Topic 05 Word Reference

Julia Parish

2022-05-03

Word Reference

This text sentiment analysis was completed as an assignment for the course, Environmental Data Science 231: Text and Sentiment Analysis for Environmental Problems. The data was sourced from articles written by the Environmental Protection Agency.

Original assignment instructions can be found here

Load Libraries

```
library(tidyr) #text analysis in R
library(pdftools)
library(lubridate) #working with date data
library(tidyverse)
library(tidytext)
library(readr)
library(quanteda)
library(readtext) #quanteda subpackage for reading pdf
library(quanteda.textstats)
library(quanteda.textplots)
library(ggplot2)
library(forcats)
library(stringr)
library(quanteda.textplots)
library(widyr)# pairwise correlations
library(igraph) #network plots
library(ggraph)
library(here)
```

```
setwd("/Users/julia/Documents/_MEDS/04_spring/EDS231_TextSentiment/repository/EDS231_TextSentimentAnalys
```

Assignment Set Up

Read in data files, clean the data, create objects, and conduct frequency statistics

Load Data Files

```
files <- list.files(path = "data/", pattern = "pdf$", full.names = T)

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext(file = "data/*.pdf", docvarsfrom = "filenames",
                   docvarnames = c("type", "subj", "year"),
                     sep = "_")

#create an initial corpus containing the EPA EJ data
epa_corp <- corpus(x = ej_pdf, text_field = "text" )
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
```

Add Stop Words

```
# add context-specific stop words to stop word lexicon
more_stops <-c("2015","2016", "2017", "2018", "2019", "2020", "www.epa.gov", "https")

add_stops<- tibble(word = c(stop_words$word, more_stops))

stop_vec <- as_vector(add_stops)
```

Create different data objects for the subsequent analyses

```
#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)

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

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

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

```
tokens <- tokens(epa_corp, remove_punct = TRUE)
toks1<- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec))
dfm <- dfm(toks1)
```

Conduct Frequency Statistics

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

head(tstat_freq, 10) %>%
  knitr::kable(caption = "Subset of Top 10 Words")
```

Table 1: Subset of Top 10 Words

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

Assignment Questions

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?

```r
# bigrams
toks2 <- tokens_ngrams(toks1, n=2)
dfm2 <- dfm(toks2) # document feature matrix
dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))

freq_words2 <- textstat_frequency(dfm2, n=20)
freq_words2$token <- rep("bigram", 20)

bigrams <- freq_words2 %>%
  knitr::kable(caption = "Bigrams")

bigrams
```

Table 2: Bigrams

feature	frequency	rank	docfreq	group	token
environmental_justice	556	1	6	all	bigram
technical_assistance	139	2	6	all	bigram
drinking_water	133	3	6	all	bigram
public_health	123	4	6	all	bigram
progress_report	108	5	6	all	bigram
air_quality	73	6	6	all	bigram
water_systems	66	7	6	all	bigram
vulnerable_communities	65	8	6	all	bigram
epa_region	62	9	5	all	bigram
environmental_public	57	10	6	all	bigram
federal_agencies	56	11	6	all	bigram
national_environmental	51	12	6	all	bigram
justice_fy2017	51	12	1	all	bigram
fy2017_progress	51	12	1	all	bigram
superfund_sites	48	15	4	all	bigram
indigenous_peoples	46	16	6	all	bigram
civil_rights	46	16	5	all	bigram
local_governments	45	18	6	all	bigram
urban_waters	44	19	6	all	bigram
overburdened_communities	43	20	6	all	bigram

```r
# trigrams
toks3 <- tokens_ngrams(toks1, n=3)
dfm3 <- dfm(toks3) # document feature matrix
dfm3 <- dfm_remove(dfm3, pattern = c(stop_vec))

freq_words3 <- textstat_frequency(dfm3, n=20)
freq_words3$token <- rep("trigram", 20)

trigrams <- freq_words3 %>%
  knitr::kable(caption = "Trigrams")

trigrams
```

Table 3: Trigrams

feature	frequency	rank	docfreq	group	token
justice_fy2017_progress	51	1	1	all	trigram
fy2017_progress_report	51	1	1	all	trigram
environmental_public_health	50	3	6	all	trigram
environmental_justice_fy2017	50	3	1	all	trigram
national_environmental_justice	37	5	6	all	trigram
office_environmental_justice	32	6	6	all	trigram
epa's_environmental_justice	32	6	6	all	trigram
environmental_justice_progress	30	8	4	all	trigram
justice_progress_report	30	8	4	all	trigram
environmental_justice_concerns	30	8	5	all	trigram
drinking_water_systems	29	11	5	all	trigram
annual_environmental_justice	27	12	5	all	trigram
environmental_justice_advisory	27	12	6	all	trigram
fiscal_annual_environmental	25	14	3	all	trigram
justice_advisory_council	24	15	6	all	trigram
environmental_justice_grants	22	16	5	all	trigram
technical_assistance_communities	20	17	6	all	trigram
communities_environmental_justice	20	17	5	all	trigram
safe_drinking_water	19	19	5	all	trigram
technical_assistance_services	19	19	5	all	trigram

The five most frequent bigrams are environmental_justice, technical_assistance, drinking_water, public_health, and progress_report.

