Keyword Analysis of Regenerative Agriculture in the NY Times

Benjamin Moscona

4/6/2022

```
library(jsonlite) #convert results from API queries into R-friendly formats
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
```

```
library(tidytext) #text data management and analysis
library(ggplot2) #plot word frequencies and publication dates
library(corpus)
```

```
#create an object called x with the results of our query ("regenerative agriculture")
# the from JSON flatten the JSON object, then convert to a data frame
t <- fromJSON("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=regenerative+agriculture&api-k
```

```
class(t) #what type of object is x?
```

```
## [1] "list"
```

```
t <- t %>%
  data.frame()


#Inspect our data
class(t) #now what is it?
```

```
## [1] "data.frame"
```

```
dim(t) # how big is it?
```

```
## [1] 10 33
```

```
names(t) # what variables are we working with?
```

```
##  [1] "status"
##  [2] "copyright"
##  [3] "response.docs.abstract"
##  [4] "response.docs.web_url"
##  [5] "response.docs.snippet"
```

```
##  [6] "response.docs.lead_paragraph"
##  [7] "response.docs.source"
##  [8] "response.docs.multimedia"
##  [9] "response.docs.keywords"
## [10] "response.docs.pub_date"
## [11] "response.docs.document_type"
## [12] "response.docs.news_desk"
## [13] "response.docs.section_name"
## [14] "response.docs.type_of_material"
## [15] "response.docs._id"
## [16] "response.docs.word_count"
## [17] "response.docs.uri"
## [18] "response.docs.subsection_name"
## [19] "response.docs.print_section"
## [20] "response.docs.print_page"
## [21] "response.docs.headline.main"
## [22] "response.docs.headline.kicker"
## [23] "response.docs.headline.content_kicker"
## [24] "response.docs.headline.print_headline"
## [25] "response.docs.headline.name"
## [26] "response.docs.headline.seo"
## [27] "response.docs.headline.sub"
## [28] "response.docs.byline.original"
## [29] "response.docs.byline.person"
## [30] "response.docs.byline.organization"
## [31] "response.meta.hits"
## [32] "response.meta.offset"
## [33] "response.meta.time"
```

```
#t <- readRDS("nytDat.rds") #in case of API emergency :)
```

```
term <- "Regenerative+Agriculture" # Need to use + to string together separate words
begin_date <- "19900120"
end_date <- "20220401"
pk <- "pQyo4zhAeOpIUtea56uCrw7MiDIbIlKh"

#construct the query url using API operators
baseurl <- paste0("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=",term,
                  "&begin_date=",begin_date,"&end_date=",end_date,
                  "&facet_filter=true&api-key=", pk, sep="")

#examine our query url
```

```
#this code allows for obtaining multiple pages of query results
initialQuery <- fromJSON(baseurl)
maxPages <- round((initialQuery$response$meta$hits[1] / 10)-1)

pages <- list()
for(i in 0:maxPages){
  nytSearch <- fromJSON(paste0(baseurl, "&page=", i), flatten = TRUE) %>% data.frame()
  message("Retrieving page ", i)
  pages[[i+1]] <- nytSearch
  Sys.sleep(6)
}
```

```
## Retrieving page 0
## Retrieving page 1
## Retrieving page 2
## Retrieving page 3
## Retrieving page 4
## Retrieving page 5
## Retrieving page 6
## Retrieving page 7
## Retrieving page 8
## Retrieving page 9
## Retrieving page 10
## Retrieving page 11
## Retrieving page 12
## Retrieving page 13
## Retrieving page 14
## Retrieving page 15
## Retrieving page 16
## Retrieving page 17
## Retrieving page 18
## Retrieving page 19
## Retrieving page 20
## Retrieving page 21
## Retrieving page 22
## Retrieving page 23
## Retrieving page 24
## Retrieving page 25
## Retrieving page 26
## Retrieving page 27
```

