LAB 10 - Further topics

Iulia Cioroianu

24/07/2020

1. Reading in data

```r
require(quanteda)
```

```
## Loading required package: quanteda

## Package version: 2.1.0

## Parallel computing: 2 of 16 threads used.

## See https://quanteda.io for tutorials and examples.

##
## Attaching package: 'quanteda'

## The following object is masked from 'package:utils':
##
##     View
```

```r
require(quanteda.textmodels)
```

```
## Loading required package: quanteda.textmodels

##
## Attaching package: 'quanteda.textmodels'

## The following object is masked from 'package:quanteda':
##
##     data_dfm_lbgexample
```

```r
require(topicmodels)
```

```
## Loading required package: topicmodels
```

```r
require(stm)
```

```
## Loading required package: stm

## stm v1.3.5 successfully loaded. See ?stm for help.
##  Papers, resources, and other materials at structuraltopicmodel.com
```

```r
require(lubridate)
```

```
## Loading required package: lubridate

##
## Attaching package: 'lubridate'
```

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

```r
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(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
articles <- read.csv("diffbot_text_basic.csv", stringsAsFactors = FALSE, encoding = "utf-8")

colnames(articles) <- c("X", "Date", "Sentiment","Region", "Country", "Site", "Type", "Title", "text")

# In the first part of the analysis we will focus on the UK.
art_uk <- subset(articles, Country == "United Kingdom")

# Change date to simpler format
art_uk$Date2 <- dmy(substr(art_uk$Date, 6, 16))

# Define the year and discard all articles before 2020
art_uk$year <- year(art_uk$Date2)
art_uk <- subset(art_uk, year >= 2020)

# Redefine year, month, week
art_uk$year <- year(art_uk$Date2)
art_uk$month <- month(art_uk$Date2)
art_uk$week <- week(art_uk$Date2)

# Create our corpus
corp_uk <- corpus(art_uk)
```

2. Processing: Include compounding multi-word expressions based on collocation analysis

```r
# Tokenize corpus
toks <- tokens(corp_uk,
               remove_numbers=TRUE,
               remove_punct=TRUE,
               remove_symbols=TRUE,
               remove_separators=TRUE,
               remove_url = TRUE,
               verbose=TRUE)
```

```
## Creating a tokens object from a corpus input...
##  ...starting tokenization
##  ...text1 to text666
##  ...preserving hyphens
##  ...preserving social media tags (#, @)
##  ...segmenting into words
##  ...22,709 unique types
##  ...removing separators, punctuation, symbols, numbers, URLs
##  ...complete, elapsed time: 1.36 seconds.
## Finished constructing tokens from 666 documents.
```

```r
# Remove stopwords
toks <- tokens_remove(toks, pattern = stopwords('en'))

# Identify collocations

tstat_col_cap <- textstat_collocations(toks, size=2, min_count = 10, tolower = FALSE)

head(tstat_col_cap, 20)
```

```
##           collocation count count_nested length   lambda         z
## 1      contact tracing  1458            0      2 6.421352 115.61147
## 2         Apple Google   575            0      2 6.600089  89.11548
## 3        public health   389            0      2 4.989355  73.36020
## 4      tested positive   208            0      2 6.562935  62.88122
## 5         human rights   185            0      2 7.665732  59.31179
## 6           Mr Hancock   179            0      2 6.210423  58.66975
## 7         Google Apple   234            0      2 4.737815  58.48396
## 8          track trace   161            0      2 6.130155  55.97554
## 9  contact-tracing app   396            0      2 4.139739  54.90559
## 10        around world   145            0      2 6.111660  54.19148
## 11   social distancing   277            0      2 8.855249  54.06475
## 12       location data   279            0      2 4.523757  54.05236
## 13        come contact   246            0      2 4.551813  53.38716
## 14         tracing app   490            0      2 2.827168  53.08473
## 15      Secretary Matt   118            0      2 7.933278  50.87028
## 16    Health Secretary   175            0      2 7.441061  49.64031
## 17           make sure   136            0      2 6.961924  49.15818
## 18        mobile phone   143            0      2 5.184597  49.11494
## 19        tracing apps   224            0      2 3.781514  48.97105
## 20        spread virus   156            0      2 4.667330  48.67244
```

Compound multi-word expressions

Results of collocation analysis can be use it to compound tokens. We will compound strongly associated multi-word expressions by sub-setting tstat_col_cap$collocation.

