Text Mining the Anonymous Marking Audit Trail Excel Export

Fiona MacNeill

06/06/2019

Learning Technologies Scenario:

You have access to data from the Anonymous Marking Audit Excel file which can you download from the Turnitin administrative account. You would like to find out if there are any patterns in this text-based data and whether there are any misconceptions about anonymous marking.

This data was simulated by Fiona MacNeill on 6 May 2019. This tutorial takes you from the original Microsoft Excel file, all the way to a finished word cloud. Tweaking will be needed to get the text-based data how you want it, but the good news is that the parameters that you need are in this tutorial. Certain sections are commented out as you may or may not need them (just remove the "#" in order to try them out).

Turnitin allows you to export the Anonymous Marking Audit Trail data if you have access to the administrative account and for information about how to do that, please take a look at the tutorial from help.turnitin.com: https://help.turnitin.com/feedback-studio/moodle/direct-v2/administrator/anonymous-marking/viewing-an-anonymous-marking-audit-trail.htm?Highlight=turn%20off%20anonymous%20marking || CC BY-NC 4.0.

I made use of the following tutorial from STHDA for learning text mining skills and I owe the authors a debt of immense gratitude.

If you are new to using RStudio you will need to get setup first by…

1. Installing R from the R Archive:

   i. Pick a mirror close-by (geographically speaking), e.g in UK – University of Bristol, Imperial College London
   ii. Mac tip: make sure that you check the MD5 hash and SHA hash match. You can do this quickly and easily in terminal. As shown in this video: https://youtu.be/HHdrIlHS2-4
   iii. [Mac only] XQuartz Install – information about this is provided at R Archive above.

2. Install RStudio: again do check the MD5. You can get the installer here – https://www.rstudio.com/products/rstudio/download/#download

1. Install the packages that you need

There are a number of packages that you will need. The first four packages below are only for importing, extracting and transforming the data from the original Excel spreadsheet. You can do this more simply by copying the "Reasons" column into a text editor like TextWrangler and then saving it as a .txt file. Using this option would allow you to skip step 3, but there is something to be said for understanding how to transform your data.

2. Load the libraries for the packages that you have installed

Now that you have installed the requisite packages you need to load the libraries so that they are available in your R environment. If you copy the complete directory, *'wordcloud_tutorial'* to your computer and open the RMarkdown file *'wordcloud_tutorial.Rmd'* from within it, you should not need to change the file path information in Steps 1-4.

```r
library("tm")
```

```
## Loading required package: NLP
```

```r
library("SnowballC")
library("wordcloud")
```

```
## Loading required package: RColorBrewer
```

```r
library("RColorBrewer")
library("ggplot2")
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
##
## Attaching package: 'ggplot2'
```

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

```r
library("readxl")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library("rio")

# Copy the directory to your computer
reasons.all <- read_excel("wordcloud_tutorial_files/reasons_tutorial.xlsx")

# Find out about the columns in the spreadsheet
names(reasons.all)
```

```
## [1] "Instructor Last Name"  "Instructor First Name" "Student Last Name"
## [4] "Student First Name"    "Paper ID"              "Date Turned Off"
## [7] "Reason"
```

3. Pull the "Reasons" column data and export it as text (.txt)

We want to pull the "Reason" column and convert it to a new data frame with a single variable and then export it as a text file so that we can create our text corpus (What is a text corpus? - Wikipedia).

Tip: You can take a look at the text file itself prior to completing step 4 and remove anything odd (such as strange characters that are hard to remove automatically) with a "find and replace" option in your text editor. It is worth going back to this if you see odd patterns in your data in the summary provided by 'inspect' at the end of step 5. Having made changes and saved your text file you can then re-run chunks 4-8 with your cleansed text file.

Side note: if you wanted to extract only unique reasons so that you have a concise list as a separate data.frame then you can quickly use the distinct option from the dplyr package - remove "#" to use it. This will keep only 'distinct' data.

```
reason <- pull(reasons.all, var="Reason") %>%
  data.frame %>% export(file = "wordcloud_tutorial_files/reasons_demo.txt")

#reason.individual <- distinct(reasons.all, Reason)
#reason.individual
```

4. Create the text corpus

This is using the base tools.

```
filePath <- "wordcloud_tutorial_files/reasons_demo.txt"

# Read the lines in the text document.
reason <- readLines(filePath)

# Create a text corpus based on our document
#- we are telling our existing variable 'reason' to become this new corpus.
reason <- Corpus(VectorSource(reason))
```

