Web Scraping Congressional Leaders' Press Releases

Samuel Jens

11/14/2021

Introduction

Let's begin by web scraping press releases of members of Congress (MCs). [NOTE: Only web scrape publicly available data.] For the current project, I focus on the Speaker of the House, Nancy Pelosi (D-CA), and the minority leader, Kevin McCarthy (R-CA). I examine these two members to understand the language and topics each use since January 2021.

There are a few helpful and necessary tools to web scrape. The first is the *rvest* package in R. The second is the "selector gadget" extension for the Chrome internet browser. This extension provides users with the specific HTML tag for sections of individual websites. The information (text) held in these section tags is collected through web scraping.

Through exploring their press releases, I also apply several text-based machine learning approaches. I use variations of latent Dirichlet allocation (LDA) to uncover differences in each party's House leader over time.

House Speaker Nancy Pelosi's Press Releases

The code below shows how to scrape Nancy Pelosi's press releases. Her press releases use two different HTML tags for the main body text. I do not know if this is unique to Nancy Pelosi's website (as she is the Speaker), but I had to combine the two sections.

The first part of web scraping is sending the software to a website. The benefit of Congressional press releases is that they are indexed similarly via each member's *house.gov* website. However, the tags that identify the text can be different. For the current project, I collect the URL link, title, date, and statement for every press release. The code below shows all of the steps for reaching a website, locating the page, collecting the desired information, searching through all the pre-determined pages, and finally binding it together.

```
# Gather the text from each web page/release
get_text <- function(pr_links){
  pr_page = read_html(pr_links)
  pr_text = pr_page %>% html_nodes("#block-system-main td") %>% html_text() %>%
    paste(collapse = "")
  return(pr_text)
}

get_text2 <- function(pr_links){
  pr_page = read_html(pr_links)
  pr_text2 = pr_page %>% html_nodes(".even div~ div+ div") %>% html_text() %>%
    paste(collapse = "")
  return(pr_text2)
}

# Create empty dataset to store information
pelosi_pr <- data.frame(NULL)

for(i in seq(from = 0, to = 34, by = 1)){

# Provide the base url for all the press releases
base_url = paste0("https://pelosi.house.gov/news/press-releases?page=",{i})

# Read the base_url
page = read_html(base_url)

# Use "inspector gadget" tool in Chrome browser to find title, date, hyperlink, etc.
title = page %>% html_nodes(".views-field-title a") %>% html_text()
date = page %>% html_nodes(".views-field-created .field-content") %>% html_text()

pr_links = page %>% html_nodes(".views-field-title a") %>%
  html_attr("href") %>% paste0("https://pelosi.house.gov", .)
```

```
# Apply the function created above to all the links
statement1 = sapply(pr_links, FUN = get_text, USE.NAMES = FALSE)
statement2 = sapply(pr_links, FUN = get_text2, USE.NAMES = FALSE)

# Create a data frame to store all of the press release information
pelosi_pr = rbind(pelosi_pr,
                  data.frame(title, date, pr_links, statement1, statement2,
                             stringsAsFactors = FALSE))

# Show pages
#print(paste("Page:",{i}))

}
```

After web scraping Nancy Pelosi's press releases, the text requires cleaning. I also create a date variable that is formatted correctly for R to recognize going forward.

The text models are coded using the *quanteda* package. I remove any symbols, punctuation, separators, and numbers when creating the document feature matrix (dfm). I also remove stopwords, or words that are common and required to form complete sentences (e.g., of, and, the, etc.).

Below are the top 15 words across Pelosi's 350 press releases since January 14th. All of her top language relates to either the legislative process (e.g., president, legislation, bill) or to her responsibility as a representative and Speaker (e.g., speaker, people, thank, know, and house).