The five most frequent trigrams are justice_fy2017_progress, fy2017_progress_report, environmental_public_health, environmental_justice_fy2017, and national_environmental_justice.

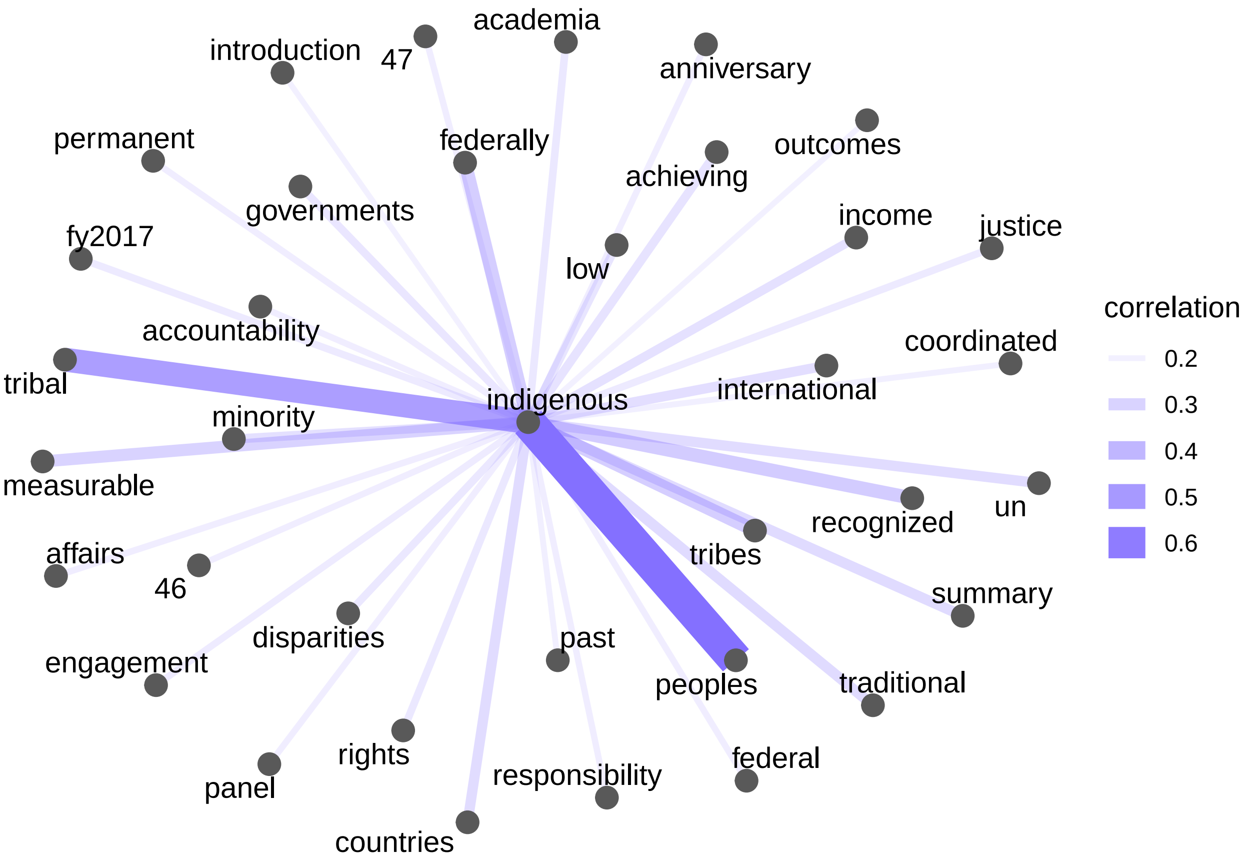
The words `environmental`, `justice`, `water`, `progress`, and `epa` appear frequently in both the bigrams and trigrams lists. The `bigrams` list provides more detailed, diverse words relevant to EPA policy. The `trigrams` list focuses more on progress report tokens than policy terms.

