CSC110 Project: Analysis of Public Health Discussion in The News over Time

Brian Wang, Yahya Elgabra, Kareem Salem, Wenqi Zhan

Friday, November 5, 2021

Problem Description and Research Question

Research question: To what degree is public health being discussed in the news over time as a result of the pandemic? What impact does COVID-19 have on the media coverage regarding public health globally? And in other words, how did the outbreak of the virus shape the authors' attention on the public health sector?

During the peak of the COVID-19 pandemic, public health was at the top of discussion in journalism since many people were extremely concerned with the latest lockdown measures, safety recommendations by health experts, and the global situation as a whole. Since the representation of public health in the media closely reflects the prevalence of discussion of coronavirus in the public consciousness, it is critical to examine how much influence COVID-19 has left on the media as a whole. Specifically, we are questioning if the topic of public health has become more or less frequent in the Guardians news with respect to each categories, namely news, culture, science and technology, politics, business, and other, over time.

Our group has created this research question because we believe there could be interesting findings that can help guide our perception about the significance of public health. Are news outlets starting to pay more or less attention to public health? Do they pay more attention because of the ever changing regulations from the government, or less attention because more and more people are starting to get used to the status quo?

It is just as crucial to investigate the overall impact of COVID-19 on the news as a whole, not just news about public health. For example, a news article about a soccer player being unable to participate due to COVID-19 is not about public health, but still shows the influence of COVID-19. The media tends to report about the most attention grabbing events, and by scouring news articles from Guardians and categorizing six major sections of the news website, we can create a picture of which sectors are affected more by COVID-19, and which are less affected or even not affected at all.

To conclude, our group will be collecting, categorizing, and assessing tens of thousands of news articles about public health and generating an average influence index for each category, in order to compare how closely, at different stages of COVID-19, different sections of media coverage and the virus are related.

Dataset Description

Our primary dataset for this research question is the Guardian Open API. This API is sourced from the Guardian Open Platform and provides basic information about the news articles in JSON format, such as the headline, section, body text, and publication date. The API provides every single article the Guardian has written, but we will only use articles from the last 2 years, which is 150000 articles worth of data. We only use data on the publication date, section, news headline, word count, and body text.

Since the entire raw data is too large to send to Markus, approximately 1 GB, we will pre-process this data. Most of the file size comes from the body text and the headline, so we process this into an influence score using a custom algorithm which will be explained in the computational plan.

Computational Overview

To obtain the data we need, we will extract information from the news APIs to which we have access. This is done with the function pull_from_api in api_pull.py. In this function, we will send multiple GET requests to the API with parameters created by get_params to filter by date, show the desired fields, and provide an api key. Here, we use a new library called requests. This library can send a GET request to the API in order to get the data in JSON form,

which is parsed into a python dictionary using requests.json(). The data gathered by the API will then be saved into a raw_data folder, where each file is named after a date and contains only articles published on that day.

Then, we use a custom algorithm to calculate an influence score for each article by looking through each word in the headline and body then weighting it appropriately. We also soft cap the scores using the log function so that very relevant articles do not overshadow other articles and hide the true overall trends. The word count of the articles is also accounted for so as to not advantage longer articles.

Then, we process the data in the raw_data folder through this custom algorithm, and save the resulting influence scores into files in a processed_data folder, where each file is a csv file named after a date and contains only articles published on that day. This is done with the function process_raw_data in the file preprocess_raw_data.py. The columns of the resulting csv file are the time of publication, section id, section name, and influence score.

All of the above was done to preprocess the data for later use, an API takes much longer to access than a local file.

When running the program, we first load the articles using the load_articles function, and save this in a dictionary mapping dates to a list of articles on that day. These articles are stored in an article class, which neatly stores information on Articles (i.e, the date which the article has been published, the section id (which is how The Guardian API refers to this section), the section name, and the influence score that was calculated during the pre-processing stage)

In terms of reporting the data, our group created a user-friendly GUI that allows the user to view the overall statistics on COVID-19 relevance rates in the news, and an interactive graph that displays different graphs based on genre and time specifics. The GUI created using Pygame which essentially allows for a presentation screen to be presented and the specific representations like text, buttons, drop downs, and graphs to be displayed on it, each represented using classes. We start with the *main_loop* function which is called and opens the window. The system runs using a while loop based stack implementation, where the user starts at the first while loop, then progresses to each inner loop (push) as they move into different screens. Essentially, each while loop iterates constantly, gets the pygame events and displays the necessary visuals, and if a button it will set the next inner loop to true and the screen will be cleared and display new visuals. If the user wants to go back from interactive graph or over report they can hit the back button at the top left which makes that current loop false and thus moves back a stack (pop). For the overall report, the program calls *get_overall_report_data* and retrieves the necessary statistics, displaying these into the text objects to be displayed in the loop. In the interactive graph, the program calls *send_change* initially to set an initial graph, then calls *send_change* to update the graph whenever the graph button is pressed, getting current genre and time selected and passing these to the function to retrieve and display the proper graph through the function *plot_data*.