Collocations are automatically recognized as multi-word expressions by tokens_compound() in case-sensitive fixed pattern matching. This is the fastest way to compound large numbers of multi-word expressions, but make sure that tolower = FALSE in textstat_collocations() to do this.

```r
toks_comp <- tokens_compound(toks, pattern = tstat_col_cap[tstat_col_cap$z > 3])
# Text1 without compounding
toks[['text1']][1:50]
```

```
##  [1] "NHS"             "begun"           "feeding"         "health"
##  [5] "workers"         "use"             "personal"        "protective"
##  [9] "equipment"       "PPE"             "data"            "store"
## [13] "system"          "designed"        "identify"        "hospitals"
## [17] "GP"              "surgeries"       "risk"            "running"
## [21] "kit"             "address"         "problem"         "occurs"
## [25] "High-level"      "decision-makers" "able"            "start"
## [29] "seeing"          "information"     "via"             "computer"
## [33] "dashboard"       "within"          "fortnight"       "NHS"
## [37] "staff"           "say"             "lives"           "put"
## [41] "risk"            "PPE"             "shortages"       "government"
## [45] "said"            "working"         "around"          "clock"
## [49] "address"         "issue"
```

```r
# Text 1 with compounding
toks_comp[['text1']][1:50]
```

```
##  [1] "NHS"                               "begun"
##  [3] "feeding"                           "health_workers"
##  [5] "use_personal_protective_equipment" "PPE"
##  [7] "data_store"                        "system"
##  [9] "designed"                          "identify"
## [11] "hospitals"                         "GP"
## [13] "surgeries"                         "risk"
## [15] "running"                           "kit"
## [17] "address"                           "problem"
## [19] "occurs"                            "High-level"
## [21] "decision-makers"                   "able"
## [23] "start"                             "seeing"
## [25] "information"                       "via"
## [27] "computer"                          "dashboard"
## [29] "within"                            "fortnight"
## [31] "NHS_staff"                         "say"
## [33] "lives"                             "put_risk"
## [35] "PPE"                               "shortages"
## [37] "government_said_working"           "around"
## [39] "clock"                             "address"
## [41] "issue"                             "NHS"
## [43] "Providers"                         "represents"
## [45] "hospitals"                         "NHS"
## [47] "trusts"                            "England"
## [49] "told_BBC"                          "supplies"
```