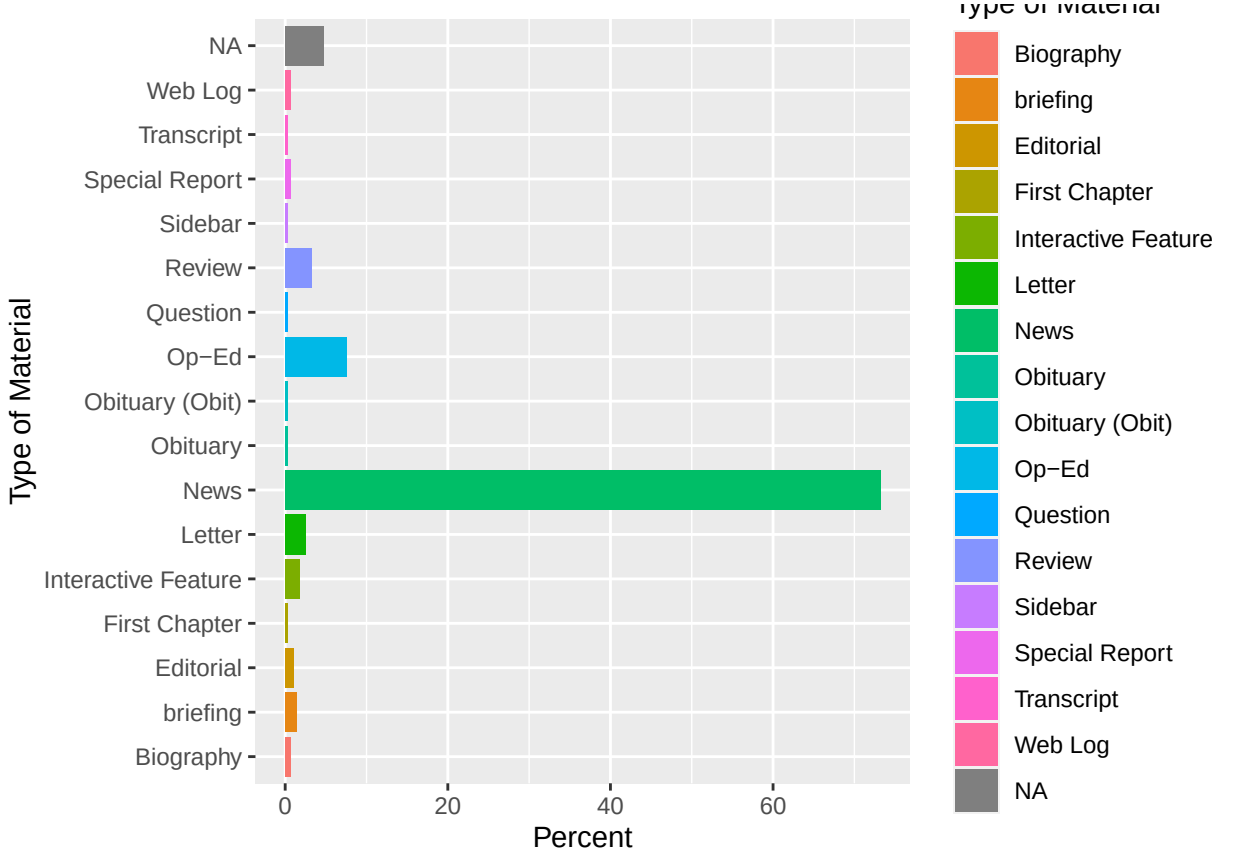
```
class(nytSearch)
```

```
## [1] "data.frame"
```

