Narrative

Ava Sharifi

12/10/2020

Brief substantive background / goal

Islamic Republic of Iran Broadcasting is the most popular state-sponsored news source in Iran. The goal of this project was to scrape all mentions of the United States in IRIB news articles from 1/1/2020 to 6/30/2020. This date range captures a number of important events in Iran-US relations including the killing of Qassem Soleimani, the downing of Ukraine Airlines flight 752, and the rise of Black Lives Matter protests in the United States.

I wanted to build a corpus of news articles from IRIB translated to English in order to provide the basis for my future thesis research. For this project, I intended to conduct basic sentiment analysis using dictionary methods and the translated text. However, in executing the project, I ran into a number of issues that prevented me from reaching the final goal of sentiment analysis.

Collecting data

Sources

I originally intended to scrape data from both PressTV and IRIB. PressTV is another state-sponsored news sources from Iran, but they only publish in French and English. I intended to compare the coverage of PressTV to IRIB. However, the search function on PressTV's website does not work correctly. The field that should limit the date range of results does not work and so any keyword search will return results from all time. Because of the issue with scraping PressTV, I decided to only scrape IRIB.

Volume of data

I ran into a number of small issues when scraping IRIB. First, the website itself is quite slow. Even opening a single article through Chrome takes a while. This made webscraping extremely time consuming. As a result, I decided to write code that would work to scrape the entire corpus, but I only scraped the first 50 articles in the search result. In the future, it would be helpful to build progress tracking tools into my map functions so that I know when I should quit an operation and when I should just let it run for hours.

Language and formatting

The URL for the search result on my computer had Farsi characters, which created further issues. To resolve that, I encoded the URL as UTF-8 within my webscraping function. Because webscraping was moving so slowly, I scraped individual article links in four separate chunks.

The articles within IRIB's website are sometimes formatted strangely. The CSS Selector Chrome plugin was not effective because it indicated that the CSS path for article text was different among different articles. To find a standard CSS path that could apply to every article, I used Chrome's inspect tool and found universal CSS paths.

There were two further issues with scraping articles. First, some articles did not have subtitles, which meant that my function could not be run through map_dfr. To solve this, I added a line of code which produces an "NA" coding when subtitle is not present, instead of just breaking the map_dfr. Second, the CSS path for section picks up two different sections, a main one and a secondary one. This also broke the map_dfr function so I added a line of code to collapse the two sections into one, separated by `\n\n`

Cleaning / pre-processing data

I attempted to clean the data before translating it. Some articles have videos embedded in them, so I created a function to remove the video player from article text. This may pose problems if there is article text after the video; however, I did not see that in any of the articles.

I also cleaned the data by separating the article section into primary and secondary section. Because the CSS selector is unable to select only the primary section, I used the CSS path to select both and later separated the two sections within my dataframe.

I was hoping to be able to clean the article dates as well. However, this was way more difficult than anticipated. There isn't a simple equation to convert between the two calendars, and others have attmepted to create extremely long formulas to do this in Java (https://stackoverflow.com/questions/54838700/how-to-convert-persian-date-to-gregorian-date).

Translation

My first attempt at translating the data used translateR, a package which hasn't been updated since 2014 and, for that reason, thinks that my scraped content is Esperanto.

The major issue with my project that I was not able to resolve was exporting it to a csv that looks and works correctly. The current final output csv looks very messy. I assumed this had to do with the encoding, so I made sure to encode the whole dataframe and csv output as UTF8 and it still did not work. I also tried a few different data cleaning methods such as removing `\t` and `\n` from the text data but that also did not impact the csv output. The only other solution I could think of was to turn the entire dataframe into unicode, which would render it illegible for most readers. This is a problem that I will need to solve moving forward with the project.

GoogleLanguageR's package is extremely useful for translating data, but it has one quirk that I have not been able to figure out. No matter which summarise, mutate, or select functions I use, the translated dataframe will have certain columns and column names. I even tried subsetting each individual column of the dataframe and then re-merging them to contain only the columns I want, but googleLanguageR still insists on keeping 2 extra columns for each column I want. Documentation on the package isn't super helpful and I haven't found any other ways to fix this issue.

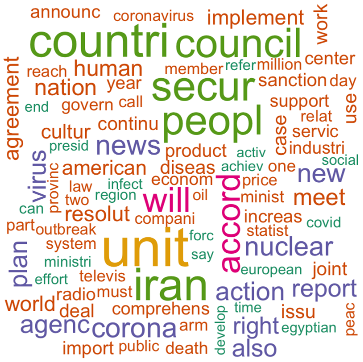
Analysis and visualization

Because of issues with scraping the volume of data I wanted and the messy translation process, I wasn't able to create visuals based on my data. I attempted to create a word cloud based on the English translation of the text and I also attempted a sentiment analysis, neither of which functioned as planned.

When trying to create a corpus for my word cloud, even though I pulled the English translation of the text specifically, the word cloud would use Persian words. I tried doing this many ways and did not have time to debug. As a result, the list of frequency of words in the text included both Farsi and English words. This is particularly problematic because I cannot remove Farsi stop words, which means that the most frequent Farsi word in the document is "een" which means "this" in Farsi.

Because the csv looks so wonky when it is exported, it is impossible to import and use to create a word cloud or sentiment analysis. I was able to create a word cloud by using the translated dataframe already existing in my environment, but it is not possible to create a word cloud using the tr_sample.csv.

This is the word cloud I was able to create:



Future work

Going forward, I hope to be able to find a faster way to scrape and parse the data. I also need to build familiarity with GoogleLanguageR so that I can change column names and create functional csvs. Greater familiarity with googleLanguageR would also allow me to run sentiment analysis and create word clouds. I'm not certain what the current issue with creating the word cloud is, but if I can make a cleaner dataframe with only the columns I want, I will be better able to navigate bugs.