




Activity: CS 5010 Homework 3: Python and Web Scraper

Headline Scraping from Major News Sites and Word Frequency Analysis

Name: H. Diana McSpadden

UID: hdm5s

September 19, 2020



Overview

I performed web scraping of headlines from six news sites over four days using a python script. I collected headlines, URLs, datetime of the scraping, and website. Using AllSides Media Bias Chart (AllSides, n.d.), I categorized each headline as sourced from a center, lightblue, or lightred news site. After collecting 1,616 headlines, I used a Jupyter Notebook to clean, process, and perform word frequency analysis of these data.

Approach

I had interest in scraping multiple sites because I wanted to experience the challenge of scraping varied html structures. I focused on scraping words, not numbers, because I wanted to work with a natural language processing library. For the assignment, I scraped headlines from major U.S. news sources. I used headlines because headline text is simple text using few “filler” or “stop” words, and theoretically, there should be alignment of news topics between major news sites because news is topical. To implement web scraping, I used the requests library (Kumar et al., 2020). To parse html structured content, I used the BeautifulSoup 4 (Real Python, 2020) library.

I also had interest in filtering the data set collected via web scraping. For the assignment I included filtering by categorizing each news sites as center, left-leaning, or right-leaning. I used AllSides.com’s Media Bias Chart (AllSides, n.d.), Figure 1, to determine the category for each of the six news sites. The news sites and their categories are listed in Table 1.

News Site	URL Used For Scraping	AllSides.com Category
ABC News	https://abcnews.go.com/	lightblue
AP News	https://apnews.com/	center
CBS News	https://www.cbsnews.com/	lightblue
Fox News	https://www.foxnews.com/	lightred
NPR News	https://www.npr.org/sections/news/	center
Reason	https://reason.com/	lightred

Table 1: Scraped News Sites with AllSides.com Category

I wanted to use language processing. I added this dimension to the assignment by utilizing the Natural Language Toolkit library, NLTK (*Tokenizing Words and Sentences with NLTK*, n.d.), to remove stop words, and analyze word frequency in the complete data set, and in the data set filtered by center, left-leaning, or right-leaning news websites.



Figure 1: AllSides Media Bias Chart with Selected Sites Circled

Utility

With additional data, over longer time periods, and with a consistent number of records for each of the categories, I believe my web scraping and headline word analysis can identify where left-leaning, center, and right-leaning news coverage diverges, or converges on topics. Identifying words frequently used by each category of news sites, and not by other news site categories can identify blind spots in coverage, or topics that are over-covered.

I believe that it would be interesting to investigate if, with enough training, a categorization machine learning algorithm could use scraped headlines from a particular time period to determine if a news website should be categorized as center, left-leaning, or right-leaning.

Wish List

With more time I would have added several more news sites to my data set. Each news site has an entirely different html structure, and I found it difficult to reuse my code between websites. Because I was not able to add more websites to my data set, the data set contained many more “lightblue” records than other categories. Table 2 shows the number of headlines by category.

Category	Number of Headlines
center	134
lightred	195
lightblue	339

Table 2: Number of Headlines by Category

With more time I would investigate the type of words that were in a similar ratio of headlines for each of the site categories, and words that were over or underrepresented in each of the site categories. I also would have included news sites in the additional categories shown in the AllSides Media Bias Chart, i.e. “darkblue” and “darkred”.

Extra Credit

My use of the NLTK library to create word frequency plots added an extra dimension to this project. I enjoyed trying to determine how to massage my data frame of headlines to remove stop words and create valid data for the NLTK library to process. I repeatedly needed to use `map()`, `split()`, and `join()` to format my data, and I unexpectedly found a use for `reduce()`.

References

AllSides. (n.d.). *AllSides Media Bias Chart* [Illustration]. AllSides.

<https://www.allsides.com/sites/default/files/AllSidesMediaBiasChart-Version2.jpg>

Beautiful Soup - Navigating by Tags - Tutorialspoint. (n.d.). Tutorialspoint. Retrieved September 14, 2020, from

https://www.tutorialspoint.com/beautiful_soup/beautiful_soup_navigating_by_tags.htm

Kumar, N., the_galaxy_hunter, & Hasan, S. (2020, August 20). *Implementing Web Scraping in Python with BeautifulSoup*. GeeksforGeeks. <https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/?ref=lbp>

Real Python. (2020, August 21). *Beautiful Soup: Build a Web Scraper With Python*.

<https://realpython.com/beautiful-soup-web-scraper-python/#part-3-parse-html-code-with-beautiful-soup>

Tokenizing Words and Sentences with NLTK. (n.d.). PythonProgramming. Retrieved September 15, 2020, from <https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/>