5. Clean up the text in your text corpus

You will need to do quite a bit of clean up on the text, particularly if you are working on several years worth of data. Thankfully the tm library is here to help and you can use the various transformations that it offers to complete clean-up tasks in-bulk!

```
# Change to lower case
reason <- tm_map(reason, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(reason, content_transformer(tolower)):
## transformation drops documents
```

```
# Remove numbers
reason <- tm_map(reason, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(reason, removeNumbers): transformation drops
## documents
```

```
# Remove english common stopwords
#reason <- tm_map(reason, removeWords, stopwords("en"))

# Remove punctuation
reason <- tm_map(reason, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(reason, removePunctuation): transformation
## drops documents
```

```r
# Eliminate extra white spaces
reason <- tm_map(reason, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(reason, stripWhitespace): transformation
## drops documents
```

```r
# Using inspect will allow you to see how the text-based data has changed
# based on your transformations
inspect(reason)
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 108
##
##   [1]
##   [2] marking is not anonymous on this module
##   [3] no student id to id lsp
##   [4] to identify who the marker should be
##   [5] no student id
##   [6] checking id
##   [7] no student id
##   [8] no student id
##   [9] checking id
##  [10] tailor feedback
##  [11] suspected case of plagiarism
##  [12] set up in error
##  [13] not required
##  [14]
##  [15]
##  [16]
##  [17] set up in error
##  [18] set up in error
##  [19] for feedback
##  [20] not required
##  [21] marking
##  [22] set up in error
##  [23] not required
##  [24] not required
##  [25] set up in error
##  [26] set up in error
##  [27] set up in error
##  [28] not required
##  [29] not required
##  [30] set up in error
##  [31] set up in error
##  [32] set up in error
##  [33] set by mistake
##  [34] set by mistake
##  [35] set by mistake
##  [36] set by mistake
##  [37] set by mistake
##  [38] set up in error
##  [39] set by mistake
##  [40] set by mistake
##  [41] set by mistake
##  [42] set by mistake
##  [43] incorrect setting
##  [44] incorrect setting
##  [45] incorrect setting
##  [46] set by mistake
##  [47] set by mistake
##  [48] set by mistake
##  [49] set up in error
##  [50] set by mistake
##  [51] set up in error
##  [52] set up in error
##  [53] set by mistake
##  [54] set up in error
##  [55] set up in error
##  [56] set by mistake
##  [57] mistake
##  [58]
##  [59]
##  [60]
##  [61] mistake
##  [62] mistake
##  [63] mistake
##  [64] mistake
##  [65] need to assess
##  [66] need to review
##  [67] need to assess
##  [68] students name is on the submission
##  [69] students name is on the submission
##  [70] students name is on the submission
##  [71] students name is on the submission
##  [72] students name is on the submission
##  [73] students name is on the submission
##  [74] students name is on the submission
##  [75] students name is on the submission
##  [76] to moderate
##  [77] to moderate
##  [78] to moderate
##  [79] to moderate
##  [80] to moderate
##  [81] to moderate
##  [82] to moderate
##  [83] to moderate
##  [84] to moderate
##  [85] to moderate
##  [86] to moderate
##  [87] to moderate
##  [88] duplication
##  [89] poss collusion
##  [90] plagiarism procedure
##  [91] academic misconduct report
##  [92] need to fill out an academic misconduct form with the students name
##  [93] no student number
##  [94] possible plagiarism investigation
##  [95] possible investigation for plagiarism
##  [96] provide personalised feedback
##  [97] provide personalised feedback
##  [98] provide personalised feedback
##  [99] provide personalised feedback
## [100] provide personalised feedback
## [101]
## [102]
## [103]
## [104] provide personalised feedback
## [105] to provide personalised feedback
## [106] provide personalised feedback
## [107] provide personalised feedback
## [108] provide personalised feedback
```

```r
# Remove some words that don't make sense
reason <- tm_map(reason, removeWords,
                 c("marking",
                   "provide",
                   "anonymous",
                   "set",
                   "need",
                   "poss",
                   "possible",
                   "submission",
                   "students",
                   "student's",
                   "the",
                   "for",
                   "this",
                   "not",
                   "module",
                   "name",
                   "with",
                   "&quot;"))
```

```
## Warning in tm_map.SimpleCorpus(reason, removeWords, c("marking",
## "provide", : transformation drops documents
```

6. When you are ready create the TermDocumentMatrix

This is about creating table or matrix containing the count information for each word.

a). Create the TermDocumentMatrix which is essentially applies a list of controls for manipulating your text corpus. Delete "#" before *inspect(dtm)* to see what happens.