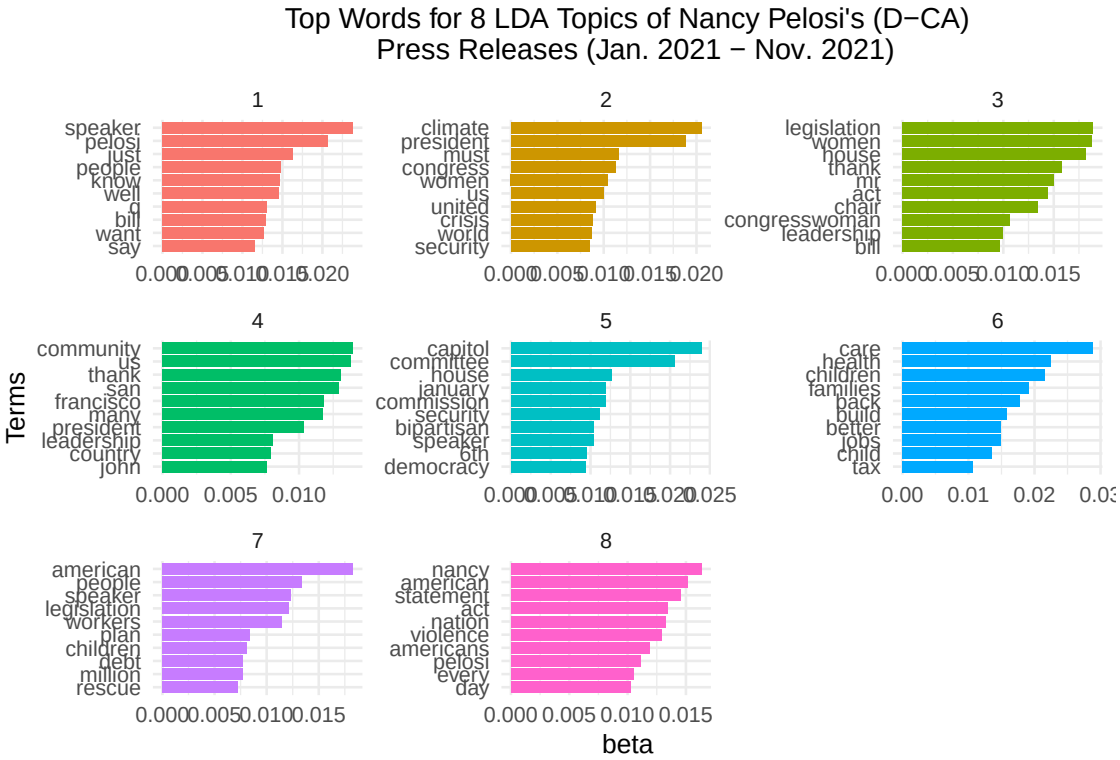
```
##     speaker      people      pelosi   president       thank        just
##        2015        1541        1520        1126        1034         939
##       house          us        bill         now         can        know
##         924         896         872         864         835         800
##        well legislation        said
##         782         759         740
```

The main model I use for the current project is called latent Dirichlet allocation (LDA). This is an unsupervised machine learning algorithm that takes the researcher's designated number of topics (assigned the letter "k") and probabilistically assigns documents to topics based on each document's words. I decide to look at 8 topics. However, it is important to note that LDA models do not "know" the correct number of

topics. Researchers need to examine the topics to see if they make sense given their knowledge and theory. Other approaches, like a semi-supervised seeded-LDA, take a dictionary as the base for each topic.

The plot below shows which words are most common for each of the eight topics. I give each topic the following name:

- Topic 1: Speaker
- Topic 2: Climate Issues
- Topic 3: Legislative Process
- Topic 4: San Francisco (Home Style)
- Topic 5: Capitol Committees
- Topic 6: Children and Health
- Topic 7: American Workers
- Topic 8: Nancy Pelosi