3. Describing the data: Finding words associated with a certain word.

We can find words associated with target words using the window argument of tokens_select().

```r
# Create subset of tokens around the term "privacy"
toks_privacy <- tokens_keep(toks_comp, pattern = 'privacy', window = 10) # equivalent to tokens_select(
# Create subset of tokens which are not around the term "privacy"
toks_noprivacy <- tokens_remove(toks_comp, pattern = 'privacy', window = 10) # equivalent to tokens_sel

# Turn both of them into a DFM
dfmat_privacy <- dfm(toks_privacy)
dfmat_noprivacy <- dfm(toks_noprivacy)

# Calculate keyness beetween the two categories - the score for features that occur differentially
# across the two categories
tstat_key_privacy <- textstat_keyness(rbind(dfmat_privacy, dfmat_noprivacy), seq_len(ndoc(dfmat_privacy)
tstat_key_privacy_subset <- tstat_key_privacy[tstat_key_privacy$n_target > 10, ]
head(tstat_key_privacy_subset, 50)
```

```
##                          feature        chi2            p n_target n_reference
## 1                        privacy 14582.99461 0.000000e+00      734           0
## 3                     preserving   154.55254 0.000000e+00       13           6
## 4   covid-19_exposure_logging   129.44714 0.000000e+00       11           5
## 16                    assurances    88.92404 0.000000e+00       19          41
## 22                       default    73.76737 0.000000e+00       12          18
## 23                      settings    73.16860 0.000000e+00       21          61
## 25                        thinks    67.96699 1.110223e-16       13          24
## 31    information_commissioner    62.27760 2.997602e-15       15          36
## 43                      invasive    58.88863 1.665335e-14       17          49
## 51                           job    48.38537 3.501754e-12       21          90
## 52                        debate    46.83615 7.717715e-12       15          47
## 59               civil_liberties    45.70936 1.371658e-11       15          48
## 87                        design    39.35906 3.526137e-10       21         104
## 94                 controversial    37.47155 9.275609e-10       11          31
## 95                    assessment    37.47155 9.275609e-10       11          31
## 96                       concern    36.53211 1.501718e-09       18          80
## 98                       balance    36.18691 1.792706e-09       11          32
## 99         amnesty_international    36.18691 1.792706e-09       11          32
## 106                        legal    32.44274 1.227569e-08       19         100
## 114                   protecting    29.59496 5.324320e-08       13          52
## 133                  governments    28.15931 1.117289e-07       33         254
## 137                     designed    27.77797 1.360670e-07       17          92
## 138                    concerned    27.77797 1.360670e-07       17          92
## 139                    questions    27.42790 1.630614e-07       25         170
## 140                 human_rights    26.96172 2.075247e-07       23         151
## 144                      freedom    26.33593 2.869017e-07       12          49
## 147                      comment    25.31931 4.858208e-07       11          43
## 148          contact_tracing_apps    25.18626 5.205161e-07       18         107
## 231                  fundamental    21.15676 4.232052e-06       11          49
## 233                      ethical    20.54840 5.814232e-06       11          50
## 234             privacy_security    20.54840 5.814232e-06       11          50
## 243                         mean    19.38133 1.070485e-05       15          88
## 246                       choice    18.84684 1.416457e-05       11          53
## 250                     analysis    18.43820 1.755044e-05       16         106
## 269                location_data    17.26484 3.251487e-05       18         131
## 270                       expert    17.25818 3.262900e-05       14          84
## 271                     citizens    16.11703 5.954605e-05       20         158
## 286                        chair    15.06419 1.039163e-04       11          61
```

```
## 290                         comes    14.88770 1.141045e-04       19          152
## 291                          tech    14.82012 1.182667e-04       23          200
## 299                          risk    14.58678 1.338503e-04       40          426
## 301                        groups    14.43103 1.453872e-04       15          109
## 302                      concerns    14.27970 1.575552e-04       25          228
## 303                         apple    14.01793 1.810761e-04       24          217
## 307                           app    13.53313 2.343883e-04      133         1900
## 309                       believe    13.44512 2.456445e-04       16          124
## 352                         needs    12.75175 3.556764e-04       21          187
## 357                         often    12.44982 4.180310e-04       11           68
## 366                       systems    11.86525 5.719103e-04       17          143
## 369                           law    11.43750 7.197649e-04       21          195
```

These are the terms that are more linley to occur around the term "privacy" than anywhere else in the text.

4. Dictionary methods: Targeted sentiment analysis

You can use tokens_select() with window argument to perform *TARGETED* sentiment analysis.

Let's evaluate the sentiment around mentions of privacy and security.

```r
# Define keyterms
privacy <- c('privacy', 'security')

# Define relevant tokens - 20 tokens before and after the keyword
toks_privacy <- tokens_keep(toks_comp, pattern = phrase(privacy), window = 20)
toks_privacy
```

```
## Tokens consisting of 666 documents and 12 docvars.
## text1 :
##  [1] "involved"                          "NHSX's"
##  [3] "coronavirus_contact-tracing_app" "two"
##  [5] "efforts"                           "otherwise"
##  [7] "independent"                       "plans"
##  [9] "mix"                               "information_gathered"
## [11] "via_app"                           "data_store"
## [ ... and 70 more ]
##
## text2 :
## character(0)
##
## text3 :
## character(0)
##
## text4 :
##  [1] "unprecedented"  "collaborations" "government"     "tech_giants"
##  [5] "sounds"         "familiar"       "pre-Covid"      "precise"
##  [9] "app-driven"     "gig-fuelled"    "future"         "sold"
## [ ... and 170 more ]
##
## text5 :
##  [1] "said"                 "see"                  "think"
##  [4] "inevitable"           "view"                 "crucially"
##  [7] "planning"             "around"               "want"
```

```
## [10] "position"              "protective_equipment" "face_masks"
## [ ... and 29 more ]
##
## text6 :
## character(0)
##
## [ reached max_ndoc ... 660 more documents ]
```

```
# Put it into a dataframe matrix, defining the week as the grouping variable
dfmat_privacy_lsd <- dfm(toks_privacy, dictionary = data_dictionary_LSD2015[1:2]) %>%
  dfm_group(group = 'week', fill = TRUE)
dfmat_privacy_lsd
```