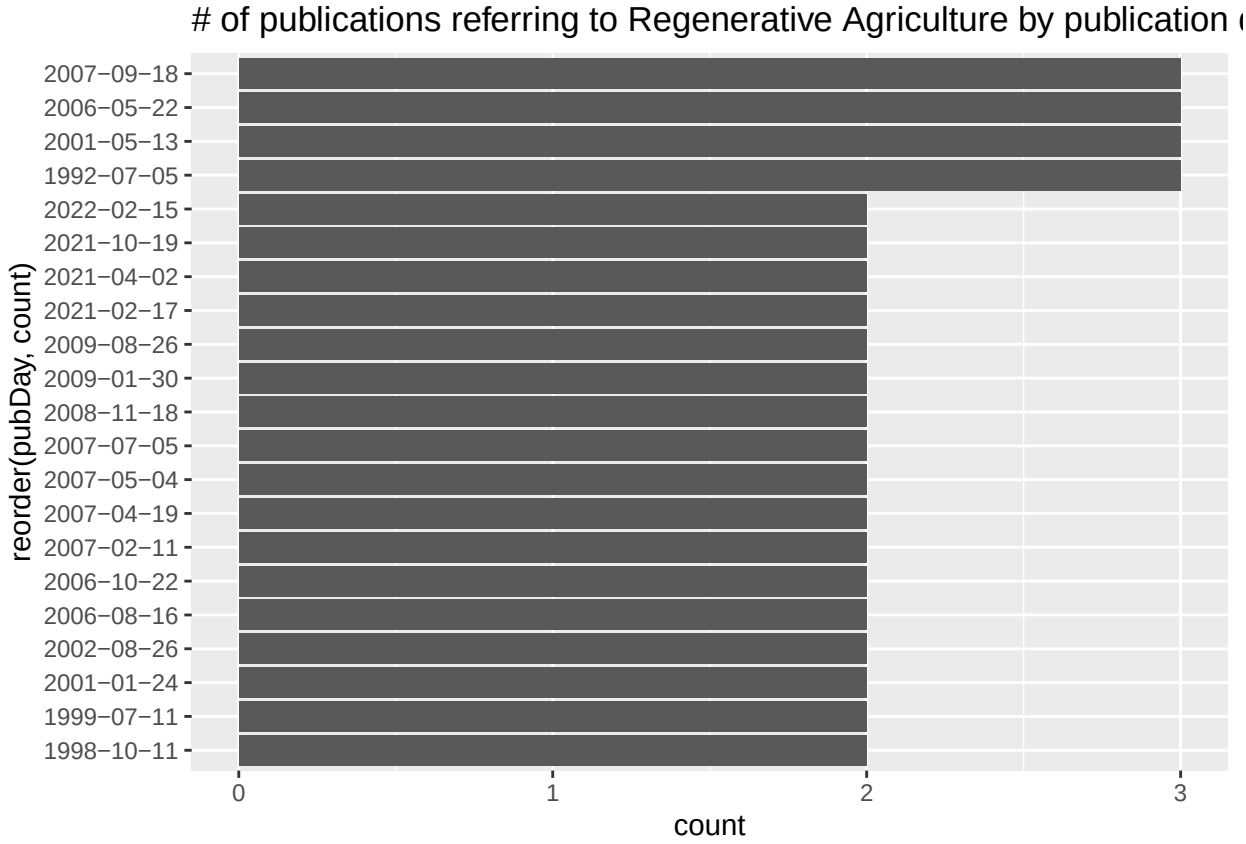
```
#need to bind the pages and create a tibble from nytDa
nytSearch <- rbind_pages(pages)
```

```
nytSearch %>%
  group_by(response.docs.type_of_material) %>%
  summarize(count=n()) %>%
  mutate(percent = (count / sum(count))*100) %>%
  ggplot() +
  geom_bar(aes(y=percent,x=response.docs.type_of_material,
```

```
            fill=response.docs.type_of_material), stat = "identity") + coord_flip() +
  labs(x = "Type of Material", y = "Percent", fill = "Type of Material")
```




```
nytSearch %>%
  mutate(pubDay=gsub("T.*","",response.docs.pub_date)) %>%
  group_by(pubDay) %>%
  summarise(count=n()) %>%
  filter(count >= 2) %>%
  ggplot() +
  geom_bar(aes(x=reorder(pubDay, count), y=count), stat="identity") + coord_flip() +
  labs(title = "# of publications referring to Regenerative Agriculture by publication day")
```




```
names(nytSearch)
```

```
