# Week 5: Assignment 4: Word relationship analysis

Scout Leonard

2022-05-01

## Load Libraries

```
packages=c("tidyr",
           "pdftools",
           "lubridate",
           "tidyverse",
           "tidytext",
           "readr",
           "quanteda",
           "readtext",
           "quanteda.textstats",
           "quanteda.textplots",
           "ggplot2",
           "forcats",
           "stringr",
           "quanteda.textplots",
           "widyr",
           "igraph",
           "ggraph",
           "here")

for (i in packages) {
  if (require(i,character.only=TRUE)==FALSE) {
    install.packages(i,repos='http://cran.us.r-project.org')
  }
  else {
    require(i,character.only=TRUE)
  }
}
```

## Read in data

```
#filepath to data
files <- list.files(path = here("data/week5_data/"),
                    pattern = "*pdf$",
                    full.names = TRUE)

#renders all textboxes on a text canvas and returns a character vector of equal length to the number of
ej_reports <- lapply(files, pdf_text)
```

```r
#read texts and (if any) associated document-level meta-data from one or more source files - makes a df
ej_pdf <- readtext(file = here("data/week5_data/*.pdf"),
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "subj", "year"),
                   sep = "_")

#creating an initial corpus containing our data
epa_corp <- corpus(x = ej_pdf, text_field = "text" )

#return details of the corpus
summary(epa_corp)
```

```
## Corpus consisting of 6 documents, showing 6 documents:
##
##              Text Types Tokens Sentences type subj year
##   EPA_EJ_2015.pdf  2136   8944       263  EPA   EJ 2015
##   EPA_EJ_2016.pdf  1599   7965       176  EPA   EJ 2016
##   EPA_EJ_2017.pdf  2774  16658       447  EPA   EJ 2017
##   EPA_EJ_2018.pdf  3973  30564       653  EPA   EJ 2018
##   EPA_EJ_2019.pdf  3773  22648       672  EPA   EJ 2019
##   EPA_EJ_2020.pdf  4493  30523       987  EPA   EJ 2020
```

```r
#I'm adding some additional, context-specific stop words to stop word lexicon
more_stops <-c("2015","2016", "2017", "2018",
               "2019", "2020", "www.epa.gov", "https")

#add the additional stopwords to the stop word lexicon
add_stops<- tibble(word = c(stop_words$word, more_stops))

stop_vec <- as_vector(add_stops)
```

```r
#convert to tidy format and apply my stop words
raw_text <- tidy(epa_corp)

#Distribution of most frequent words across documents
raw_words <- raw_text %>%
  mutate(year = as.factor(year)) %>%
  unnest_tokens(word, text) %>%
  anti_join(add_stops, by = 'word') %>%
  count(year, word, sort = TRUE)

#number of total words by document
total_words <- raw_words %>%
  group_by(year) %>%
  summarize(total = sum(n))

report_words <- left_join(raw_words, total_words)
```

```
## Joining, by = "year"
```

```r
par_tokens <- unnest_tokens(raw_text,
                            output = paragraphs,
                            input = text,
                            token = "paragraphs")
```

```r
par_tokens <- par_tokens %>%
 mutate(par_id = 1:n())

par_words <- unnest_tokens(par_tokens,
                           output = word,
                           input = paragraphs,
                           token = "words")
```

## Part 1

What are the most frequent trigrams in the dataset? How does this compare to the most frequent bigrams? Which n-gram seems more informative here, and why?

```
tokens <- tokens(epa_corp, remove_punct = TRUE) #list of character vectors - takes each document and sp

toks1<- tokens_select(tokens, min_nchar = 3)

toks1 <- tokens_tolower(toks1)

toks1 <- tokens_remove(toks1, pattern = (stop_vec))

dfm <- dfm(toks1) #create document feature matrix - rows are number of occurances of each word within e

#first the basic frequency stat
tstat_freq <- textstat_frequency(dfm, n = 5, groups = year)

head(tstat_freq, 10) %>%
  knitr::kable()
```

| feature | frequency | rank | docfreq | group |
|---|---|---|---|---|
| environmental | 127 | 1 | 1 | 2015 |
| communities | 99 | 2 | 1 | 2015 |
| epa | 92 | 3 | 1 | 2015 |
| justice | 84 | 4 | 1 | 2015 |
| community | 47 | 5 | 1 | 2015 |
| environmental | 109 | 1 | 1 | 2016 |
| communities | 85 | 2 | 1 | 2016 |
| justice | 71 | 3 | 1 | 2016 |
| epa | 48 | 4 | 1 | 2016 |
| federal | 31 | 5 | 1 | 2016 |