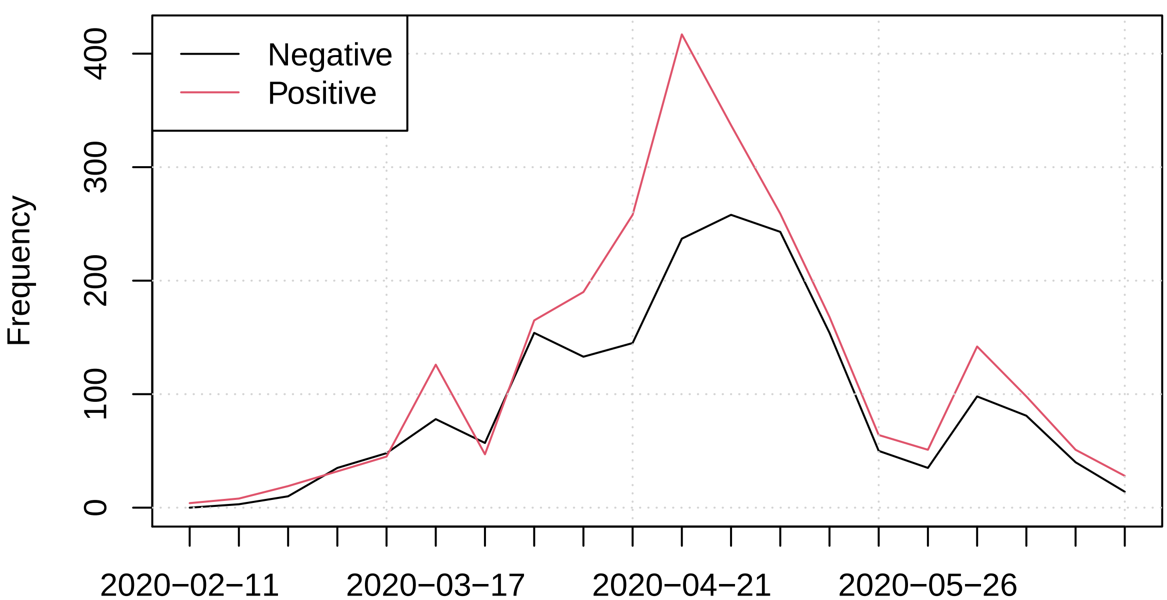
```
## Document-feature matrix of: 20 documents, 2 features (2.5% sparse) and 4 docvars.
##     features
## docs negative positive
##   6         0        4
##   9         3        8
##   10       10       19
##   11       35       32
##   12       48       45
##   13       78      126
## [ reached max_ndoc ... 14 more documents ]
```

Let's plot the results. Start with positive and negative sentiment.

```
# What is the minimum date?
min(art_uk$Date2)
```

