"            "package:viridisLite"
 [5] "package:wordcloud"        "package:SnowballC"
 [7] "package:qdap"             "package:RColorBrewer"
 [9] "package:qdapTools"        "package:qdapRegex"
[11] "package:qdapDictionaries" "package:tm"
[13] "package:NLP"              "ESSR"
[15] "package:stats"            "package:graphics"
[17] "package:grDevices"        "package:utils"
[19] "package:datasets"         "package:stringr"
[21] "package:httr"             "package:methods"
[23] "Autoloads"                "package:base"
 [1] ".GlobalEnv"               "package:plotrix"
 [3] "package:dplyr"            "package:viridisLite"
 [5] "package:wordcloud"        "package:SnowballC"
 [7] "package:qdap"             "package:RColorBrewer"
 [9] "package:qdapTools"        "package:qdapRegex"
[11] "package:qdapDictionaries" "package:tm"
[13] "package:NLP"              "ESSR"
[15] "package:stats"            "package:graphics"
[17] "package:grDevices"        "package:utils"
[19] "package:datasets"         "package:stringr"
[21] "package:httr"             "package:methods"
[23] "Autoloads"                "package:base"
 [1] "all_chardonnay"          "all_clean"
```

```
 [3] "all_coffee"              "all_corpus"
 [5] "all_m"                   "all_tdm"
 [7] "all_tweets"              "api_key"
 [9] "ask_chatgpt"             "chardonnay_corpus"
[11] "chardonnay_df"           "chardonnay_m"
[13] "chardonnay_src"          "chardonnay_tdm"
[15] "chardonnay_vec"          "clean_chardonnay"
[17] "clean_chardonnay_corpus" "clean_coffee"
[19] "clean_coffee_corpus"     "clean_corpus"
[21] "coffee_corpus"           "coffee_df"
[23] "coffee_m"                "coffee_src"
[25] "coffee_tdm"              "coffee_vec"
[27] "color_pal"               "idx"
[29] "load_packages"           "m"
[31] "M"                       "max"
[33] "stops"                   "term_frequency"
[35] "terms"                   "terms_vec"
[37] "top25_df"                "word_freq"
```

- You need the `clean_coffee` and `clean_chardonnay` corpora.
- If we don't finish with a session, save your data from now on:

```
save.image(file=".RData")
shell("ls -al .RData")
```

```
-rwx------+ 1 Birkenkrahe LYONNET+Group(513) 1105229 Mar 30 11:30 .RData
```

## The workflow

We're looking at two corpora at a time to find out which words they have in common (the intersection), and which words they do not have in common (the disjoint). To do this, we must:

1. Paste datasets
2. Collapse datasets into one
3. Make clean corpus
4. Make TDM

5. Make term matrix
6. Visualize term matrix

## Common and disjoint word sets

- Think of your corpora as sets and visualize them in Venn diagrams

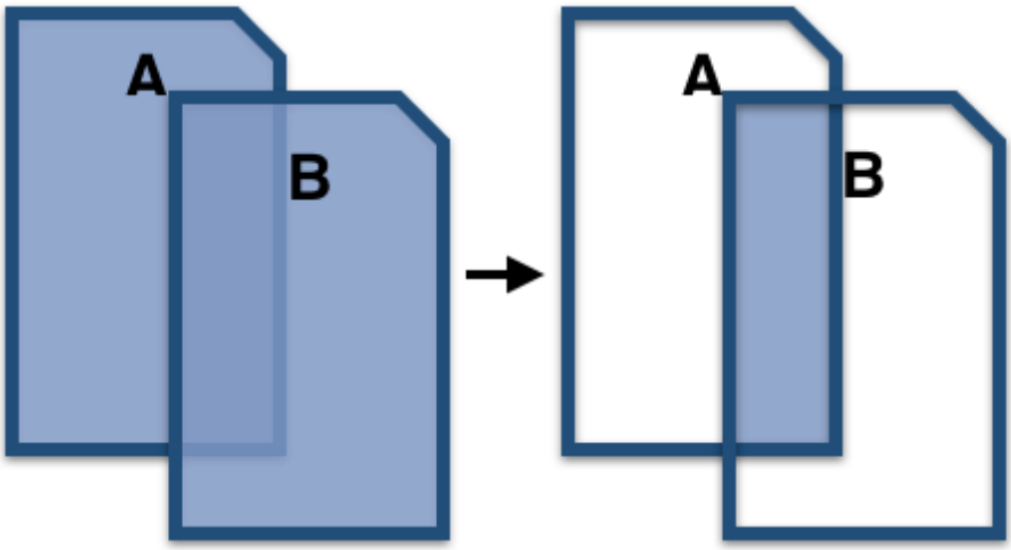



Figure 1: Visualizing common words as Venn diagram

## Find common words

- We're going to use `wordcloud::commonality.cloud`: the function requires a TDM of the terms from both datasets:

```
args(commonality.cloud)
```

```
function (term.matrix, comonality.measure = min, max.words = 300,
    ...)
NULL
```

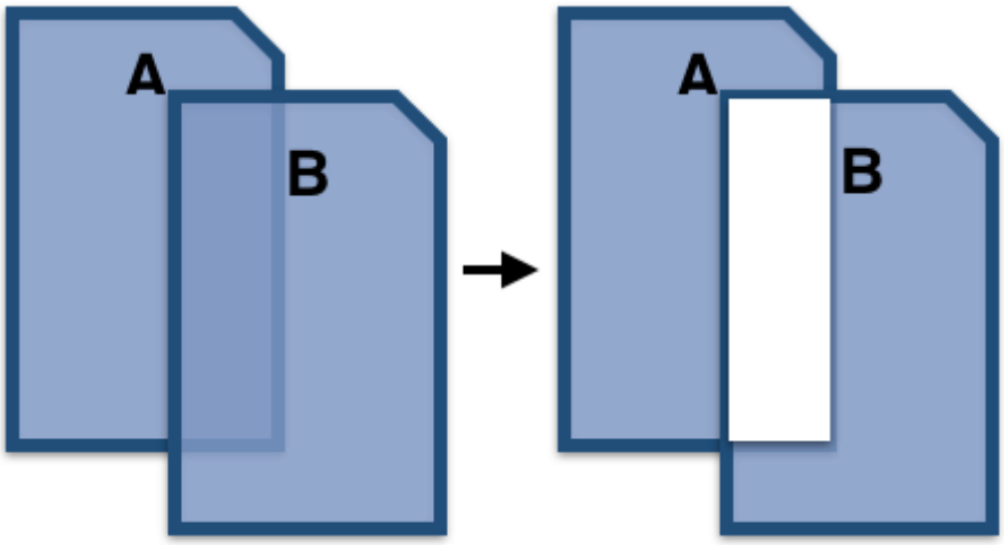



Figure 2: Visualizing comparison of disjoint words as Venn diagram

- Use `paste` with `collapse=" "` to separate the tweets in both data frames containing the tweets, `coffee_df` and `chardonnay_df`.
- Paste coffee tweets, look at structure of result, count characters:

```
all_coffee <- paste(coffee_df$text, collapse=" ")
str(all_coffee)
nchar(all_coffee)
```

```
 chr "@ayyytylerb that is so true drink lots of coffee RT @bryzy_brib: Senior Marc
[1] 88230
```

- Paste Chardonnay tweets, look at structure, count characters:

```
all_chardonnay <- paste(chardonnay_df$text, collapse=" ")
str(all_chardonnay)
nchar(all_chardonnay)
```

```
 chr "RT @oceanclub: @eilisohanlon @stonyjim @vonprond Eilis, I'm from Pearse St a
[1] 96880
```

- Combine all tweets from `all_coffee` and `all_chardonnay` in one vector `all_tweets`, show structure and number of characters:

```
all_tweets <- c(all_coffee, all_chardonnay)
str(all_tweets)
nchar(all_tweets)
```

```
 chr [1:2] "@ayyytylerb that is so true drink lots of coffee RT @bryzy_brib: Senio
[1] 88230 96880
```

- Create the corpus from vector and source and `inspect` it:

```
all_corpus <- VCorpus(VectorSource(all_tweets))
inspect(all_corpus)
```

```
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 88230

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 96880
```

## Visualize common words with `commonality.cloud`

- You need to clean the corpus, create a TDM that you can then visualize using `commonality.cloud` from the `wordcloud` package
- Clean the corpus by applying `clean_corpus` to `all_corpus`

```
clean_corpus <- function(corpus) {
  corpus <- tm_map(corpus,
                   removeNumbers)
```

