Final Project Report: Pessimism in Digital Media: Before and During the Pandemic

Friday, December 13th, 2021

Problem Description and Research Question

People are influenced by the media. Whether it be news or social media, every bit of media we read garners emotional responses from us, both positive and negative. For example, finding out about our favourite sports team winning or a musician we like coming to town are things that make us feel good. However, the majority of the media we read tends to be negative, and what's worse is that this negative news is addictive, leading to stress and - in the worst of cases - depression.

This is the point made in "Coronavirus: How much news is too much?", a BBC article written by Maddie Savage. According to her article, 7 in 10 Americans said that they had to take breaks from COVID-19-related news, as it was causing "corona fatigue", a term dubbed in the article for fatigue from COVID-19-related news ("Coronavirus: How much news is too much?"). Of course, this significantly impacts people with a history of mental health and anxiety issues, and addiction to the media can lead them to dangerous sources of stimulation to alleviate their stress ("Coronavirus: How much news is too much?"). Unsurprisingly, it seems those being hit the hardest are media industry workers, as the seemingly never-ending news about the current pandemic has required them to stay on top of data and trends in a way they have never experienced before.

Considering all of this, we became interested in the topic, leading us to investigate the change in people's pessimistic views on digital media before and during the pandemic. We began to ask ourselves: **how did the pandemic affect the pessimism we see in digital media, compared to the pessimism in digital media before the pandemic?** Naturally, there was only one way to find out, and that was to program it ourselves.

Dataset Description

We compiled our own dataset for this project, based on news articles from different sources before and during the pandemic. First, going through each article's HTML, we took all the text from each article, and stored them inside a text file; this process would vary for websites with different types of articles. For example, with CNN, for live stories, the text is inside paragraph tags, so we accessed text from every p-tag element, and for regular stories, the text is inside divs with specific classes (zn-body_paragraph), so we accessed text from every div element with that specific class. Then, we stored each article's text (with its title, publisher and type of stories (live or regular)) inside a JSON file. Each JSON element contained the title, news source, and type of article, all of which were string value types.

Computational Overview

The most challenging part of answering the question is gathering the dataset, as the article websites, we use as our sources use web frameworks that fill the HTML with nested divs and abstract class names. The web pages that we gather data from also contain a lot of irrelevant and unrelated text and this will require a manual look through to ensure tidiness. We have used a simple implementation of a web-scraper that can parse XML which we also use to parse our lexicon. In order to make the code easier to understand and scale we use an abstract Source class that acts as a framework for sub classes that are specific to each news agency. The sub classes handle jobs like aggregating links and articles using the Scraper module that we built. This takes a lot of time given that each sub class makes between 50 to 200 requests so we use the Source class to serialize the files to the JSON file format so that they can be retrieved once.

Word Form	WordNetID	POS	Sense	Polarity
great	a-01123879	JJ	very good	1.0
terrible	a-01278818	JJ	exceptionally bad or displeasing	-1.0
thick	a-01386883	JJ	dense in consistency	0.1
elegant	a-01677433	JJ	refined, tasteful in appearance	0.5

Table 1: A table showing how our program ranks the polarity of sentences:

All the articles are stored in an article dataclass from which they can be mutated. To give them a 'happiness' or polarity score ranging from -100 to 100 we use a lexicon developed by Dr. Tom De Smedt, a linguist at University of Antwerp. It contains just under 3000 words each with a polarity score and so we use a bag of words approach to give each article a score. This score represents how positive or negative each word is and allows us to find the overall sentiment. The score is then averaged out by the number of words and multiplied by 100 in the article saved, where its polarity score ranges from +100 (for positive) to -100 (for negative).

This is repeated for every article for each year and plotted onto two histograms that are overlaid. This provides an excellent representation of the range and concentration of the values found in our analysis and allows us to easily visualize the difference between pre-pandemic and pandemic articles. These values are also used to calculate a mean and median score for each of the years for a quantitative comparison. These visualizations will be possible with the plotly library. All the sources are also plotted against each other for an interesting visual comparison.

We use our hypothesis module to compare parameters of each source. For example we may want to test the statistical significance of the average polarity of 2019 articles and 2020 articles. The function can take any source as well as any function(mean, median) that can act on the polarities of the 2019 and 2020 articles. We take the null hypothesis to be that the 2019 and 2020 articles have the same mean. We then take the existing polarity scores and randomly assign them to 2019 and 2020 articles and calculate the new mean. We then add this to our list of means and repeat the process 5000 times to give us a sampling distribution. We then count the number of extreme values and divide them by the number of samples to give us a p-value. It is standard to take 0.05 or lower as the p-value at which we reject the null hypothesis. This along with the sampling distribution is also plotted on a histogram.

Instructions for obtaining data sets and program

Attached along with `main.py` and the supporting python files is the file `en-sentiment.xml`. This file serves as a lexicon for natural language processing. This file must be in the same directory as the `main.py` file. If this is not possible, modify the `LEXICON_PATH` constant in `main.py` to the new path. To retrieve the dataset we use the standard `urllib` python package to make requests to the following sites:

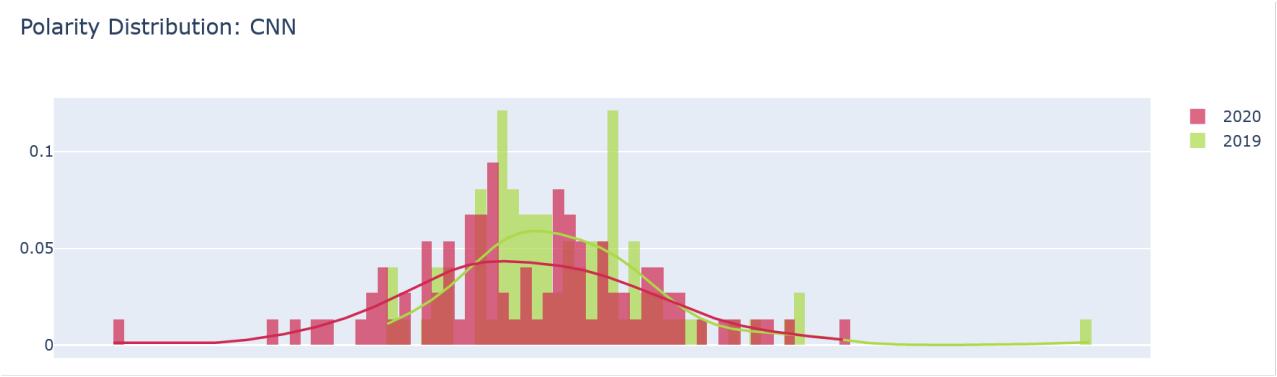
- https://www.cnn.com/
- https://www.newyorker.com/
- https://www.theguardian.com/

These websites require an SSL certificate which is not configured on Python installations on MacOS. If you are using a MacOS based device refer to the following StackOverflow answer to configure it: https://stackoverflow.com/a/53310545. If you are using WSL the program does not show any output; it may be a result of an active issue in WSL and can be fixed using the instructions in this StackOverflow answer: https://stackoverflow.com/a/63578387. The program also creates directories to store the data for efficiency and to store the graphs that are being output.

New Changes

After receiving feedback, we decided to make a few changes to our original proposal. We also decided to scrap one of the libraries we were going to use to process the data for us, and do the heavy lifting - regarding processing the data - ourselves . This would require our own implementation of a web scraping module and sentiment analysis. We have also introduced hypothesis testing to quantitatively measure changes in the average polarity of the articles and measure the statistical significance of our results. We have increased the computational complexity by just using plotly in order to display our results using interactive graphs.

Discuss, Analyze and Interpret results



Looking at the distribution of polarities of top 100 CNN articles we see that the 2020 article in red skew negatively, to the left whereas the 2019 articles skew positively to the right. This is confirmed by the difference in the average polarity we see which has a p-value below 0.05.

Source	mean 2019	mean 2020	Difference	p value
CNN	8.56	5.87	2.690284209679896	0.031
NewYorker	9.9	7.59	2.313482667185829	0.0552
Guardian	9.38	8.91	0.4759168492661008	0.6794

This shows that our results were statistically significant for CNN and that articles have become more negative over 2020. As for The New Yorker we believe that the results match, but our smaller sample size of just 25 articles may be what caused a p-value slightly over 0.05. Regardless, we still come to the same conclusion that articles have become more negative in The New Yorker. The lack of significance for The Guardian was expected as this was a list of opinion pieces picked by the writers however this was the only dataset we could find outside the USA.

Our project went successfully; we were able to make pre-COVID and current COVID-related models that gave an in-depth answer to our question. However, the most important part of the project was answering our question, and after finishing the code for our project, we did not get an answer that heavily favoured one side or the other. While the pandemic did have an impact on the pessimism we see in the media, the impact was not nearly as large as we had thought it would be; the best answer would be to say that it had a moderate impact on the news. We had expected to see a much more significant result, but at the end of the day, we did have an answer to our question, albeit not the one we were expecting. However we can say with certainty that an impact existed through our use of hypothesis testing.

Whilst developing the program to answer our question, there was one major limitation we encountered: a lack of data. We found that mainstream organizations tend not to publish their most-read articles of the year. In fact, only a few of these organizations did publish their most-read articles, which meant getting our data took quite a bit of time and patience. This led us to alternate strategies such as scraping the Internet Archive however this proved to be too slow as it is built for storing huge amounts of data. It took over 45 minutes to retrieve the data for CNN during 2019 so we decided to scrap that attempt. However, after a bit of searching, we were able to create a sufficient dataset for our project.

Of course, answering a question - especially one as vague as ours - never stops at just that. There were definitely some ways we could have expanded on our project, and developed it into a much larger one. The first step to this would have been expanding our answer to a more global perspective. For instance, seeing how different countries' articles' polarity scores differ from the Canadian and American news sources would have shifted our answer from one that panders specifically to North American media to the whole world. As well, it would have been interesting to see how the new variants affected the news in comparison to the original COVID outbreak news. This would have helped us get more of an idea of whether it was the idea of COVID as a whole driving the negativity in current media or the individual variants instead. The final step would have been predicting changes to other viruses in a similar manner - such as SARS and Ebola. This would have effectively completed the project, answering every possible facet of the question we asked ourselves in the first place.

While facing limitations and obstacles, we were able to effectively answer our question. Additionally, we developed our Python programming tool set by successfully creating a program that both analyzed the polarity of numerous articles and represented it visually for a clear, concise analysis. While there were definitely some next steps we would have loved to expand upon had we been granted more time, we feel satisfied at being able to say that we achieved our goal, and became better programmers along the way.

Works Cited

"Coronavirus: How Much News Is Too Much?" BBC Worklife, BBC, https://www.bbc.com/worklife/article/20200505-coronavirus-how-much-news-is-too-much.

CNN's Top 100 Digital Stories of 2019 — CNN. https://www.cnn.com/2019/12/22/us/top-100-digital-stories-2019-trnd/index.html.

"Plotly Python Open Source Graphing Library" Plotly Phython Graphing Library — Python — Plotly, https://plotly.com/python/.