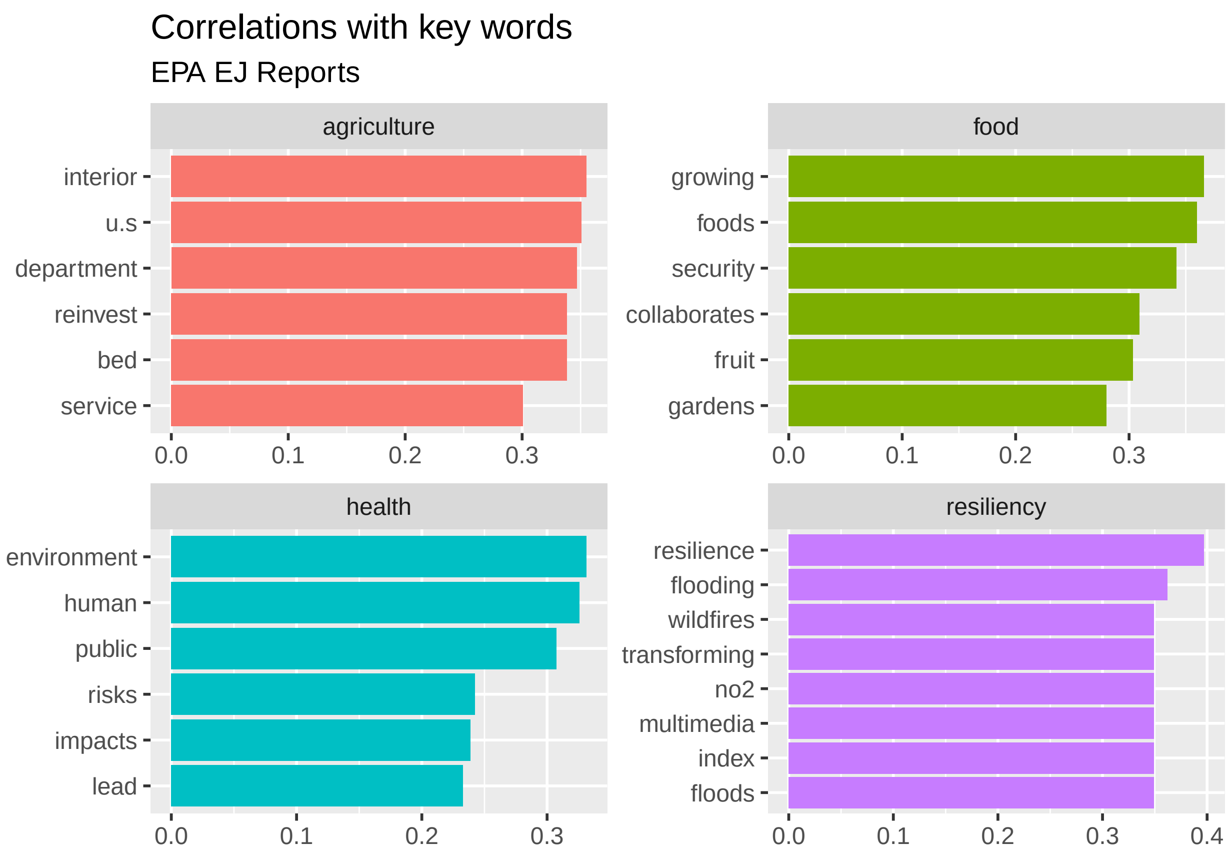
```
toks2 <- tokens_ngrams(toks1, n = 3)

dfm2 <- dfm(toks2)

dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))
#gives more coherent terms - power of chunking at a different token level

freq_words2 <- textstat_frequency(dfm2, n = 20)

freq_words2$token <- rep("trigram", 20)
#tokens1 <- tokens_select(tokens1,pattern = stopwords("en"), selection = "remove")
```

## Part 2

Choose a new focal term to replace "justice" and recreate the correlation table and network (see corr_paragraphs and corr_network chunks). Explore some of the plotting parameters in the cor_network chunk to see if you can improve the clarity or amount of information your plot conveys. Make sure to use a different color for the ties!

```r
#correlation between co-occuring words
word_cors <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)

#now we can select words cooccurring with the word justice and get correlation coefficients
food_cors <- word_cors %>%
  filter(item1 == "food")

  word_cors %>%
  filter(item1 %in% c("food", "agriculture", "health", "resiliency")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
  name = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(y = name, x = correlation, fill = item1)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~item1, ncol = 2, scales = "free")+
  scale_y_reordered() +
  labs(y = NULL,
         x = NULL,
         title = "Correlations with key words",
         subtitle = "EPA EJ Reports")
```

```
## Selecting by correlation
```