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!

```
# pairwise correlation

word_cors <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

```
# filter for the term 'indigenous'
indigenous_cors <- word_cors %>%
  filter(item1 == "indigenous") %>%
  mutate(n = 1:n())
```

```
# create correlation network

cor_network <- indigenous_cors  %>%
  filter(n <= 35) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "lightslateblue
  geom_node_point(color = "grey35", size = 3.5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()

cor_network
```



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.

Create the function

```
keyness_function <- function(reference_report, target_report) {
  files <- list.files(path = "data/", pattern = "pdf$", full.names = T)
  ej_reports <- lapply(files, pdf_text)
  ej_pdf <- readtext(file = "data/*.pdf", docvarsfrom = "filenames",
                    docvarnames = c("type", "subj", "year"),
                    sep = "_")
  epa_corp <- corpus(x = ej_pdf, text_field = "text")
  tokens <- tokens(epa_corp, remove_punct = TRUE)
  toks1<- tokens_select(tokens, min_nchar = 3)
  toks1 <- tokens_tolower(toks1)
  toks1 <- tokens_remove(toks1, pattern = (stop_vec))
  dfm <- dfm(toks1)

  keyness_function_plot <- dfm %>%
    dfm_subset(year %in% c(reference_report, target_report)) %>%
    textstat_keyness(target = paste0("EPA_EJ_", target_report, ".pdf")) %>%
    textplot_keyness()
  keyness_function_plot
  }
```

Use function to analyze EPA Reports 2015 & 2016

```
keyness_function(reference_report = 2015, target_report = 2016)
```

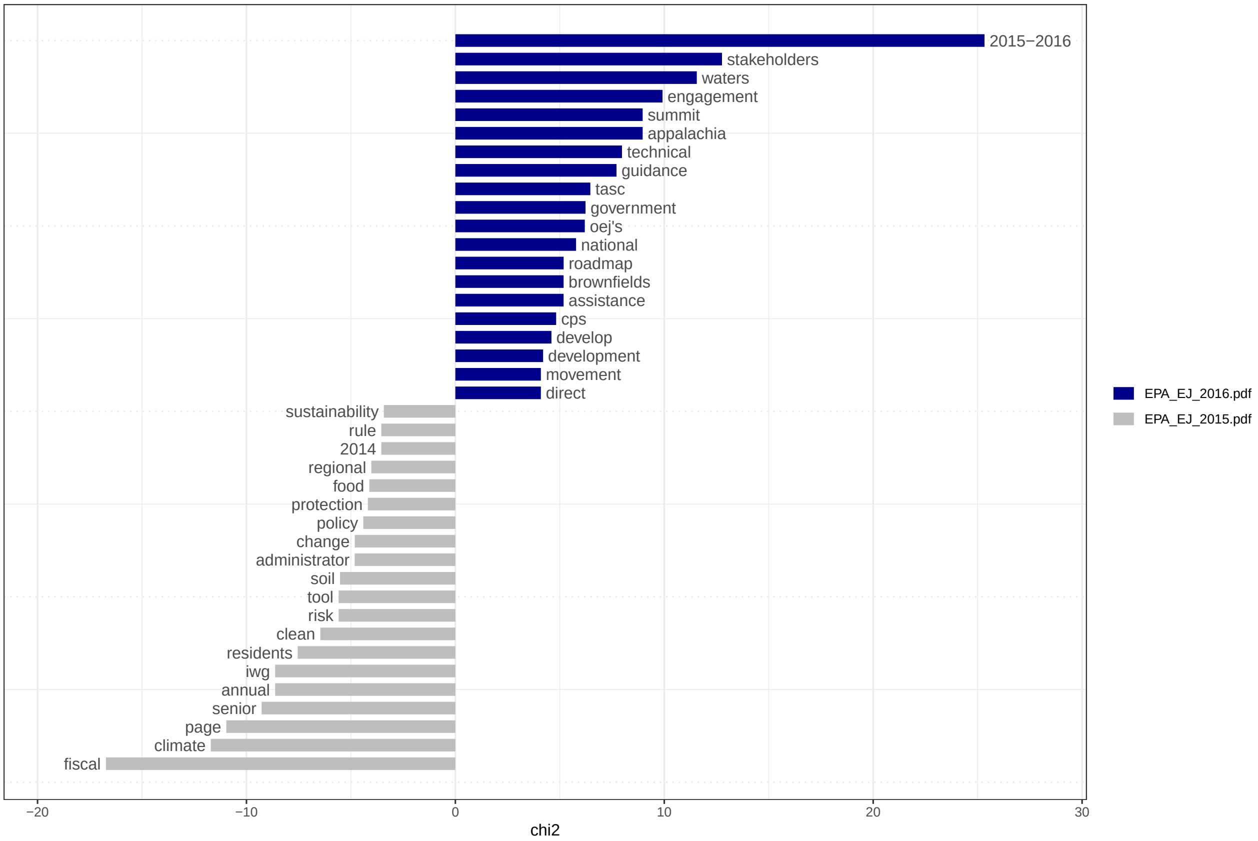


Figure 1: Analysis of most frequent terms in the reference file, EPA FY2015, and target file, EPA FY2016.

Analyze EPA Reports 2016 & 2017