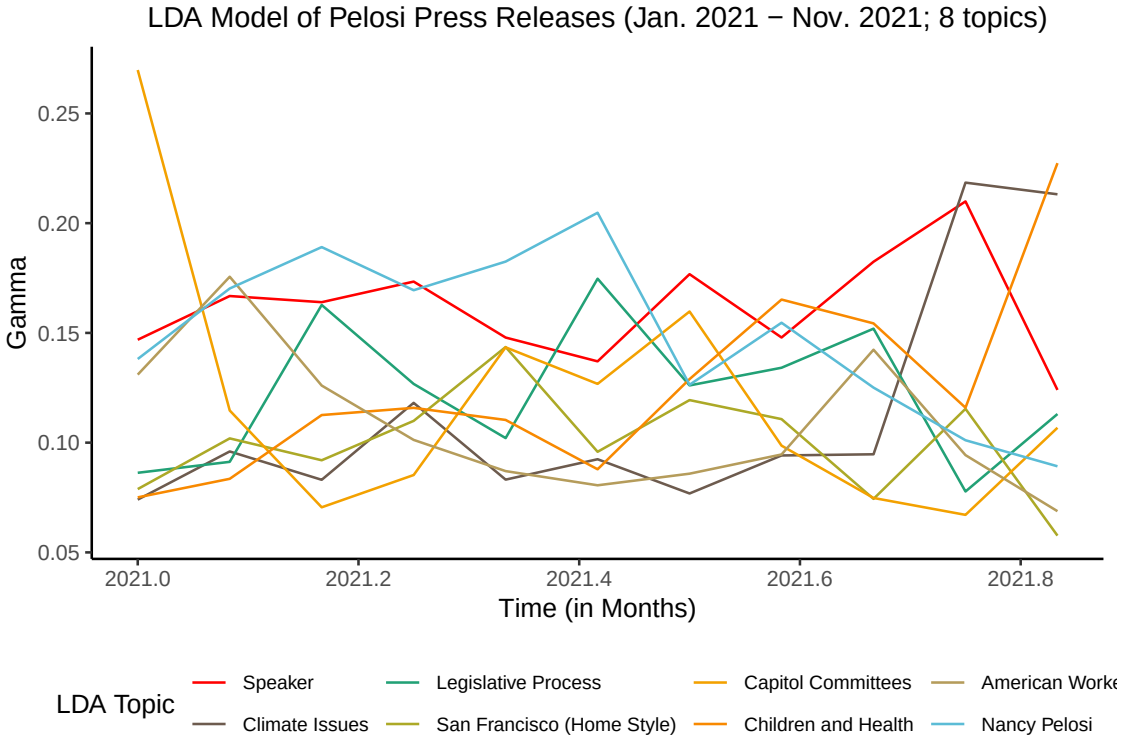


The next section presents the results for a longitudinal comparison in the frequency of each topic since the beginning of the 117th Congress in January 2021. Each of the eight topics is plotted across time. A number of intriguing results are found. For one, Speaker Pelosi's discussion of the January 6th insurrection is the most common topic in January (light orange). It tapers off across March when President Biden and

Congress are prioritizing his first months in office. However, it begins to reappear in early November – when former Trump administration officials disregard Congressional subpoenas.

The "Nancy Pelosi" (light blue) topic is common early on in the year when President Biden is striving to pass his landmark legislation and Speaker Pelosi uses her influence as Speaker – the "Speaker" (red) topic is also used similarly across 2021. The LDA model also picks up on the recent climate conferences with the "Climate Issues" topic being infrequent early in 2021 and spiking in October and November.

This plot provides a brief look into how the topics fluctuate across time. I now move to present similar analyses and figures for the Republican minority leader, Kevin McCarthy (R-CA).



The Minority Leader's Press Releases

I next scrape Kevin McCarthy's (R-CA) press releases to compare them to House Speaker Nancy Pelosi's political focus. I focus on approximately the same time frame (mid-January – mid-November). Interestingly, Kevin McCarthy has many fewer press releases. Over these ~11 months, his office has issued only 70 press releases compared to Speaker Pelosi's 350. The code to web scrape his press releases is largely identical to Nancy Pelosi's. However, the HTML tags are different (as is the homepage).

```
get_text <- function(pr_links){
  pr_page = read_html(pr_links)
  pr_text = pr_page %>% html_nodes("p") %>% html_text() %>%
    paste(collapse = "")
  return(pr_text)
}

# Create empty dataset to store information
mccarthy_pr <- data.frame(NULL)

for(i in seq(from = 0, to = 6, by = 1)){

# Provide the base url for all the press releases
base_url = paste0("https://kevinmccarthy.house.gov/media-center/press-releases?page=",{i})



# Read the base_url
page = read_html(base_url)

# Use "inspector gadget" tool in Chrome browser to find title, date, hyperlink, etc.
title = page %>% html_nodes(".views-field-title a") %>% html_text()
date = page %>% html_nodes(".views-field-created .field-content") %>% html_text()

pr_links = page %>% html_nodes(".views-field-title a") %>%
  html_attr("href") %>% paste0("https://kevinmccarthy.house.gov", .)
```

```
# Apply the function created above to all the links
statement1 = sapply(pr_links, FUN = get_text, USE.NAMES = FALSE)
#statement2 = sapply(pr_links, FUN = get_text2, USE.NAMES = FALSE)

# Create a data frame to store all of the press release information
mccarthy_pr = rbind(mccarthy_pr,
                    data.frame(title, date, pr_links, statement1,
                               stringsAsFactors = FALSE))

# Show pages
#print(paste("Page:",{i}))

}
```

