UPPSALA
UNIVERSITET

Text analysis II: Commending the tm package

Introduction to R for Social Sciences

Josef Ginnerskov
Department of Sociology
2022-08-23

Today's outline

1. A conceptual background to (computational) text analysis in R
2. **Basic text analysis tasks performed on vector strings following the logic of the tm package**
3. More advance text analysis tasks conducted on digitized books based on the tidytext package
4. Individually solving the problem set by building your own Gutenberg corpus

Loading character vector documents (1/2)

To get going with our text analysis we need something to work with and to make it really simple we can generate our own documents by writing a few character vectors, such as a few juicy quotes from classical sociologists.

Loading character vector documents (2/2)

```
Soctxt.raw <- c(Durkheim = "Sociological method as we practice it rests wholly on the basic principle that social facts
   must be studied as things, that is, as realities external to the individual. There is no principle for which we
   have received more criticism; but none is more fundamental. Indubitably for sociology to be possible, it must above
   all have an object all its own. It must take cognizance of a reality which is not in the domain of other
   sciences... there can be no sociology unless societies exist, and that societies cannot exist if there are only
   individuals.",
               Weber = "'Sociology' is a word which is used in many different senses. In the sense adopted here, it
   means the science whose object is to interpret the meaning of social action and thereby give a causal explanation
   of the way in which the action proceeds and the effects which it produces. By 'action' in this definition is meant
   human behaviour when and to the extent that the agent of agents see it as subjectively meaningful: the behaviour
   may be either internal or external, and may consist in the agent's doing something, omitting to do something, or
   having something done to him. By 'social' action is meant an action in which the meaning intended by the agent or
   agents involves a relation to another person's behaviour and in which that relation determines the way in which the
   action proceeds.",
               Simmel = "I UNDERSTAND the task of sociology to be description and determination of the historico-
   psychological origin of those forms in which interactions take place between human beings. The totality of these
   interactions, springing from the most diverse impulses, directed toward the most diverse objects, and aiming at the
   most diverse ends, constitutes 'society'. Those different contents in connection with which the forms of
   interaction manifest themselves are the subject-matter of special sciences. These contents attain the character of
   social facts by virtue of occurring in this particular form in the interactions of men.",
             Tarde = "I will pass over a number of secondary objections which the application of the sociological
   point of view may encounter along its way. Since, after all, the fundamental nature of things is strictly
   inaccessible, and we are obliged to construct hypotheses in order to penetrate it, let us openly adopt this one and
   push it to its conclusion. Hypotheses fingo, I say naively. What is dangerous in the sciences are not tightly
   linked conjectures, logically followed to the ultimate depths or the ultimate precipices, but rather the ghosts of
   ideas which float aimlessly in the mind. The universal sociological point of view seems to me to be one of these
   spectres which haunt the brains of our speculative contemporaries.")
```

Creating a corpus (vector source)

Today we going to utilize the tm package, which is the text mining package used in part 5.1 Discovery with Textual Data in the course literature *Quantitative Social Science* by Kosuke Imai. To facilitate the analysis we can start by building a corpus object so that we can easily run each task on the whole body of texts.

```
library(tm) # package for general text mining tasks
Soctxt.corp <- VCorpus(VectorSource(Soctxt.raw)) # create a volatile corpus, kept in memory as a R object.
inspect(Soctxt.corp) # let us have an overarching look with the tm function inspect
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 4
##
## $Durkheim
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 549
##
## $Weber
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 793
##
## $Simmel
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 624
##
```

Exporting corpus and importing txt

Perhaps we would at some point like to export our corpus from Rstudio to a folder on our disk. This can be made conveniently with the tm package.

```
writeCorpus(Soctxt.corp) # write corpus disk
Soctxt.dir <- VCorpus(DirSource(pattern = ".txt")) # generate a new corpus directly from disk; default is reading txt
   from the working directory
inspect(Soctxt.dir)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 4
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 549
##
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 793
##
## [[3]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 624
##
## [[4]]
```

