the_watchdogs

uchicago CAPP30122 Project - Winter 2023

Group Members

Tiwaa Bruks (*tiwaabruks*), Rohit Kandala (*rohitk*), Lindsey Kilpatrick *(lkilpatrick*)

Abstract

Critical to a functioning democracy, the job of the free press is to force the government to be accountable to whom it governs; a role commonly referred to as, watchdogs. However, as the modes of media consumption evolve, and American citizens become as polarized as ever, trust in national news media is declining. Coverage of the January 6th insurrection at the capitol and the events to follow made this issue glaring. There is even disagreement among the use of the word “insurrection” itself. Through the process of data scraping, we will gather articles that discuss the attack at the Capitol, the January 6th House Committee, and the trials of rioters, from two of the most visited national news websites: CNN, and FOX News. We will then use token and sentiment analyses to inspect the language used to describe this polarizing topic, and compare it across media sources, and over time. Finally, a data visualization component will be implemented allowing users to further examine our data, through the option of isolating variables, times, and topics.

Software Structure:

The software has three main components: data collection, data analysis, and data visualization. Data collection and data analysis were both performed and had datasets saved down for a final data set. These two parts can also be run in the interpreter individually to see how everything ties together, but is not necessary for the visualization module to run.

The data collection module can be accessed in the interpreter by running `$ python -m the_watchdogs.scrape_sources` . However, this takes about 5 minutes to run so we have saved down these files (json) in `the_watchdogs/data` directory.

The data analysis module can be completed upon running `$ python3 the_watchdogs/preprocess.py the_watchdogs/data/fox_articles.json` and `$ python3 the_watchdogs/preprocess.py the_watchdogs/data/cnn_articles.json` in the interpreter. These files have also been saved down in `the_watchdogs/data` directory.

Finally, to see the data collection and data analysis come together, our data visualization can be accessed through the command: `$ python3 -m the_watchdogs.data_viz.plot` . After running this command, a port (7991) will open on the Flask app and upon opening in your browser, you can see our graphs.

Code Responsibilities

<u>Data Collection: Lindsey</u>

Articles discussing the January 6th insurrection from FOX News and CNN were gathered through web scraping and the use of an API.

FOX: Search results were able to be filtered by date on the site and results could be filtered to article only, however only 100 results would show up per search query. Given this, a series of search queries were generated using various combinations of keywords (‘capitol’, ‘riot’, ‘Trump’, ‘insurrection’, ‘January 6’, and ‘january 6’) to gather as many articles as possible. FOX’s api (api.foxnews.com) that is called upon a search query provided urls, and some other information, but a majority of desired information had to be accessed through web scraping.

When the module is called in the interpreter `$ python -m the_watchdogs.fox.scrape_fox`

1. several api.foxnews.com urls are generated
2. article urls from those search result pages are gathered
3. articles from the respective urls are scraped
4. that data is put into a dictionary, and those dictionaries are written to a json file (the_watchdogs/data/fox_articles.json)

CNN: Unlike FOX, one url was enough on CNN to get all results from the search. However, CNN’s search does not give us a lot of options, so there are several functions in the_watchdogs/cnn/cnn_utils.py to restrict date and source type. CNN’s api (search.cnn.api.com) that is called upon a search query provided all desired information besides the article description so most data could be accessed through the api, but the article description needed to be accessed through the html.

When the module is called in the interpreter `$ python -m the_watchdogs.cnn.scrape_cnn`

1. article urls from the search result page are gathered
2. articles from the respective urls are scraped
3. that data is put into a dictionary, and those dictionaries are written to a json file (the_watchdogs/data/cnn_articles.json)

Data Processing and Analysis: Tiwaa

To transform the scraped data into a usable format for visualization, the data was first loaded into Pandas dataframes. The date field was formatted correctly for time series analysis. Stopwords and punctuation were then removed from articles using the NLTK library. Then the article was broken down into tokens that could be used for word frequency analysis.

A sentiment analysis was also performed on the text prior to tokenization. The results of that analysis were broken down by score (-1 to 1) and category (very negative, slightly negative, neutral, slightly positive, and very positive).

Finally, the cleaned results for each source were output to a CSV for easier visualization.

