News-Scraper ETL Project

Kyle Cielencki, Luis Cerrilla and Will Pumphey

The purpose of this project is to ***Extract*** information from three main news websites, ***Transform*** the scraped data into a structured data frame (with categorizations), and ***Load*** this data frame into a SQL data base, to be able to retrieve the latest news from our tree main websites on one single table.

Extract:

- We primarily used Jupyter Notebook's, Splinter and Beautiful Soup to extract the latest news headlines, URLs and dates from three sources. For this project, we pulled news from CNN, NPR and CNBC. We choose these websites as they are more reliable news sources, have neaty laid-out websites for ease of data scraping and are constantly being updated.
- Additionally, we chose these websites as their pages are laid out in a similar fashion (although NPR differed slightly). For the sake of simplicity, we chose to scrape the article headers, dates and URLs for each news story. This limited the amount of null information in our database.
    - When we initially set out on this project, we had hoped to pull a subheader/teaser for each story; however, not all sources provided this information (with the exception of NPR).

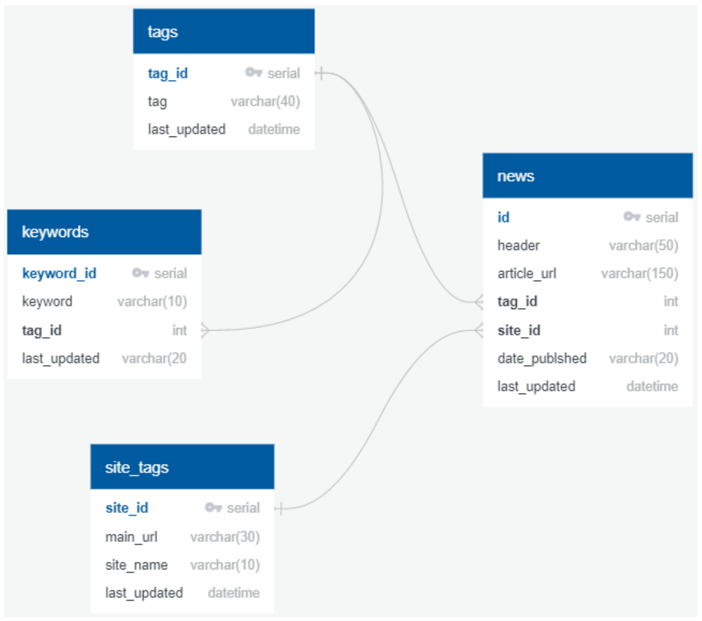
Transform:

- We input our scraped data into uniform data frames to ensure that this data could be loaded into our SQL database properly. As mentioned above, this meant removing columns we had initially hoped to have in our dataset. However, there were issues with site layouts and information provided by each data source that limited our ability to do so. Furthermore, we needed to ensure that all the data frames had the same headers as the database so that all scraped data could be accurately pushed into the SQL database.
    - This required sanity checks querying the data base to make sure it was correctly pushed.
- Additionally, we had to apply appropriate tags to each article. This required creating a formula to search each headline for keywords that related to each tag *(see below)*:
    -

Keyword	Tag	Tag ID
covid	COVID-19	1
coronavirus	COVID-19	1
trump	Politics	2
biden	Politics	2
election	Politics	2

Load:

- We utilized the Google Cloud to house our SQL database. Our database consists of four different tables, connected by ID keys.
    - Our "News" table is the only dynamic table, meaning the information will constantly be updating as it contains all the articles headers, URL's, date published, and tag/site ID's.
    - Our "Site Tag" table is static and is a reference point to each news site used (contains the Site ID, Main URL and Site Name.
    - Our "Tag" table is static and contains the Tag ID and Tag.
    - The "Keyword" table is a static table housing all the keywords to categorize the articles in reference to the appropriate tag. All tables have a "Last Updated" column indicating the last time a row was updated:

▪ 

Future work and limitations:

- For further work, we would like to expose this database on a table in a front-end website. This would allow the end-user to utilize our database as a news aggregator. Furthermore, it would be beneficial to transform our "ID" columns (tag and site ID) into actual language to be of use to the reader. This would provide more information if it read "political" instead of a 1.
- **Limitations**:
  - The largest limitation we encountered was timing. The timeframe for completing the project limited our ability to transform and present the data as we would have hoped.
  - A lot of our initial work surrounded getting the articles URLs, as this proved to be the most difficult item to scrape from each website. Each URL was located deep within certain classes and required for loops.
  - Another limitation encountered was the number of keywords that would be assigned to each news category as we had to limit it into a few words (otherwise, this would have been a very long for loop).
    - This would be an area we could improve upon given additional time.
  - An additional limitation we encountered was the differences in website formats. Each of the three websites had different styles, labels, bins, and layouts. Some had information that was more hidden than others, while some had constraints on what information we could use in general.
  -
    - For example, one of our sources did not provide a date time in the same format as the other two. Because of this issue, we had to change the table date column from a date type into a "varchar" type.
    - Furthermore, not all websites provided a subheader or "teaser" for each news story.
      - In the case of NPR, we encountered issues pulling these "teasers" as there was a mismatch between classes (21 "teasers" vs 20 stories). This was due to them utilizing this class for parts of the website that did not relate to a story.
      - We first used Beatuful Soup on CNN and then we realize that it was a dynamic website, so we had to change the code to Splinter.