Managing corpus metadata

The two main reasons for why you would like to have a corpus object is to store the documents' textual content and metadata. The later is use to keeping track of and comparing the documents. With the following code, we can couple the quote with its author.

```
meta(Soctxt.corp[[1]], "author") <- "Durkheim" # select the first quote and add the name to the author variable.
meta(Soctxt.corp[[2]], "author") <- "Weber"
meta(Soctxt.corp[[3]], "author") <- "Simmel"
meta(Soctxt.corp[[4]], "author") <- "Tarde"

meta(Soctxt.corp[[3]]) # take a look at the metadata of the third quoute by Simmel
```

```
##    author       : Simmel
##    datetimestamp: 2022-08-23 13:11:09
##    description  : character(0)
##    heading      : character(0)
##    id           : 3
##    language     : en
##    origin       : character(0)
```

Preprocessing - to lower case

We will here work with the bag-of-words model and, thus, we want words with the same meaning to end up in the same count. To enable this, we need to do some preprocessing tasks with the tm_map function. First off, we would like to treat words with uppercase and/or lowercase letters as the same.

```
Soctxt.corp.low <- tm_map(Soctxt.corp, content_transformer(tolower)) # we transform all letters to lowercase

Soctxt.corp.low[[3]]$content # what happened to Simmel?
```

```
## [1] "i understand the task of sociology to be description and determination of the historico-psychological origin of
   those forms in which interactions take place between human beings. the totality of these interactions, springing from
   the most diverse impulses, directed toward the most diverse objects, and aiming at the most diverse ends, constitutes
   'society'. those different contents in connection with which the forms of interaction manifest themselves are the
   subject-matter of special sciences. these contents attain the character of social facts by virtue of occurring in
   this particular form in the interactions of men."
```

Preprocessing - remove punctuation

Second, punctuation within and around words can also disturb our preferred word couplings (e.g. "full-text" and "fulltext").

```
Soctxt.corp.punct <- tm_map(Soctxt.corp, removePunctuation) # we simply remove all forms of puntuations

Soctxt.corp.punct[[3]]$content # what happened to Simmel?
```

```
## [1] "I UNDERSTAND the task of sociology to be description and determination of the historicopsychological origin of
    those forms in which interactions take place between human beings The totality of these interactions springing from
    the most diverse impulses directed toward the most diverse objects and aiming at the most diverse ends constitutes
    society Those different contents in connection with which the forms of interaction manifest themselves are the
    subjectmatter of special sciences These contents attain the character of social facts by virtue of occurring in this
    particular form in the interactions of men"
```

Preprocessing - managing stop words (1/2)

Thirdly, too common and indistinct words, which are refereed to as stop words, will create noise for most text analysis tasks. Thus, these will have to be removed. We can both make up our own set of words and rely on universal dictionaries of words, in this case we will use English stop words.

```
stopwords("en") # tm's list is taken from the Snowball stemmer project

Soctxt.stop.words <- c("something", "can", "must", "since") # adding some words not covered in the list
```

```
##   [1] "i"         "me"        "my"        "myself"    "we"
##   [6] "our"       "ours"      "ourselves" "you"       "your"
##  [11] "yours"     "yourself"  "yourselves" "he"       "him"
##  [16] "his"       "himself"   "she"       "her"       "hers"
##  [21] "herself"   "it"        "its"       "itself"    "they"
##  [26] "them"      "their"     "theirs"    "themselves" "what"
##  [31] "which"     "who"       "whom"      "this"      "that"
##  [36] "these"     "those"     "am"        "is"        "are"
##  [41] "was"       "were"      "be"        "been"      "being"
##  [46] "have"      "has"       "had"       "having"    "do"
##  [51] "does"      "did"       "doing"     "would"     "should"
##  [56] "could"     "ought"     "i'm"       "you're"    "he's"
##  [61] "she's"     "it's"      "we're"     "they're"   "i've"
##  [66] "you've"    "we've"     "they've"   "i'd"       "you'd"
##  [71] "he'd"      "she'd"     "we'd"      "they'd"    "i'll"
##  [76] "you'll"    "he'll"     "she'll"    "we'll"     "they'll"
##  [81] "isn't"     "aren't"    "wasn't"    "weren't"   "hasn't"
##  [86] "haven't"   "hadn't"    "doesn't"   "don't"     "didn't"
##  [91] "won't"     "wouldn't"  "shan't"    "shouldn't" "can't"
```