Data Visualization: Rohit

The transformed data is converted into a Pandas dataframe, and is parsed through helper functions in data_viz_prep.py. There is an additional supplemental file called app.py that is necessary for the main file, “plot.py” that contains the data visualization functions as well as HTML elements for the page structure.

Running the command, `$ python -m the_watchdogs.data_viz.plot”` ,opens a port (7991) on the Flask app where the user can see the visualizations and interact with them.

There are three unique plots: two wordclouds that shows the most frequent words—with more frequent words appearing larger–, one line graph that shows the number of articles by news source, and the user can toggle between the year, and a bar graph that shows the number of articles by news source based on the sentiment category from the processing and analysis step.

Application Interaction Instructions:

This application can scrape, analyze, and visualize data. Further instructions about specific commands can be found in the README.md. It produces raw data that’s been scraped from CNN & FOX in the form of a json, analyzed data in the form of a csv, and a data visualization in the form of a Flask App. The output you will see is:

1. Two word clouds, one with CNN data, and one with FOX data.
2. A line graph showing the number of articles by source, and you can toggle the year.
3. A bar graph showing the sentiments (5 categories) by news source.

Conclusion

Our hypothesis before starting this process was that FOX would be far less likely than CNN to use words to describe the events on January 6th as an ‘insurrection’ rather they would frame it more as a peaceful protest. In terms of sentiment analysis, we expected both networks to have an angry sentiment when discussing January 6th, but thought potentially FOX would be more angry in other news coverage of the event itself, as well as with the hearings from the January 6th Committee and ongoing legalities with insurrectionists.

From a search result alone it was evident that CNN had far more articles on the events of January 6th than FOX. Notably, a search for ‘capitol riot’ on FOX’s site was more successful in returning a number of articles on FOX than ‘insurrection’. There were various hindrances to scraping all search results like special types of articles, poor relevancy generators on the news website, and additional issues as outlined in the data collection section. In turn, our final data set that we felt confident in had 338 articles from CNN and 327 from FOX.

We hypothesized that CNN would be much more likely to use words like ‘insurrection’ in their coverage than FOX, and that FOX’s language would be more neutral, framing the events as a peaceful protest however this wasn’t overwhelmingly the case. The most frequently occuring words as displayed in the wordclouds, both shared very similar keywords like “Trump”, “president”, “election”, and “house”. There were some words that were more prominent than others. For CNN, “republicans” was bigger in their word cloud while “democrats” was more prominent in the FOX word cloud. This also followed for “police”, which was way less prominent in the CNN word cloud. Something notable as well is the word “news” was relatively big on FOX’s word cloud, indicating that they were talking about other news media’s coverage of the event, which we predicted would contribute to a negative sentiment for the network.

For sentiment analysis, we expected the coverage of the events on January 6th to be polarizing like the event itself. Generally, we expected both networks to show more negativity than an average news topic. We found that CNN had significantly more articles than FOX in the “very positive” category; 129 to FOX’s 65. On all other categories, notably “very negative”, FOX was the leader. It is important to note that the sentiment analysis was derived from the “Vader” python package, and is a rough generalization of the tone and attitude of the article. Moreover, packages like Vader often lack social and political context and analyze sentiment in a vacuum. Both CNN and FOX could have a negative sentiment in their article discussing January 6th, but for two very different reasons.

An interesting trend we found in our line graph was that, while FOX & CNN had a similar number of articles pertaining to January 6th, FOX’s coverage was a lot more sporadic than CNN’s. For almost all months, CNN had more coverage than FOX, but when FOX had more coverage (Jan 2021), it was substantially more. An important note is that the January 6th House Committee hearings started June 2022, and progressed till December 2022. This explains the noticeable increase in coverage when toggling the “2022” year, and navigating to June. This coverage declined until roughly mid-September when both FOX and CNN increased their coverage.

Overall, we feel that CNN did have more consistent coverage of the events of January 6th, and that keywords were similar between the two networks with a heavy emphasis on Trump. It is important to note that there is context around these keywords in the wordcloud that is missed, however seeing their similarities is still a promising sign. We saw that CNN had much more positive coverage than FOX, and FOX slightly more negative than CNN. In turn, the sentiment analysis told a polarizing story that we expected to see for an event that two news media companies see from very different lenses.