```
keyness_function(reference_report = 2016, target_report = 2017)
```

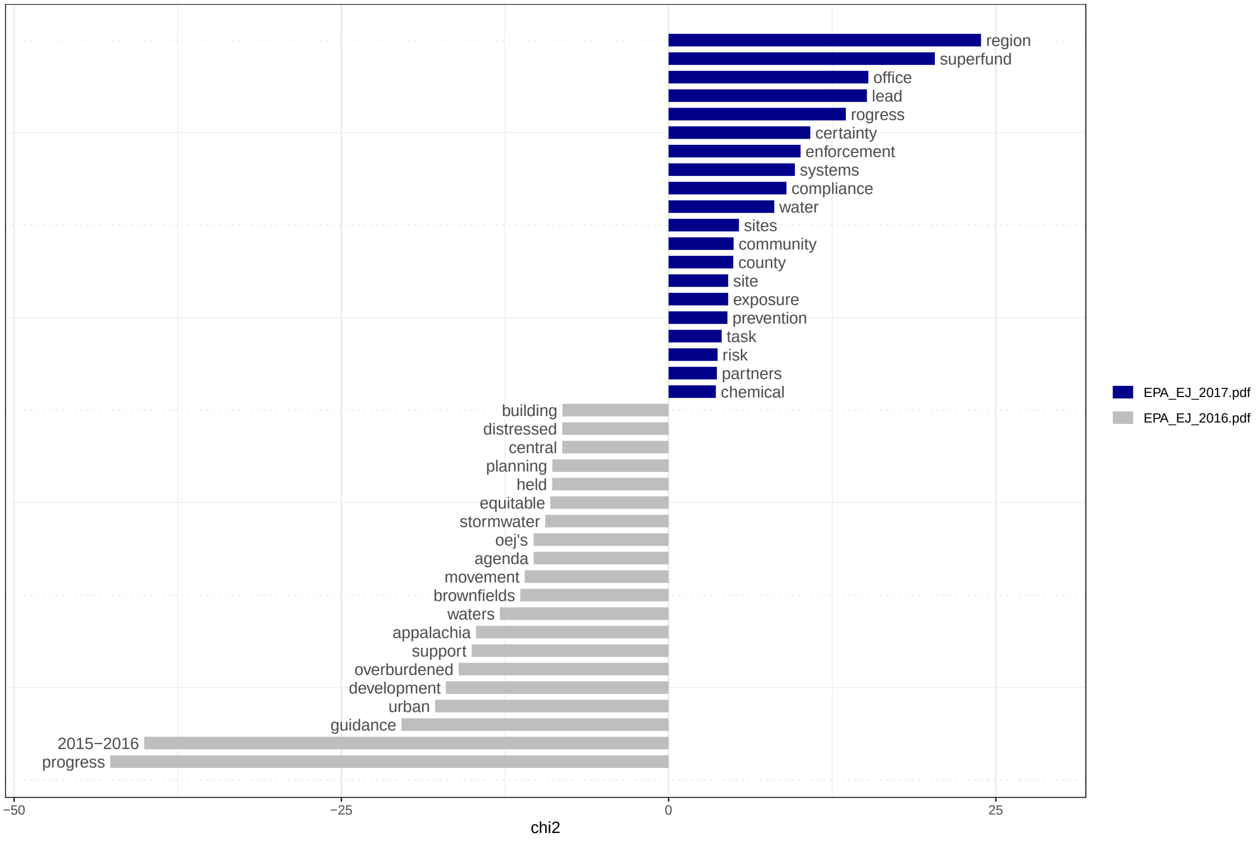


Figure 2: Analysis of most frequent terms in the reference file, EPA FY2016, and target file, EPA FY2017.

Analyze EPA Reports 2017 & 2018