b). "m <- as.matrix(dtm)" converts your TermDocumentMatrix into an actual matrix. As in a table type thing with counts per word. Delete "#" before *View(m)* to see what happens.

c). In this step we are now sorting our matrix from highest to lowest. Delete "#" before *View(v)* to see what happens.

d). Now we want to create a fresh data.frame with the data that we have mangled so that we can visualise it. Delete "#" before *View(d)* to see what happens.

```r
# create the term matrix based on your corpus
# Step a).
```

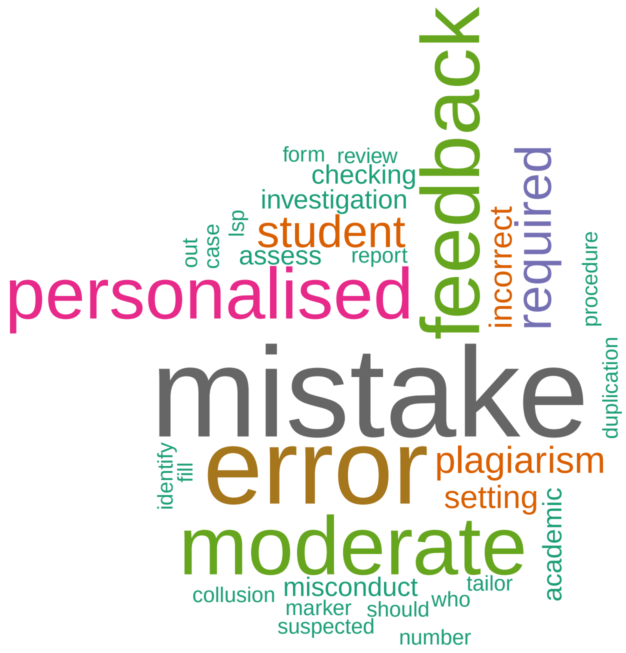
```r
dtm <- TermDocumentMatrix(reason)
#inspect(dtm)

# Step b).
m <- as.matrix(dtm)
#View(m)

# Step c).
v <- sort(rowSums(m), decreasing = TRUE)
#View(v)

# Step d).
d <- data.frame(word = names(v), freq=v)
#View(d)

# This is just for information and returns the top 30 terms
# in your new data.frame for your reference
head(d, 30)
```

```
##                        word freq
## mistake             mistake   20
## error                 error   16
## feedback           feedback   12
## moderate           moderate   12
## personalised   personalised   10
## required           required    6
## student             student    5
## plagiarism       plagiarism    4
## incorrect         incorrect    3
## setting             setting    3
## checking           checking    2
## assess               assess    2
## academic           academic    2
## misconduct       misconduct    2
## investigation investigation    2
## lsp                     lsp    1
## identify           identify    1
## marker               marker    1
## should               should    1
## who                     who    1
## tailor               tailor    1
## case                   case    1
## suspected         suspected    1
## review               review    1
## duplication     duplication    1
## collusion         collusion    1
## procedure         procedure    1
## report               report    1
## fill                   fill    1
## form                   form    1
```

7. Check if your word cloud is working

I would draw your attention to the min.freq and max.words below. So the min.freq being 1 in this case means that we are including words that are mentioned once. In a larger dataset you are going to want to set this threshold much higher; I recommend a minimum of 10. The max.words option restricts the number of words included in your cloud. You definitely need to use this if you are working with several years-worth of data. Also, do not worry if the preview below shows an error that saying that content cannot be fit on the page, the final export file in step 8 will display all the word cloud content.

```
set.seed(11249)
wordcloud(words = d$word,
          freq = d$freq,
          min.freq = 1,
          max.words = 200,
          random.order = FALSE,
          rot.per = 0.35,
          color=brewer.pal(8, "Dark2"))
```



```
# See vignette for RColorBrewer for different colour
# options by running the help command below. Remove the "#" before help.

# help("RColorBrewer")
```

8. Export your visualisation as a high-quality PNG

This is a good option to get a high-quality image for adding to reports and presentations.

```
# Set your image settings
png("wordcloud_demo_export.png", units="in", width=6, height=6, res=300)

# Create the plot
set.seed(11249)
wordcloud(words = d$word,
          freq = d$freq,
          min.freq = 1,
          max.words = 200,
          random.order = FALSE,
          rot.per = 0.35,
          color=brewer.pal(8, "Dark2"))

# Action the image production - the image should go into your directory
dev.off()
```

```
## pdf
##   2
```