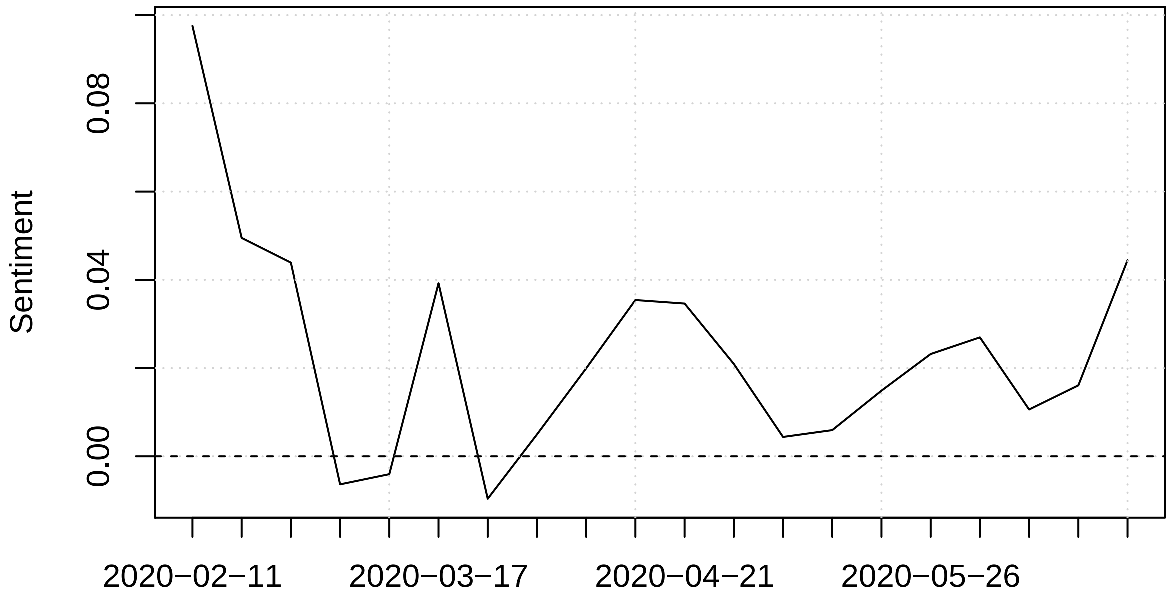
```
## [1] "2020-02-11"
```

```
matplot(dfmat_privacy_lsd, type = 'l', xaxt = 'n', lty = 1, ylab = 'Frequency')
grid()
axis(1, seq_len(ndoc(dfmat_privacy_lsd)), ymd("2020-02-11") + weeks(seq_len(ndoc(dfmat_privacy_lsd)) -
legend('topleft', col = 1:2, legend = c('Negative', 'Positive'), lty = 1, bg = 'white')
```



Now plot relative sentiment:

```r
n_eu <- ntoken(dfm(toks_privacy, group = toks_privacy$week))
plot((dfmat_privacy_lsd[,2] - dfmat_privacy_lsd[,1]) / n_eu,
     type = 'l', ylab = 'Sentiment', xlab = '', xaxt = 'n')
axis(1, seq_len(ndoc(dfmat_privacy_lsd)), ymd("2020-02-11") + weeks(seq_len(ndoc(dfmat_privacy_lsd)) -
grid()
abline(h = 0, lty = 2)
```

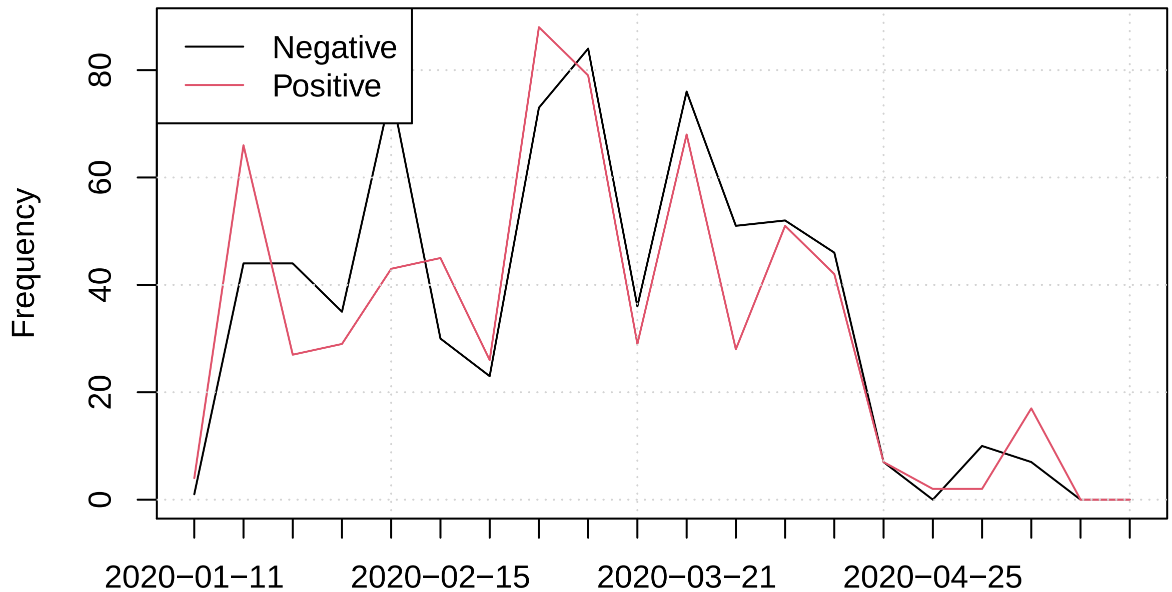


Exercise:

Work with your group to measure and plot the use of terms related to China and Wuhan in a window of 15 tokens before and after the keywords in this corpus. ##

```r
# Define the dictionary
gov <- c('china', 'China', 'chinese', 'Wuhan')
toks_gov <- tokens_keep(toks_comp, pattern = phrase(gov), window = 20)
dfmat_gov_lsd <- dfm(toks_gov, dictionary = data_dictionary_LSD2015[1:2]) %>%
  dfm_group(group = 'week', fill = TRUE)

matplot(dfmat_gov_lsd, type = 'l', xaxt = 'n', lty = 1, ylab = 'Frequency')
grid()
axis(1, seq_len(ndoc(dfmat_gov_lsd)), ymd("2020-01-11") + weeks(seq_len(ndoc(dfmat_gov_lsd)) - 1))
legend('topleft', col = 1:2, legend = c('Negative', 'Positive'), lty = 1, bg = 'white')
```




```
n_gov <- ntoken(dfm(toks_gov, group = toks_gov$week))
plot((dfmat_gov_lsd[,2] - dfmat_gov_lsd[,1]) / n_gov,
     type = 'l', ylab = 'Sentiment', xlab = '', xaxt = 'n')
axis(1, seq_len(ndoc(dfmat_gov_lsd)), ymd("2020-02-11") + weeks(seq_len(ndoc(dfmat_gov_lsd)) - 1))
grid()
abline(h = 0, lty = 2)
```

5. Dictionary methods: Which words contribute towards dictionary category counts?

```
toks_new <- tokens(head(corp_uk, 3))

dfm_list <- list()
```

```r
for (key in names(data_dictionary_LSD2015)) {
  this_dfm <- tokens_select(toks_new, data_dictionary_LSD2015[key], pad = TRUE) %>%
    tokens_compound(data_dictionary_LSD2015[key]) %>%
    tokens_replace("", "OTHER") %>%
    dfm(tolower = FALSE)
  dfm_list <- c(dfm_list, this_dfm)
}
names(dfm_list) <- names(data_dictionary_LSD2015)

dfm_list
```

```
## $negative
## Document-feature matrix of: 3 documents, 21 features (49.2% sparse) and 12 docvars.
##        features
## docs    risk problem shortages critical Dangerous warned dangerously alarm
##   text1    2       2         1        1         1      1           1     1
##   text2    0       0         0        0         0      1           0     0
##   text3    0       0         0        0         0      1           0     0
##        features
## docs    Emergencies concern
##   text1           1       1
##   text2           0       0
##   text3           0       0
## [ reached max_nfeat ... 11 more features ]
##
## $positive
## Document-feature matrix of: 3 documents, 40 features (55.0% sparse) and 12 docvars.
##        features
## docs    protective trusts help care innovation sense resources partner
##   text1          1      2    4    3          1     1         1       1
##   text2          0      0    2    0          0     0         0       0
##   text3          0      0    2    0          0     0         0       0
##        features
## docs    protectors efforts
##   text1          1       2
##   text2          0       0
##   text3          0       0
## [ reached max_nfeat ... 30 more features ]
##
## $neg_positive
## Document-feature matrix of: 3 documents, 1 feature (0.0% sparse) and 12 docvars.
##        features
## docs    OTHER
##   text1   789
##   text2   574
##   text3   586
##
## $neg_negative
## Document-feature matrix of: 3 documents, 1 feature (0.0% sparse) and 12 docvars.
##        features
## docs    OTHER
##   text1   789
##   text2   574
##   text3   586
```

7. Structural topic models

This method will only work on your own computer because of limited computer power provided by RStudio Cloud.

See the attached script “STM_example.R”.