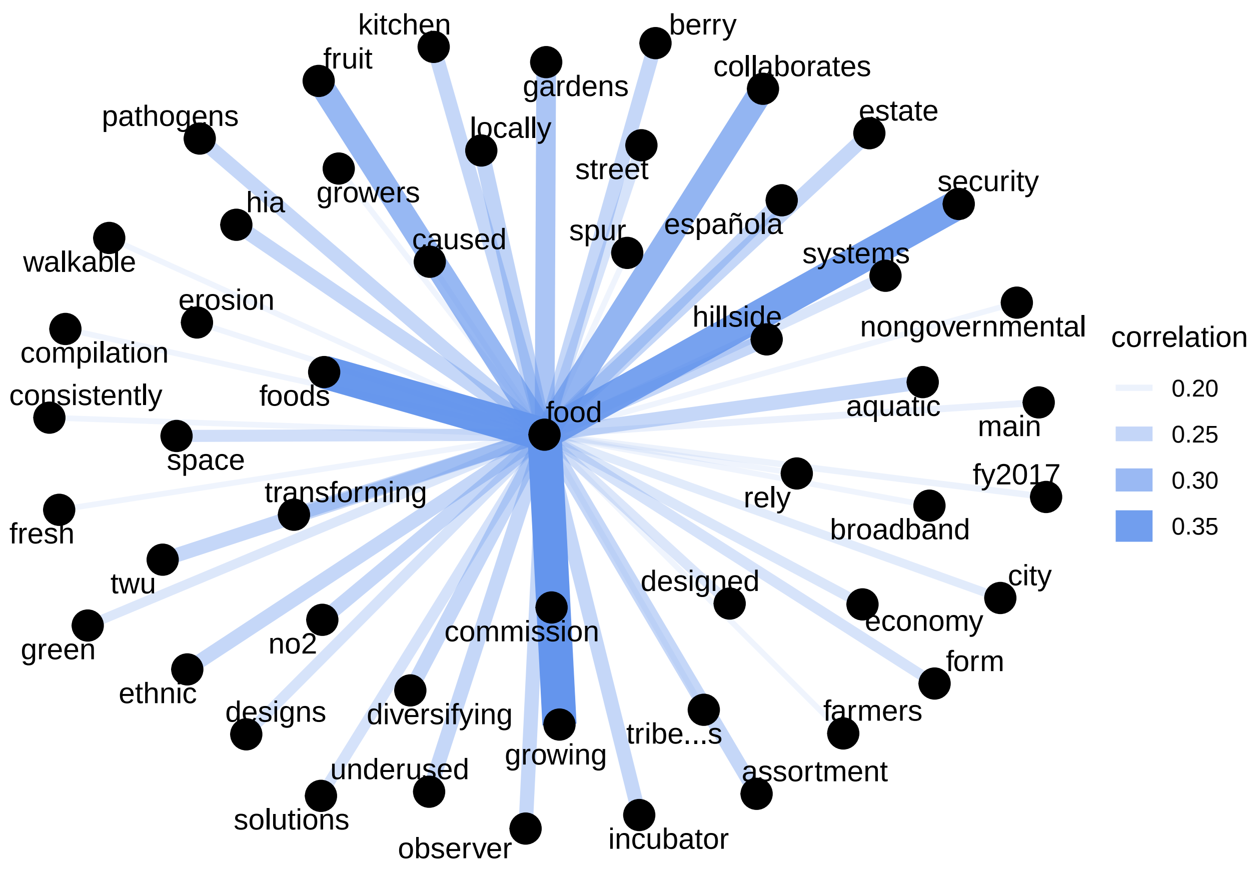


```
  #let's zoom in on just one of our key terms
   food_cors <- word_cors %>%
  filter(item1 == "food") %>%
   mutate(n = 1:n())
```

```
food_cors  %>%
  filter(n <= 50) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation,
                     edge_width = correlation),
                 edge_colour = "cornflowerblue") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name),
                 repel = TRUE,
                 point.padding = unit(0.2,
                                      "lines")) +
  theme_void()
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe's' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe's' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe's' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

```
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'tribe’s' in 'mbcsToSbcs': dot substituted for <99>
```



## Part 3

Write a function that allows you to conduct a keyness analysis to compare two individual EPA reports (hint: that means target and reference need to both be individual reports). Run the function on 3 pairs of reports, generating 3 keyness plots.

```
#keyness_comparison <- function()
```

## Part 4

Select a word or multi-word term of interest and identify words related to it using windowing and keyness comparison. To do this you will create to objects: one containing all words occurring within a 10-word window of your term of interest, and the second object containing all other words. Then run a keyness comparison on these objects. Which one is the target, and which the reference? Hint