##  [1] "status"
##  [2] "copyright"
##  [3] "response.docs.abstract"
##  [4] "response.docs.web_url"
##  [5] "response.docs.snippet"
##  [6] "response.docs.lead_paragraph"
##  [7] "response.docs.source"
##  [8] "response.docs.multimedia"
##  [9] "response.docs.keywords"
## [10] "response.docs.pub_date"
## [11] "response.docs.document_type"
## [12] "response.docs.news_desk"
## [13] "response.docs.section_name"
## [14] "response.docs.type_of_material"
## [15] "response.docs._id"
## [16] "response.docs.word_count"
## [17] "response.docs.uri"
## [18] "response.docs.subsection_name"
## [19] "response.docs.print_section"
## [20] "response.docs.print_page"
## [21] "response.docs.headline.main"
## [22] "response.docs.headline.kicker"
## [23] "response.docs.headline.content_kicker"
## [24] "response.docs.headline.print_headline"
## [25] "response.docs.headline.name"
```

```
## [26] "response.docs.headline.seo"
## [27] "response.docs.headline.sub"
## [28] "response.docs.byline.original"
## [29] "response.docs.byline.person"
## [30] "response.docs.byline.organization"
## [31] "response.meta.hits"
## [32] "response.meta.offset"
## [33] "response.meta.time"
```