```
  corpus <- tm_map(corpus,
                   removePunctuation)
  corpus <- tm_map(corpus,
                   content_transformer(tolower))
  corpus <- tm_map(corpus,
                   removeWords,
                   words = c(stopwords("en"),"coffee","beans",
                             "can", "hgtv","bean", "chardonnay",
                             "glass","glasses","wine","amp","just"))
  corpus <- tm_map(corpus,
                   stripWhitespace)
  return(corpus)
}
all_clean <- clean_corpus(all_corpus)
inspect(all_clean)

<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 55271

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 57999

<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 55960

[[2]]
```

```
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 58470
```

- Create a TDM `all_tdm` for the corpus `all_clean`:

```
all_tdm <- TermDocumentMatrix(all_clean)
all_tdm
```

```
<<TermDocumentMatrix (terms: 5406, documents: 2)>>
Non-/sparse entries: 6089/4723
Sparsity           : 44%
Maximal term length: 266
Weighting          : term frequency (tf)

<<TermDocumentMatrix (terms: 5409, documents: 2)>>
Non-/sparse entries: 6094/4724
Sparsity           : 44%
Maximal term length: 266
Weighting          : term frequency (tf)
```

- Convert `all_tdm` to a `matrix` object `all_m`

```
all_m <- as.matrix(all_tdm)
str(all_m)
```

```
num [1:5406, 1:2] 0 1 1 1 1 1 1 1 1 1 ...
- attr(*, "dimnames")=List of 2
 ..$ Terms: chr [1:5406] "aaliyahmaxwell" "abasc" "abbslovesfed" "abbycastro" ...
 ..$ Docs : chr [1:2] "1" "2"

num [1:5409, 1:2] 0 1 1 1 1 1 1 1 1 1 ...
- attr(*, "dimnames")=List of 2
 ..$ Terms: chr [1:5409] "aaliyahmaxwell" "abasc" "abbslovesfed" "abbycastro" ...
 ..$ Docs : chr [1:2] "1" "2"
```

- Create a commonality cloud from `all_m` with `max.words=100` and `colors="steelblue1"`:

```
clean_corpus <- function(corpus) {
  corpus <- tm_map(corpus,
                   removeNumbers)
  corpus <- tm_map(corpus,
                   removePunctuation)
  corpus <- tm_map(corpus,
                   content_transformer(tolower))
  corpus <- tm_map(corpus,
                   removeWords,
                   words = c(stopwords("en"),"coffee","beans",
                             "can", "hgtv","bean", "chardonnay",
                             "glass","glasses","wine","amp","just"))
  corpus <- tm_map(corpus,
                   stripWhitespace)
  return(corpus)
}
all_clean <- clean_corpus(all_corpus)
inspect(all_clean)
all_tdm <- TermDocumentMatrix(all_clean)
all_tdm
all_m <- as.matrix(all_tdm)
str(all_m)
commonality.cloud(term.matrix=all_m,
                  max.words=100,
                  colors="steelblue1")
```


## Visualize dissimilar words with `comparison.cloud`

- To visualize dissimilar words, you can use `comparison.cloud`, which has quite a few more arguments:

```
args(comparison.cloud)
```

```
function (term.matrix, scale = c(4, 0.5), max.words = 300, random.order = FALSE,
    rot.per = 0.1, colors = brewer.pal(max(3, ncol(term.matrix)),
        "Dark2"), use.r.layout = FALSE, title.size = 3, title.colors = NULL,
    match.colors = FALSE, title.bg.colors = "grey90", ...)
NULL
```

- Clean the corpus, create TDM:

```
clean_corpus <- function(corpus) {
```

```
  corpus <- tm_map(corpus,
                   removeNumbers)
  corpus <- tm_map(corpus,
                   removePunctuation)
  corpus <- tm_map(corpus,
                   content_transformer(tolower))
  corpus <- tm_map(corpus,
                   removeWords,
                   words = c(stopwords("en"),"coffee","beans",
                             "can", "hgtv","bean", "chardonnay",
                             "glass","glasses","wine","amp","just"))
  corpus <- tm_map(corpus,
                   stripWhitespace)
  return(corpus)
}
all_clean <- clean_corpus(all_corpus)
inspect(all_clean)
all_tdm <- TermDocumentMatrix(all_clean)
all_tdm

<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 2

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 55271

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 57999
<<TermDocumentMatrix (terms: 5406, documents: 2)>>
Non-/sparse entries: 6089/4723
Sparsity           : 44%
Maximal term length: 266
Weighting          : term frequency (tf)
```

- The `tdm` is organized neatly in two columns:

```
as.matrix(all_tdm)[200:205,]
```

```
           Docs
Terms       1 2
  asia      1 0
  asian     1 1
  ask       6 4
  asked     3 1
  asking    0 6
  askorange 2 0
```

- Use `colnames` to rename each distinct corpora within `all_tdm` so that we can keep track of the contributions from either corpus:

```
colnames(all_tdm) <- c("coffee","chardonnay")
as.matrix(all_tdm)[200:205,]
```

```
           Docs
Terms       coffee chardonnay
  asia           1          0
  asian          1          1
  ask            6          4
  asked          3          1
  asking         0          6
  askorange      2          0
```

```
           Docs
Terms       coffee chardonnay
  asia           1          0
  asian          1          1
  ask            6          4
  asked          3          1
  asking         0          6
  askorange      2          0
```

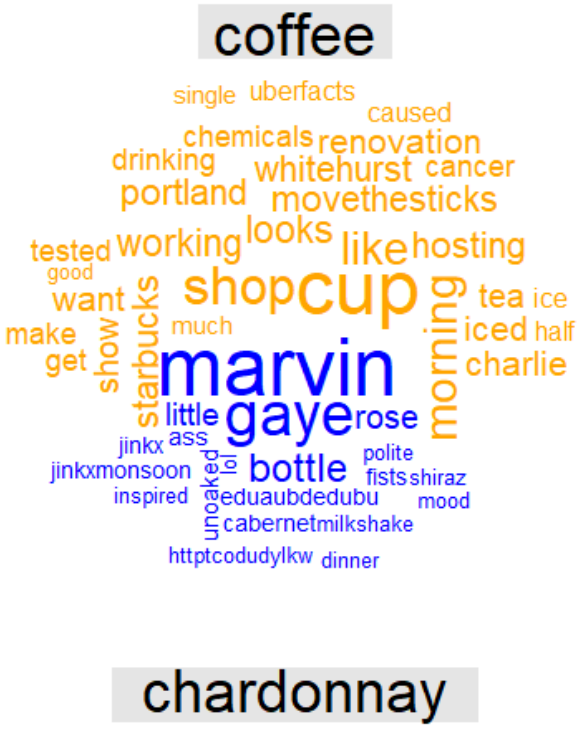
- Create a matrix `all_m` from `all_tdm`:

```
all_m <- as.matrix(all_tdm)
all_m[200:205,]
```

```
           Docs
Terms       coffee chardonnay
  asia           1          0
  asian          1          1
  ask            6          4
  asked          3          1
  asking         0          6
  askorange      2          0
```

- Create a comparison cloud with `max.words=50` and the `colors` "orange" and "blue":

```
clean_corpus <- function(corpus) {
  corpus <- tm_map(corpus,
                   removeNumbers)
  corpus <- tm_map(corpus,
                   removePunctuation)
  corpus <- tm_map(corpus,
                   content_transformer(tolower))
  corpus <- tm_map(corpus,
                   removeWords,
                   words = c(stopwords("en"),"coffee","beans",
                             "can", "hgtv","bean", "chardonnay",
                             "glass","glasses","wine","amp","just"))
  corpus <- tm_map(corpus,
                   stripWhitespace)
  return(corpus)
}
all_clean <- clean_corpus(all_corpus)
inspect(all_clean)
all_tdm <- TermDocumentMatrix(all_clean)
all_tdm
colnames(all_tdm) <- c("coffee","chardonnay")
as.matrix(all_tdm)[200:205,]
comparison.cloud(term.matrix=all_m,
                 max.words=50,
                 colors=c("orange","blue"))
```




## Compare word commonality with `pyramid_plot`

- We want to see which common words appear more often in which dataset: the `pyramid.plot` from the `plotrix` package delivers an aligned bargraph that shows this:

```
library(plotrix)
args(pyramid.plot)
```

```
function (lx, rx, labels = NA, top.labels = c("Male", "Age",
    "Female"), main = "", laxlab = NULL, raxlab = NULL, unit = "%",
    lxcol, rxcol, gap = 1, space = 0.2, ppmar = c(4, 2, 4, 2),
    labelcex = 1, add = FALSE, xlim, show.values = FALSE, ndig = 1,
    do.first = NULL)
NULL
```

- Data transformation: we need a data frame with three columns, the words contained in each document, and the counts from each:

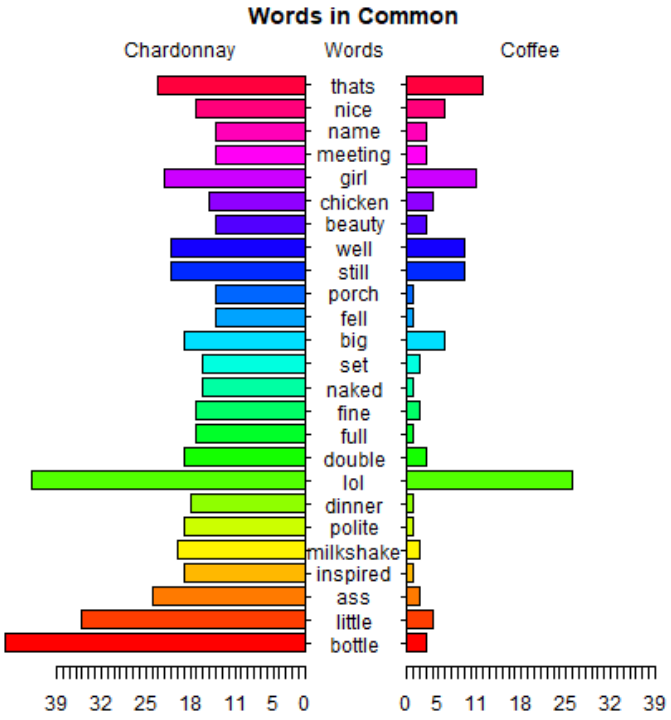
  1. Coerce `all_m` to a "tibble" (a special type of data frame)
  2. `filter` all words with non-zero frequency in either dataset
  3. add a `difference` column with the difference in counts by word
  4. extract those records with more than 25 counts difference
  5. arrange the records in descending order

```
library(dplyr)
top25_df <- all_m %>%
  ## Convert to data frame
  as_tibble(rownames = "word") %>%
  ## Keep rows where word appears everywhere
  filter(if_all(everything(), ~. > 0)) %>%
  ## Get difference in counts
  mutate(difference = chardonnay - coffee) %>%
  ## Keep rows with biggest difference
  slice_max(difference,  n = 25) %>%
  ## Arrange by descending difference
  arrange(desc(difference))
```

- To create the pyramid plot,

  1. set the left count to the `chardonnay` column
  2. set the right count to the `coffee` column
  3. set the labels to the `word` column

```
pyramid.plot(
  ## Chardonnay counts
  top25_df$chardonnay,
  ## Coffee counts
  top25_df$coffee,
  ## Words
  labels = top25_df$word,
  top.labels = c("Chardonnay", "Words", "Coffee"),
  main = "Words in Common",
  unit = NULL,
  gap = 8,
  )
```

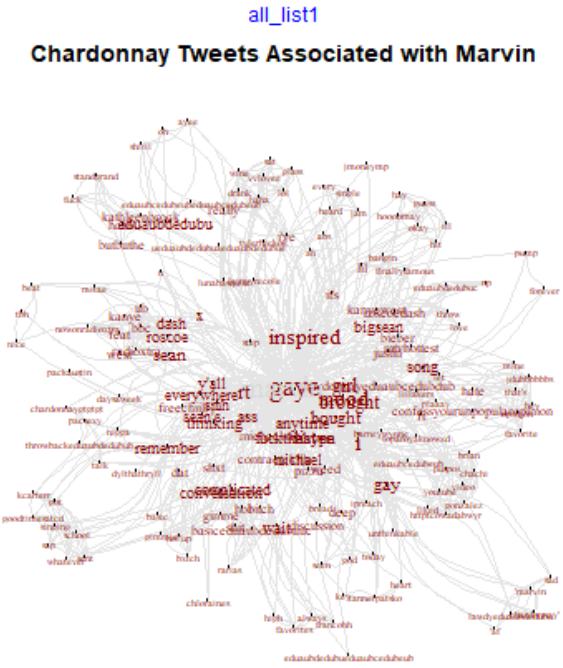



## Visualize word networks

- Word networks show term association (with a link) and cohesion (neighborhoods and density of links), like a social network.
- In a network graph, the circles are called *nodes* and represent individual terms, while the lines connecting the circles are called *edges* and represent the connections between the terms.
- The `qdap` package contains `word_network_plot` and `word_associate` to create word networks.
- This code constructs a word network for words associated with "Marvin", a dominant word in the Chardonnay tweets:

```
## Word association
word_associate(chardonnay_df$text,
```

```
                match.string = "marvin",
                stopwords = c(Top200Words, "chardonnay", "amp"),
                network.plot = TRUE,
                cloud.colors = c("gray85", "darkred"))
## Add title
title(main = "Chardonnay Tweets Associated with Marvin")
```




- To get the printed output information, run the code block again without graphics - the graph will open in a separate window:

```
## Word association
word_associate(chardonnay_df$text,
                match.string = "marvin",
                stopwords = c(Top200Words, "chardonnay", "amp"),
                network.plot = TRUE,
                cloud.colors = c("gray85", "darkred"))
```

```
## Add title
title(main = "Chardonnay Tweets Associated with Marvin")
```

```
    row group unit text
1    14   all   14 This was all inspired by a little Marvin Gaye and Chardonnay..
2    16   all   16 @LilLakers JUST TO SET THE MOOD GIRL I BROUGHT SOME MARVIN GAYE
3    17   all   17 RT @_barneywynne_: Just to set the mood girl i brought some mar
4    18   all   18 @TylerHickok was it inspired by a little Marvin gaye abs chardo
5    19   all   19 Just to set the mood girl i brought some marvin gaye and chardo
6    23   all   23 Marvin Gaye and Chardonnay
7    24   all   24 I brought some Marvin Gaye and Chardonnay.
8    26   all   26 RT @NowOnRadio1Xtra: <U+266B> Marvin &amp; Chardonnay (feat. Ka
9    27   all   27 <U+266B> Marvin &amp; Chardonnay (feat. Kanye West &amp; Roscoe
10   48   all   48 Marvin Gaye and Chardonnay
11   67   all   67 Just to set the mood, girl i brought some marvin gaye and Charo
12  101   all  101 Marvin gay and Chardonnay <ed><U+00A0><U+00BD><ed><U+00B8><U+00
13  126   all  126 This was all inspired by a little Marvin Gaye and Chardonnay <e
14  143   all  143 RT @Leinyy_Nicole: and this was all inspired by a little Marvin
15  144   all  144 and this was all inspired by a little Marvin Gaye and Chardonna
16  146   all  146 Marvin Gaye and Chardonnay
17  170   all  170 Guess ill Just Hit the Hay After a Lil Marvin Gaye &amp; Chardo
18  175   all  175 Marvin Gaye and Chardonnay
19  195   all  195 Marvin Gay &amp; Chardonnay
20  196   all  196 Marvin Gaye and Chardonnay
21  201   all  201 Marvin Gaye &amp; Chardonnay, was my shxt, I Use To Play It Bef
22  204   all  204 ?@rarias_453: Hol up give me that gimme that Marvin Gaye and Ch
23  205   all  205 Hol up give me that gimme that Marvin Gaye and Chardonnay shit
24  209   all  209 Big Sean x Roscoe Dash x Marvin Gaye N Chardonnay
25  211   all  211 RT @LunaBasquiat: This was all inspired by a little Marvin Gaye
26  212   all  212 This was all inspired by a little Marvin Gaye &amp; Chardonnay
27  225   all  225 RT @FreeChiill: Y'all remember when Big Sean's "Ass" &amp; "Mar
28  226   all  226 RT @FreeChiill: Y'all remember when Big Sean's "Ass" &amp; "Mar
29  227   all  227 Y'all remember when Big Sean's "Ass" &amp; "Marvin Gaye &amp; C
30  238   all  238 It's to set the mood girl I bought some Marvin and Chardonnay.
31  246   all  246 Just to set the mood he put some Marvin Gaye and Chardonnay <ed
32  249   all  249 RT @Contract_cKilla: ?@_FuckTheHype_: Anytime I listen to Marvi