Preprocessing - managing stop words (2/2)

Now we are ready to remove the two stop word lists with tm_map.

```
Soctxt.corp.stop <- tm_map(Soctxt.corp, removeWords, stopwords("en")) # you can remove any set of words with
   removeWords
Soctxt.corp.stop <- tm_map(Soctxt.corp.stop, removeWords, Soctxt.stop.words) # we also remove our own set of words

Soctxt.corp.stop[[3]]$content # what happened to Simmel?
```

```
## [1] "I UNDERSTAND  task  sociology  description  determination  historico-psychological origin  forms
   interactions take place  human beings. The totality  interactions, springing  diverse impulses, directed toward
   diverse objects,  aiming  diverse ends, constitutes 'society'. Those different contents  connection  forms
   interaction manifest  subject-matter  special sciences. These contents attain  character  social facts  virtue
   occurring  particular form  interactions  men."
```

Preprocessing - strip whitespace

Thirdly, we want to remove blank spaces between words, which in programming is called whitespace and refers to any character or series of characters that represent horizontal or vertical space.

```
stripcorpus <- tm_map(Soctxt.corp, stripWhitespace) # this function will erase all forms of whitespace

stripcorpus[[3]]$content # what happened to Simmel?
```

```
## [1] "I UNDERSTAND the task of sociology to be description and determination of the historico-psychological origin of
   those forms in which interactions take place between human beings. The totality of these interactions, springing from
   the most diverse impulses, directed toward the most diverse objects, and aiming at the most diverse ends, constitutes
   'society'. Those different contents in connection with which the forms of interaction manifest themselves are the
   subject-matter of special sciences. These contents attain the character of social facts by virtue of occurring in
   this particular form in the interactions of men."
```

Preprocessing - stemming

A more debated form of preprocessing is whether words should be taken for what they are or if one ought to merge words with the same stem (e.g. let "power" represent "power", "powers", "powerful" and "powerless"). This can enhance your interpretation but also cause problems. Let us try stemming our corpus.

```
stemcorpus <- tm_map(Soctxt.corp, stemDocument) # tm uses Porter's stemming algorithm

stemcorpus[[3]]$content # what happened to Simmel?
```

```
## [1] "I UNDERSTAND the task of sociolog to be descript and determin of the historico-psycholog origin of those form in
   which interact take place between human beings. The total of these interactions, spring from the most divers
   impulses, direct toward the most divers objects, and aim at the most divers ends, constitut society'. Those differ
   content in connect with which the form of interact manifest themselv are the subject-matt of special sciences. These
   content attain the charact of social fact by virtu of occur in this particular form in the interact of men."
```

Applying all preprocessing tasks

In most cases, you have an idea of what preprocessing tasks you would like to use from the get-go. Thus, it is often a more convenient strategy to run multiple tm_map commands at once.

```
Soctxt.corp.clean <- Soctxt.corp %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removePunctuation, preserve_intra_word_dashes = TRUE)  %>%
  tm_map(removeWords, stopwords("en")) %>%
  tm_map(stemDocument) %>%
  tm_map(stripWhitespace)

Soctxt.corp.clean[[3]]$content # what happened to Simmel?
```

```
## [1] "understand task sociolog descript determin historico-psycholog origin form interact take place human be total
   interact spring divers impuls direct toward divers object aim divers end constitut societi differ content connect
   form interact manifest subject-matt special scienc content attain charact social fact virtu occur particular form
   interact men"
```