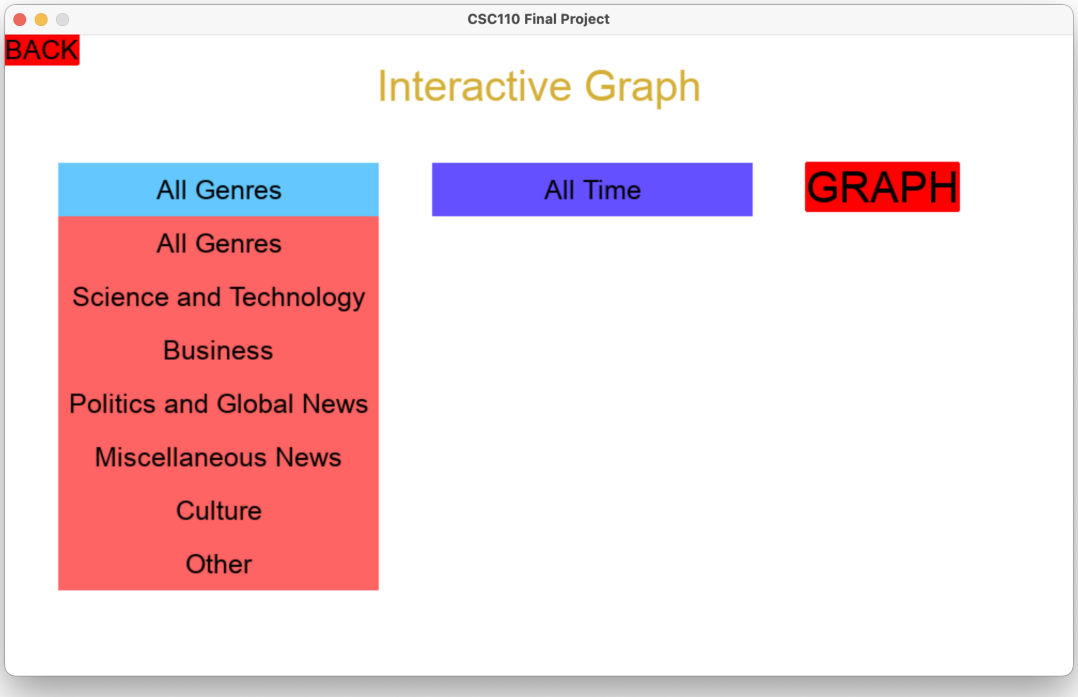
The *plot_data* function takes in the genre and the dictionary mapping time to the influence score as two parameters, and generates a graph showing the growth of Covid influence score of the selected genre over the selected time interval.

Obtaining data sets and running your program

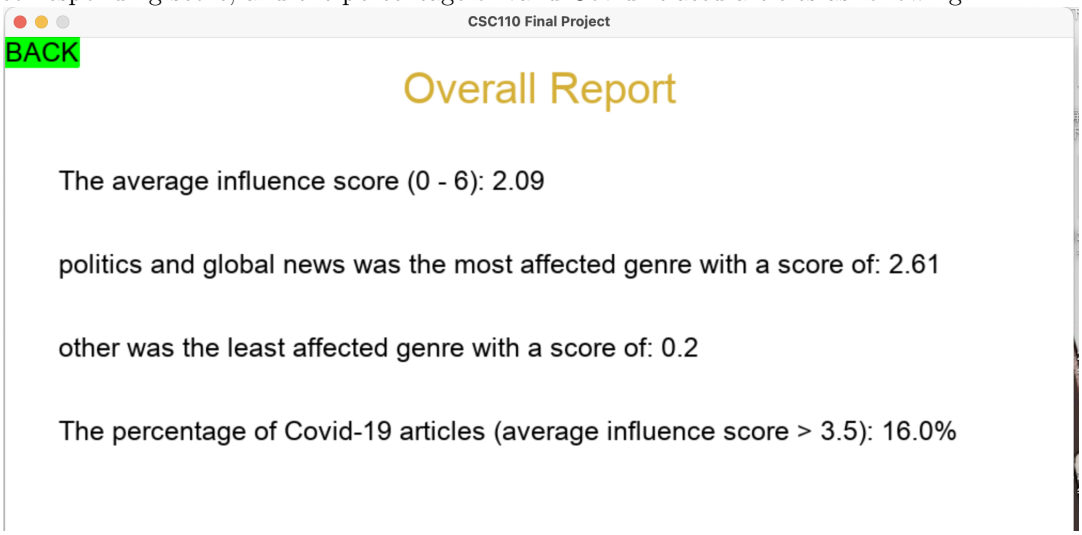
In order to pull the raw data from the API, you can uncomment the commented lines of code in main.py and run it, but be warned that it may take a really long time to download the entire dataset because of rate limiting.