33  252   all  252 RT @Contract_cKilla: ?@_FuckTheHype_: Anytime I listen to Marvi
34  253   all  253 ?@_FuckTheHype_: Anytime I listen to Marvin Gaye and Chardonnay
35  256   all  256 ?@_FuckTheHype_: Anytime I listen to Marvin Gaye and Chardonnay
36  258   all  258 Anytime I listen to Marvin Gaye and Chardonnay or She Will.. I
```

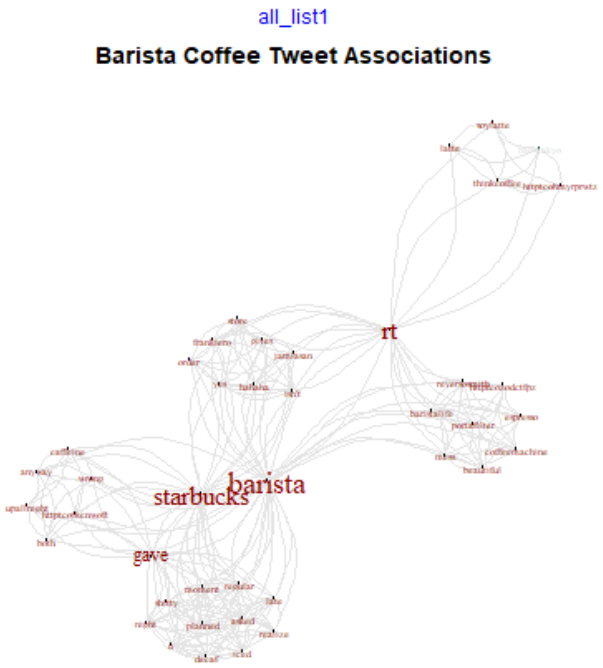
```
37  260   all  260 A little Marvin Gaye &amp;amp; Chardonnay..
38  273   all  273 What dat nigga Big Sean say.. we can do it off this Marvin Gaye
39  347   all  347 Still jam out to Marvin Gaye and Chardonnay like it's the first
40  363   all  363 RT @Dyl_Tha_Thryll: Marvin Gaye and Chardonnay
41  364   all  364 Marvin Gaye and Chardonnay
42  379   all  379 This was all inspired by ah lil Marvin Gaye and Chardonnay
43  385   all  385 Marvin Gaye and Chardonnay
44  386   all  386 ?@Stand__Grand: @kathleen_brock but....but...he hates Marvin an
45  388   all  388 @kathleen_brock but....but...he hates Marvin and Chardonnay <ed
46  390   all  390 ?@JDubbbbbbs: #confessyourunpopularopinion I HATE Marvin &amp;amp;
47  392   all  392 #confessyourunpopularopinion I HATE Marvin &amp;amp; Chardonnay
48  401   all  401 Marvin gay and Chardonnay
49  404   all  404 ?@Tanner_Patsko40: Little Marvin Gaye a Chardonnay? the key to
50  408   all  408 Little Marvin Gaye a Chardonnay
51  413   all  413 @VVLovee haha Marvin Gaye and Chardonnay? Lol I've really sat d
52  427   all  427 RT @MiTae_: Big sean x Marvin and chardonnay
53  430   all  430 Big sean x Marvin and chardonnay
54  433   all  433 @PackAustin Marvin Gaye and Chardonnay
55  435   all  435 Marvin &amp;amp; chardonnay - Big Sean
56  450   all  450 A little Marvin Gaye and Chardonnay
57  486   all  486 ?@pacsexy: Marvin Gaye &amp;amp; Chardonnay&amp;gt;&amp;gt;&amp;gt;? talk about
58  490   all  490 Marvin gaye and chardonnay
59  497   all  497 This was all inspired by a little Marvin Gaye and Chardonnay
60  500   all  500 this was all inspired by a little Marvin Gaye and Chardonnay
61  502   all  502 and this was all inspired by a little Marvin Gaye and Chardonna
62  507   all  507 Marvin gaye and chardonnay by @BigSean ft @kanyewest @roscoedas
63  521   all  521 "This was all inspired by a little Marvin Gaye and Chardonnay"
64  525   all  525 @Franc__OHH Wait for me, marvin and chardonnay, high and I do i
65  554   all  554 And this was all inspired by a little Marvin Gaye and Chardonna
66  587   all  587 RT @_iPreach: Just to set the mood I bought some Marvin Gay and
67  614   all  614 Just to set the mood I bought some Marvin Gay and Chardonnay
68  637   all  637 "Marvin and Chardonnay" by Big Sean has a nice beat tbh
69  638   all  638 @JMoney814MP marvin GAYe and chardonnay
70  658   all  658 i can not listen to unthinkable, work out, or marvin gaye &amp;amp
71  669   all  669 This was all inspired by a little Marvin Gaye and Chardonnay
72  679   all  679 Marvin Gaye and Chardonnay <ed><U+00A0><U+00BC><ed><U+00BE><U+0
73  694   all  694 Marvin gaye and chardonnay
74  696   all  696 Gimmie dat Chardonnay &amp;amp; that Marvin Gaye shxt .. but hol'up
75  718   all  718 "This was all inspired by a little Marvin Gaye an Chardonnay "
76  727   all  727 Just to set the mood girl I brought some Marvin Gay and Chardon
```