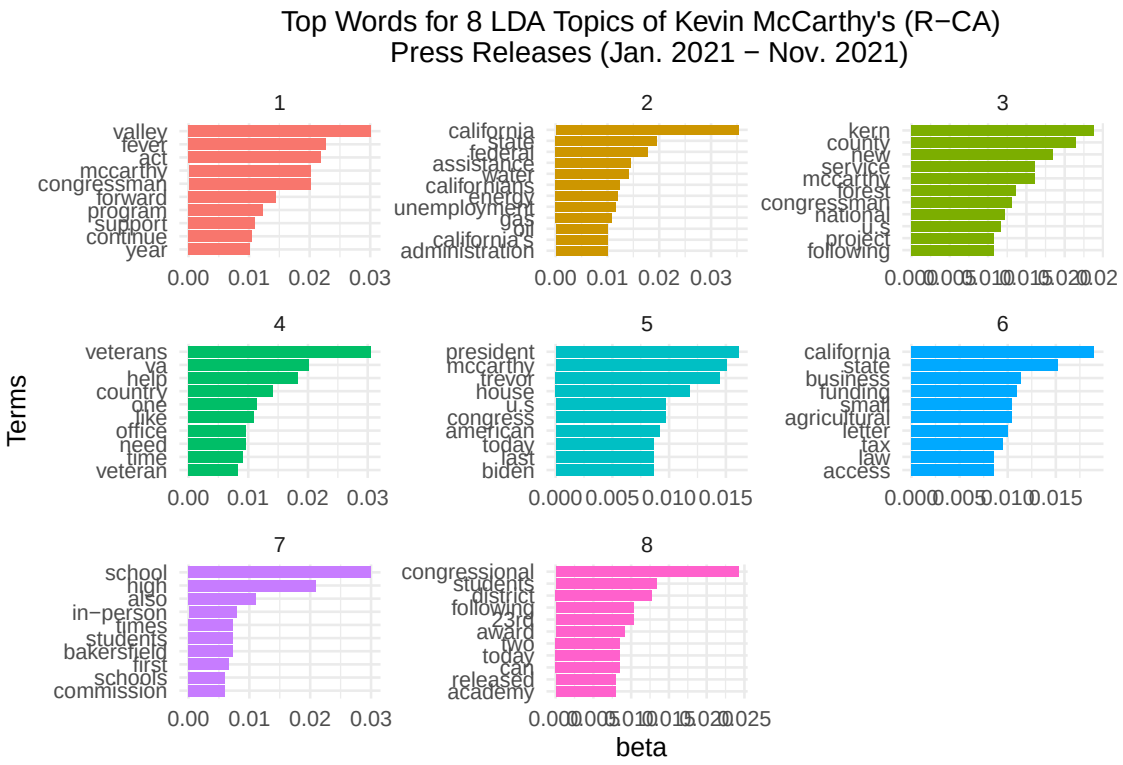
Overall, the Minority Leader's press releases require less cleaning compared to the Speaker. The main body of the press release appears to be consistent. The Minority Leader's press releases do, however, have many different phone and fax numbers within them, so I add code to clean them further. The top words across Kevin McCarthy's 70 press releases are shown below. The words relate to his district (e.g., California, valley, state, county) as well as his interests/focus in Congress (e.g., veterans, act, service).

```
##    california      mccarthy   congressman        valley         state
##           125           107            90            81            79
##      veterans          kern       service         kevin           act
##            69            61            56            56            54
##           u.s         fever        county congressional     following
##            53            52            51            51            50
```