To download processed data or previously gathered raw data, use the Claim ID: s2SnyFUNBq8hGwXT and Claim Passcode: jet2CB97KaNqe8op on send.utoronto.ca The processed data is a zip file that contains a folder called processed_data, which needs to be in the folder that contains main.py. The raw data is also a zip file that contains a folder called raw_data, which needs to be in the the folder that contains main.py. However, this folder does not need to be present to run the program.

After running main.py, an interactive application would immediately pop out on the centre of the screen, displaying our research topic as well as all the information regarding our group. At the bottom of the screen, you will expect to see a menu button directing you to a new page with the options of Overall Report and Interactive Graph.

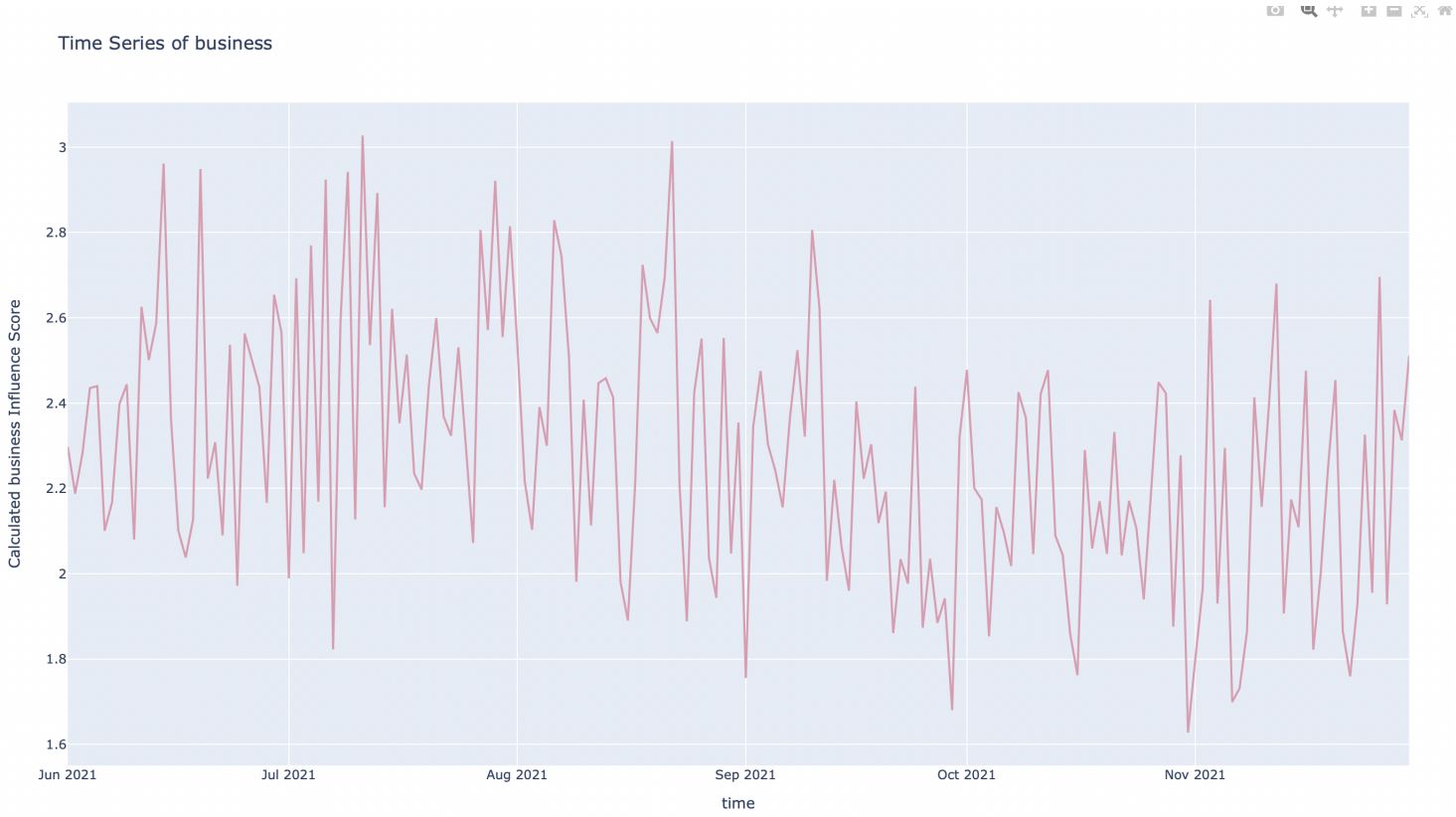


For Overall Report, we can easily access the average influence score, the most affected genre, and the least affected genre with corresponding score, and the percentage of valid Covid related articles as following:



For Interactive Graph Button, you will be directed to a new page with genre and time interval selection. After selecting the target genre and its corresponding time, our program would produce a graph in the browser that allows zoom in and zoom out options.