```
77  749   all  749 @Chlo_Raines Marvin Gaye and Chardonnay
78  752   all  752 Marvin and Chardonnay will forever be my pump up song
79  763   all  763 Do it how we want ! Just to set the mood girl I brought some Ma
80  764   all  764 #Np Marvin &amp; Chardonnay ! #BigSean ! #MTVHottest Justin Bie
81  789   all  789 Girl i brought some MARVIN GAYE AND CHARDONNAY
82  800   all  800 <U+2728><ed><U+00A0><U+00BD><ed><U+00B2><U+00A8><ed><U+00A0><U+
83  803   all  803 I seen God today.. we had a deep discussion over Marvin Gaye an
84  808   all  808 Marvin Gaye and Chardonnay
85  813   all  813 just to set the mood girl I bought some marvin gaye and chardon
86  829   all  829 Marvin gaye and chardonnay
87  849   all  849 RT @14DaysAWeek_: Marvin Gaye and Chardonnay
88  871   all  871 Marvin Gaye and Chardonnay
89  885   all  885 Marvin Gaye and Chardonnay
90  895   all  895 ?@LoParoYaKnowXD: Just to set the mood girl I brought some Marv
91  896   all  896 Just to set the mood girl I brought some Marvin Gaye and Chardo
92  899   all  899 This was all inspired by a little Marvin Gaye and Chardonnay.
93  904   all  904 Marvin &amp; Chardonnay(:
94  908   all  908 @K_Carterr35 REMEMBER WHEN YOU WAS SINGING MARVIN &amp; CHARDON
95  909   all  909 I was sad 'af but then 'Marvin Gay &amp; Chardonnay' just came
96  910   all  910 Marvin &amp; Chardonnay was my song!
97  911   all  911 - Just to set the mood girl I bought Marvin &amp; Chardonnay .
98  913   all  913 Oh shiiii Marvin &amp; Chardonnay really just came on?! Ayee <e
99  926   all  926 We had a deep discussion over Marvin Gaye and Chardonnay
100 935   all  935 I liked a @YouTube video http://t.co/waDAbwYR14 Marvin Gaye and
101 939   all  939 Marvin Gay &amp; Chardonnay
102 947   all  947 "And this was all inspired by a little Marvin Gaye and Chardonn
103 951   all  951 Just to set the mood I bought some Marvin Gaye &amp; Chardonnay

Match Terms
===========

List 1:
marvin, 'marvin


Warning message:
In text2color(words = V(g)$label, recode.words = target.words, colors = label.colo
  length of colors should be 1 more than length of recode.words
```

- This code constructs a word network for words associated with "barista",

a word in the coffee tweets:

```
## Word association
word_associate(coffee_df$text,
               match.string = "barista",
               stopwords = c(Top200Words, "coffee", "amp"),
               network.plot = TRUE,
               cloud.colors = c("gray85", "darkred"))
## Add title
title(main = "Barista Coffee Tweet Associations")
```




## Resources

`load_packages`

```
load_packages <- function() {
  library(tm)
```

```
  library(qdap)
  library(SnowballC)
  library(wordcloud)
  search()
}
load_packages()
```

```
 [1] ".GlobalEnv"               "package:plotrix"
 [3] "package:dplyr"            "package:viridisLite"
 [5] "package:wordcloud"        "package:SnowballC"
 [7] "package:qdap"             "package:RColorBrewer"
 [9] "package:qdapTools"        "package:qdapRegex"
[11] "package:qdapDictionaries" "package:tm"
[13] "package:NLP"              "ESSR"
[15] "package:stats"            "package:graphics"
[17] "package:grDevices"        "package:utils"
[19] "package:datasets"         "package:stringr"
[21] "package:httr"             "package:methods"
[23] "Autoloads"                "package:base"
```

clean_corpus

```
clean_corpus <- function(corpus) {
  corpus <- tm_map(corpus,
                   removeNumbers)
  corpus <- tm_map(corpus,
                   removePunctuation)
  corpus <- tm_map(corpus,
                   content_transformer(tolower))
  corpus <- tm_map(corpus,
                   removeWords,
                   words = c(stopwords("en"),"coffee","beans",
                             "can", "hgtv","bean", "chardonnay",
                             "glass","glasses","wine","amp","just"))
  corpus <- tm_map(corpus,
                   stripWhitespace)
  return(corpus)
}
```