```
keyness_function(reference_report = 2017, target_report = 2018)
```

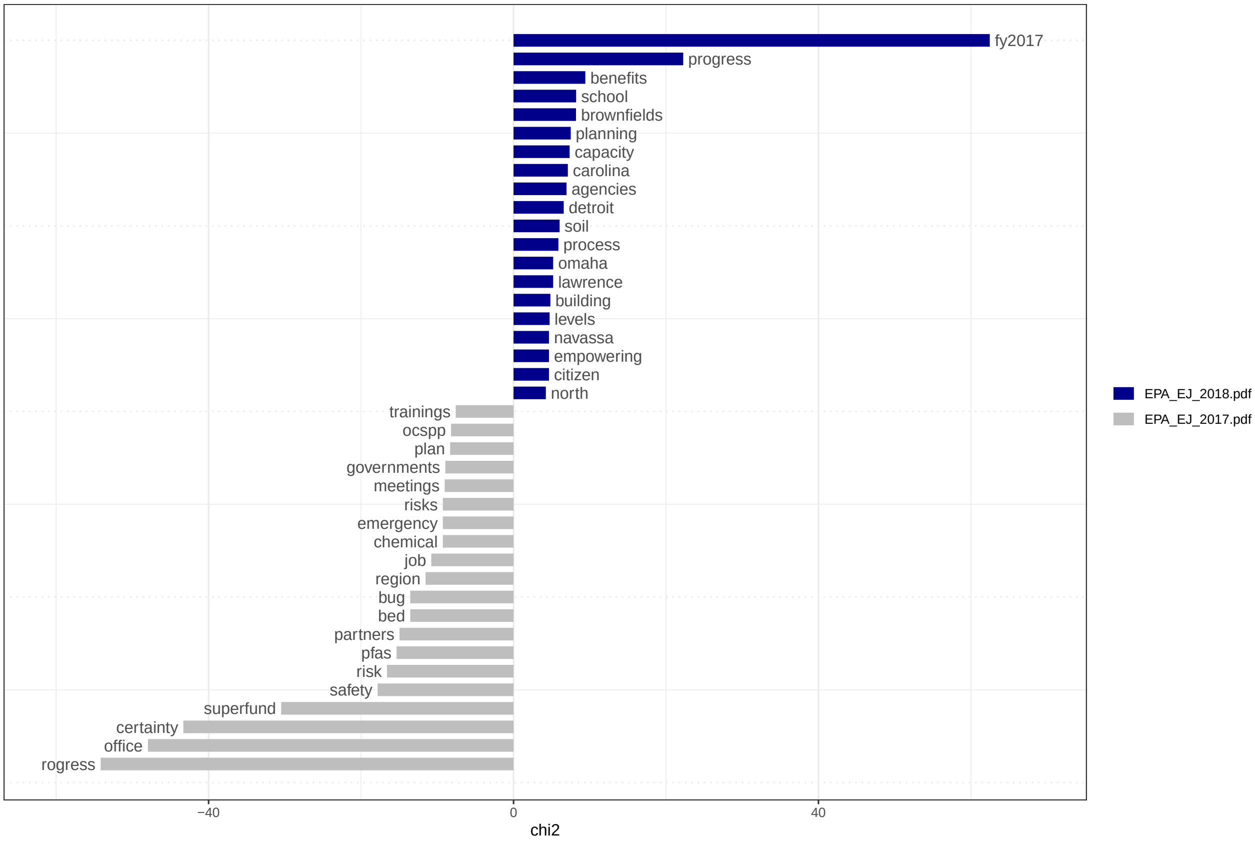


Figure 3: Analysis of msot frequent terms in the reference file, EPA FY2017, and target file, EPA FY2018.

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 two 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

```
tokens <- tokens(epa_corp, remove_punct = TRUE)
toks1<- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec))
dfm <- dfm(toks1)
```

```
toks_inside <- tokens_keep(toks1, pattern = "indigenous", window = 10)

# remove the keywords
toks_inside <- tokens_remove(toks_inside, pattern = "indigenous")

toks_outside <- tokens_remove(toks1, pattern = "indigenous", window = 10)
```

```
dfmat_inside <- dfm(toks_inside)
dfmat_outside <- dfm(toks_outside)

tstat_key_inside <- textstat_keyness(rbind(dfmat_inside, dfmat_outside),
                                     target = seq_len(ndoc(dfmat_inside)))

head(tstat_key_inside, 10)
```

```
##        feature       chi2 p n_target n_reference
## 1      peoples 1262.56075 0       49           0
## 2   recognized  309.16345 0       19           9
## 3       tribes  248.78569 0       38          86
## 4    federally  207.86257 0       13           6
## 5       tribal  166.00369 0       47         200
## 6     minority  159.91760 0       25          57
## 7  governments  133.04273 0       22          53
## 8   low-income  119.84064 0       23          65
## 9          usg   96.04113 0        6           2
## 10    academia   76.27578 0        9          13
```