Here's an example, if you select business and 6 months, there would be a graph illustrating the growth of influence score of COVID-19 over the last 6 months.



Project Plan Changes

Based on the feedback we received from the proposal, internal conversations, and the conflicts in our implementation we made several correcting changes. Firstly, in our consideration of of how variations of data from different sources would affect our data extraction we decided to only collect api from the Guardians website to streamline the process. Since the Guardians news has over 40 sections, in implementation we generalize those into 6 major categories worth reporting. Next, while trying to explicitly split the request into multiple api calls, we imported requests library since it is more efficient than urllib. After that, for the computation model, we specify the header and the body of news articles to be different weights, and customized our own key words dictionary to map the frequently-occurred keywords to the corresponding index value. Also, due to some the articles containing too many key words that contribute to the final influence score of the genre, we adopted the logarithmic function to put a constraint, or a soft cap on the score. We do not want to see one highly Covid-relevant article to overpower the ones with lower relevance. Lastly, for the graphics and data visualization stage, we implemented an interactive graphic user interface to ask user for the specific time period and genres they would like to study, and the program returns a graph displaying the average influence score of each genre as well as its growth over time. In addition, we included a show all genres graph and all time graph, in which so that user can compare the virus's influence on each genre.

Discussion Section

As we can see from the results, public health discussion in mass-media has increased significantly around the time the pandemic started, then decreased and stayed relatively constant (at a level higher than pre-pandemic). The reason we are seeing the new constant rise higher than pre-pandemic records which may be because COVID and the pandemic is still posing significant public health risks or because people have simply become more aware about public health, the specific cause remains to be seen as we do not have any articles from the post-pandemic era yet. However, our current project only shows us the level of Covid-19 prevalence in the news but does not provide understanding of the content, so it's limited by not telling us what the news is reporting about the pandemic. So the results of our computational exploration helps answer our question of the frequency of Covid-19 in media and how this has changed over time but doesn't answer what news outlets are writing about which can be a future improvement.

While The Guardian provides us with a large data set a diverse results, we are still limited to just one news outlet source which may have made our results more biased based on their style of reporting (if The Guardian was more or less reactionary of the pandemic). In addition, limiting the source to just The Guardian limited our range of topics with most articles we obtained being on politics and global news. This created a disproportion in the data we collected, and meant that we had a much larger basis in politics than other categories than others like sports, culture, etc. The custom algorithm to find the influence score could also be inaccurate since the list of weightings of each word and what words to include can't be easily created without more tools and data.

For further exploration of our project for the future, we can work on improving our algorithm quality and versatility, as well as continue to expand our algorithm and add new representations to the user to provide an extended understanding of the pandemic and how it has affected the news. As discussed earlier, our algorithm only works with parsing data from The Guardian, so perhaps for future development we can include build new algorithms that extract data from other sources or even make a custom algorithm that automatically identifies the key data in different formats to work on any source. Another improvement would be fine-tuning our algorithm so we can make our results even more accurate. This would mean reworking our influence score to be more precise and apply to different news outlet reporting styles by including these as parameters in the calculation. Lastly, we can build off from this foundation and make new representations of our data that could be doing analysis on the type of influence (positive/negative), filtering by geographical locations of media influence, and expanding to more categories and time frames to provide a more in depth representation of how Covid-19 is affecting the media. In total, from here we have a solid basis to improve our algorithmic quality and add further complexity and applications to become even more informative.

In conclusion, this project was especially beneficial in applying our skills by challenging us to take our current understandings of python and expand on them by learning how to take extract raw data from sources, synthesize it using algorithms, and then represent it on a GUI using visuals and graphs to create a real world application.

References

The Guardian Open Platform. API Documentation, version 1.
Available at https://open-platform.theguardian.com/documentation/

Stack Overflow. Trying creating dropdown menu pygame, but got stuck.
Retrieved December 12, 2021, from https://stackoverflow.com/questions/59236523/trying-creating-dropdown-menu-pygame-but-got-stuck.

Quickstart¶. Quickstart - Requests 2.26.0 documentation. (n.d.). Retrieved December 12, 2021, from https://docs.python-requests.org/en/latest/user/quickstart/.

Khan, M. W. (2021, February 10). Create directory in python. Delft Stack. Retrieved December 12, 2021, from https://www.delftstack.com/howto/python/python-create-directory/.