Now, we look at word frequency using the first paragraph. Before we do that, let's remove stopwords, remove numbers, and stem some words

```
paragraph <- names(nytSearch)[6] #The 6th column, "response.doc.lead_paragraph", is the one we want her
tokenized <- nytSearch %>%
  unnest_tokens(word, paragraph)

data(stop_words)

tokenized <- tokenized %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

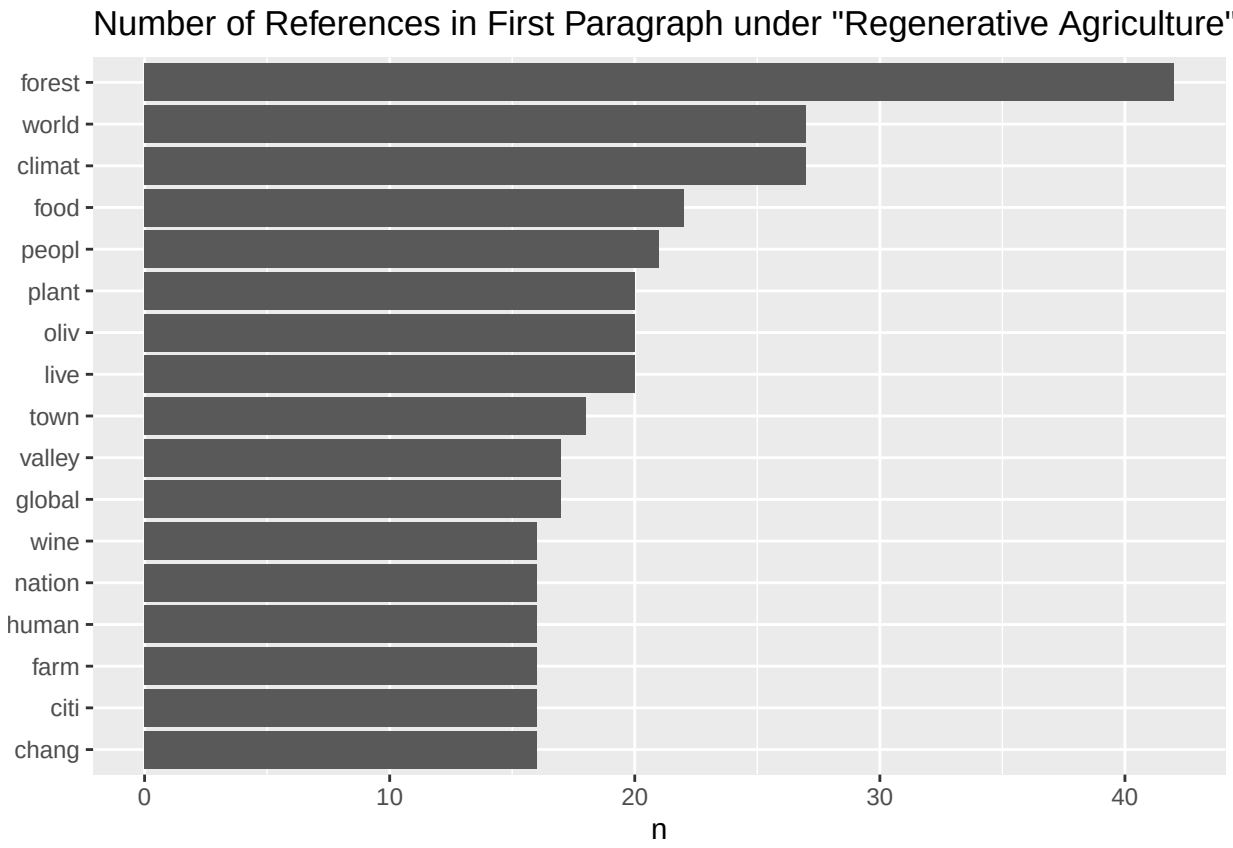
```
clean_tokens <- str_replace_all(tokenized$word,"environment[a-z A-Z]*","environment") #stem
clean_tokens <- str_replace_all(tokenized$word,"forest[a-z A-Z]*","forest")
clean_tokens <- str_replace_all(tokenized$word,"soil[a-z A-Z]*","soil")
clean_tokens <- text_tokens(tokenized$word, stemmer = "en")
clean_tokens <- str_remove_all(clean_tokens, "[:digit:]") #remove all numbers
clean_tokens <- gsub("'s", '', clean_tokens)