Generating a document-term matrix

With our clean corpus at hand we are ready to compute a document-term matrix to store word scores document by document. It is possible to do all preprocessing tasks at the same time as you create your dtm.

```
Soctxt.dtm <- DocumentTermMatrix(Soctxt.corp.clean) # generate dtm from the preprocessed corpus

Soctxt.dtm.clean <- DocumentTermMatrix(Soctxt.corp.clean, # generate and preprocessing a dtm from the original corpus
                                control = list(removePunctuation = TRUE, stripWhitespace = TRUE,
                                               removeSparseTerms = 0.99, # remove terms that are used too few times
                                               stopwords = TRUE,
                                               stemming = TRUE))
inspect(Soctxt.dtm) #Inspect dtm
```

```
## <<DocumentTermMatrix (documents: 4, terms: 146)>>
## Non-/sparse entries: 169/415
## Sparsity           : 71%
## Maximal term length: 19
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs action agent behaviour divers form interact object scienc social sociolog
##    1      0     0         0      0    0        0      1      1      1        3
##    2      6     5         3      0    0        0      1      1      2        1
##    3      0     0         0      3    3        4      1      1      1        1
##    4      0     0         0      0    0        0      1      1      0        2
```

Operating a dtm - terms per corpus

The dtm can be used for a lot of more advance machine learning methods like topic modeling, but let us begin with the more simple tasks that the tm has to offer, like counting words for the whole corpus.

```
findFreqTerms(Soctxt.dtm, 4) # find terms appearing at least 4 times
```

```
## [1] "action"   "agent"    "interact" "object"   "scienc"   "social"   "sociolog"
```

Operating a dtm - terms per doc

You can also count words for each document.

```
findMostFreqTerms(Soctxt.dtm) # find most frequent terms for each document
```

```
## $`1`
##     must sociolog    exist individu principl  realiti
##        3        3        2        2        2        2
##
## $`2`
##    action     agent behaviour      mean    someth       may
##         6         5         3         3         3         2
##
## $`3`
## interact   divers     form  content      aim   attain
##        4        3        3        2        1        1
##
## $`4`
## hypothes      one    point sociolog    ultim     view
##        2        2        2        2        2        2
```

Operating a dtm - term correlations

Perhaps your are interested in to what extent a set of words are associated, i.e. occurs together or not in the same document. For this task we can calculate correlations.

```
findAssocs(Soctxt.dtm, terms = "sociolog", corlimit = 0.6) # terms correlating to one or several specified terms
```

```
## $sociolog
## fundament     thing     basic       can    cogniz    critic    domain     exist
##      0.90      0.90      0.87      0.87      0.87      0.87      0.87      0.87
##  individu   indubit    method      must      none   possibl   practic  principl
##      0.87      0.87      0.87      0.87      0.87      0.87      0.87      0.87
##   realiti    receiv      rest     studi    unless    wholli   societi
##      0.87      0.87      0.87      0.87      0.87      0.87      0.64
```

BONUS: From dtm to word cloud (1/2)

To end on a fun note, we can move beyond the tm package and generate a word cloud for our corpus.

```
library(wordcloud) # packge for generating wordclouds
Soctxt.df <- as.matrix(Soctxt.dtm) # first we need to covert our dtm to a matrix object
Soctxt.df <- sort(colSums(Soctxt.df), decreasing = TRUE) # sort the columns in decreasing order
Soctxt.df <- data.frame(word = names(Soctxt.df), freq = Soctxt.df) # create a data frame with the names of the words
   and their frequencies

wordcloud(Soctxt.df$word, Soctxt.df$freq, colors = brewer.pal(12, "Dark2")) # generate a word cloud in a dark color
   scheme
```

BONUS: From dtm to word cloud (2/2)

Thank you for your time!

Do not hesitate to contact me


josef.ginnerskov@soc.uu.se


@doeparen


@doeparen