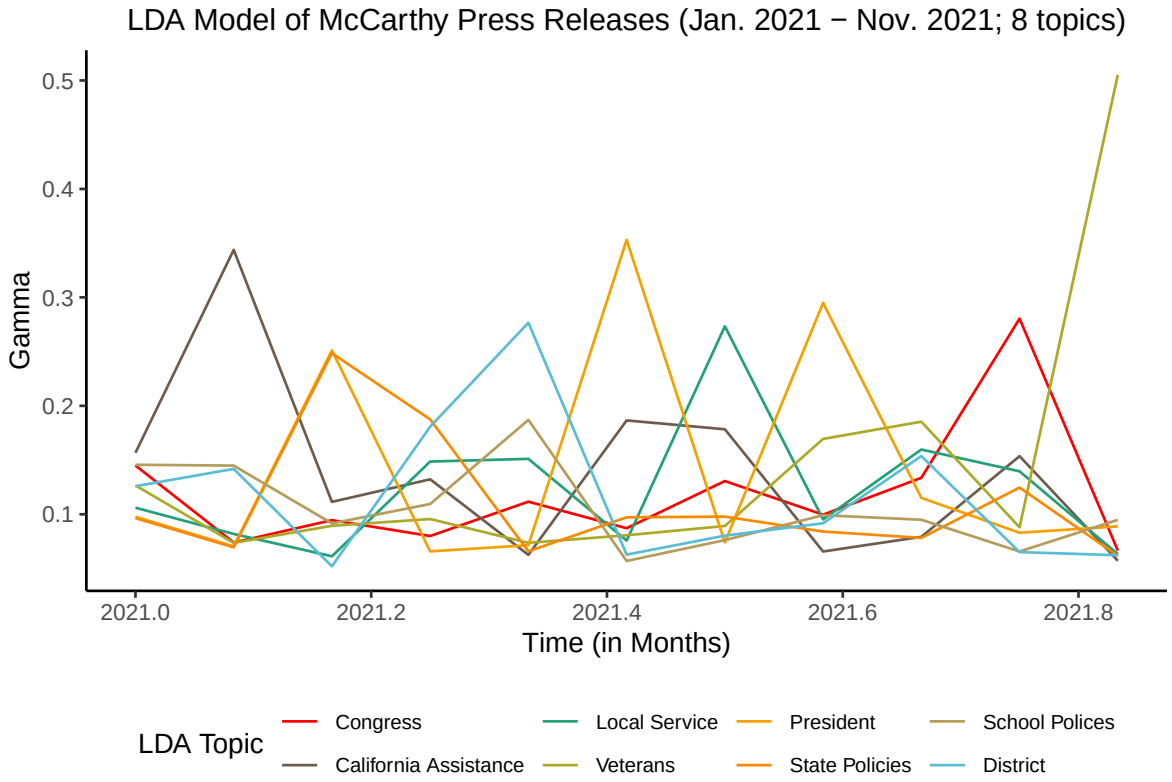
The plot for which words are most associated with each topic is shown below. These topics could change if I had more text or if I changed the number of predetermined topics. As such, there is a bit of subjectivity in determining the number of topics. However, I keep the same number of topics for both Speaker Pelosi and Kevin McCarthy. I give Minority Leader McCarthy's topics the following titles:

- Topic 1: Congress
- Topic 2: California Assistance
- Topic 3: Local Service

- Topic 4: Veterans
- Topic 5: President
- Topic 6: State Policies
- Topic 7: School Policies
- Topic 8: District



Finally, the plot below shows the topics for Kevin McCarthy over time. The uptick in veterans-related press releases occurs around Veterans Day. McCarthy's press releases also discuss President Biden regularly (lighter orange line). Compared to Speaker Pelosi, Kevin McCarthy appears more balanced in the topics covered over time. However, he has approximately 5x fewer press releases over the same time period. Perhaps his staff prioritizes certain aspects equally. Furthermore, his press releases feature more topics about his district and state than Speaker Pelosi who must split her focus not only on her district but the official duties of the Speaker of the House, too. Pelosi's press releases also appear to be more activist in nature in that they discuss popular events and topics more frequently than Kevin McCarthy's press releases.



Conclusion

The code above should provide a helpful overview of how to scrape Congressional press releases. This could be an automated process if I created a for loop and read in each member's HTML tags from the "selector gadget" Chrome tool. Imagine a spreadsheet with each member as a row and their website's tags for each piece of data as a column. Looping over that spreadsheet would allow me to collect many more members with much less code. There are many important questions that can be examined with Congressional (and White House) press releases. Overall, I hope you find this application helpful, and please e-mail me with any questions or feedback. Thanks!