tokenized$clean <- clean_tokens

#remove the empty strings
tib <-subset(tokenized, clean!="")

#reassign
tokenized <- tib

tokenized %>%
  count(clean, sort = TRUE) %>%
  filter(n > 15) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = NULL, title = "Number of References in First Paragraph under \"Regenerative Agriculture\"")
```



Now, let's do the same thing but for just the headlines.

```
title <- "response.docs.headline.main" #picking the headline now
tokenized <- nytSearch %>%
  unnest_tokens(word, title)

data(stop_words)

tokenized <- tokenized %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

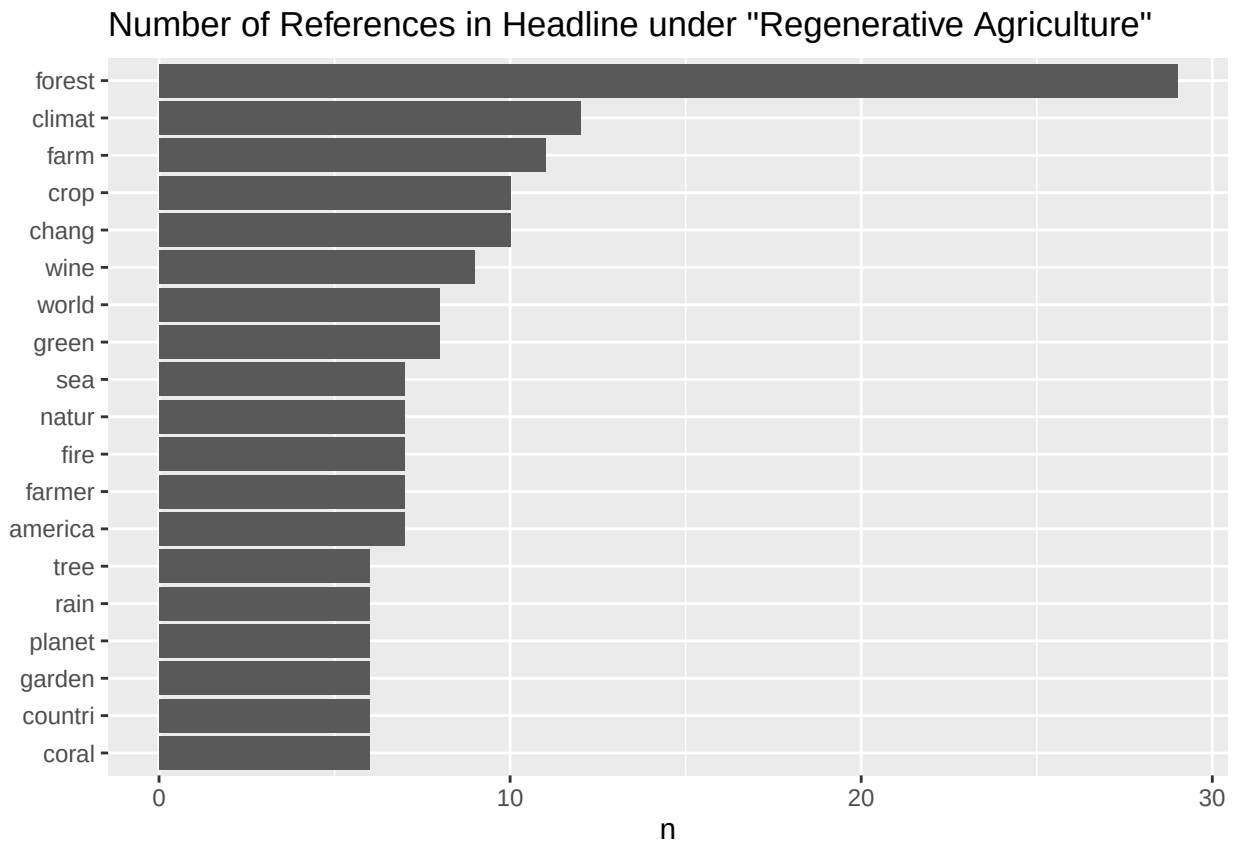
```
clean_tokens <- str_replace_all(tokenized$word,"environment[a-z A-Z]*","environment") #stem
clean_tokens <- str_replace_all(tokenized$word,"forest[a-z A-Z]*","forest")
clean_tokens <- str_replace_all(tokenized$word,"soil[a-z A-Z]*","soil")
clean_tokens <- text_tokens(tokenized$word, stemmer = "en")
clean_tokens <- str_remove_all(clean_tokens, "[:digit:]") #remove all numbers
clean_tokens <- gsub("'s", '', clean_tokens)

tokenized$clean <- clean_tokens

#remove the empty strings
tib <-subset(tokenized, clean!="")

#reassign
tokenized <- tib
```

```
tokenized %>%
  count(clean, sort = TRUE) %>%
  filter(n > 5) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = NULL, title = "Number of References in Headline under \"Regenerative Agriculture\"")
```



If we look at these two word distributions side-by-side, we notice that forests and climate are still two of the dominant words. "Farm" is much more dominant in the headline than in the first paragraph, whereas "food" is much more dominant in the first paragraph than the headline. Interestingly, the word "America" is present in seven headlines but has a lower distribution in the first paragraphs. It's interesting to consider headlines versus first paragraphs, especially considering the use of clickbait headlines with emotional or group appeals. Surprisingly, wine gets mentioned quite often in both headlines and first paragraphs, which makes me think I should explore the popularity of regenerative farming techniques in viticulture. Perhaps, wine producers are leaders in regenerative ag, or it is just a popular thing to report on.