

Monitoring and mitigating misinformation that disrupts adaptation to climate driven health threats

Students: Roberto Lopez-Trigo, Colleene McCurdy, Nathan Chiche, Brian Rodriguez, Kyle Ghani, Kyle Zimmer, Javier Oliva, Jeffrey Quispe, Khizar Nasir, Gabriel Prieto

Mentor: Sam Malloy, Mowafak Allaham, Ayse Lokmanoglu

Instructor/Faculty: Masoud Sadjadi, Florida International University

PROBLEM

Misinformation has a great impact on the public since people are obtaining their information mostly from social media and other online sources. Navigating this chaotic space is overwhelming. People unknowingly spread information from these questionable sources, magnifying the problem. One of the most concerning areas of misinformation is climate change.

In order to shed light on the issue, our team has compiled engagement data related to climate news articles from Newswhip. Using this data we are able to analyze trends and point out topics of concern.

CURRENT SYSTEM

The monitoring and mitigating information project aims to spread awareness about climate misinformation and the public perception of climate related issues.

User's can focus on multiple sources and filter the data according to their needs. This data could be filtered by engagement type, publication date, and top articles. That data can then be analyzed using pie charts, histograms, bubble charts and line graphs. We used BERTopic to cluster the data into meaningful groups allowing us to find topics related to a search term and thus the articles related to that topic.

Various climate disaster types can also be found on a United States map with the number of incidents separated by states. Estimates of the public's opinion on certain climate questions can also be seen on the survey data page. A breakdown of public interest into climate news can be found on the climate relevancy page.

REQUIREMENTS

Hardware:

- Mac
- Linux

Software:

- Python
- PostgreSQL
- BERTopic
- DASH

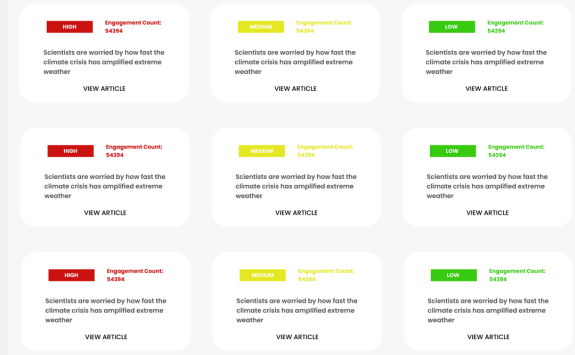
Version Control:

- **GitHub** (<http://www.github.com>).

CONTRIBUTION

Features and Efficiency – Being part of the features team I assisted in the creation of new addition to the application. One of the highlights of the work was the creation of cards of articles that help distinguish the amount of claims they had. Also being part of the efficiency team, I assisted in the clean up of the code making it more readable as well as more performant.

Database – I was able to transfer all data we had in the form of CSV to our database using a python script. This will increase the overall performance of the application. Using Heroku I was able to put the database online for remote access.



Scraper - I assisted the team leader in debugging the scraper used to get new sources of data.

```
dfrain = pd.read_csv('Clean-Data-Links/Sat_Climate_MBF_43321.csv', encoding='latin1')
# dfrain = dfrain.sample(n=12, replace=True, random_state=1)
dfrain=dfrain.head(1000)
list_of_urls = dfrain['link'].tolist()
total = len(list_of_urls)
print('total links', total)
def scraper(list_of_urls, startP, endP, fileName):
    if endP == "last":
        list_of_urls=list_of_urls[startP:]
    else:
        list_of_urls=list_of_urls[startP:endP]
    rows = []
    # print(len(list_of_urls))
    for link in list_of_urls:
        try:
            a = Article(url="{} {}".format(link, language="en"))
            a.download()
            a.parse()
            author = a.authors
            text = a.text
            title = a.title
            # print("{}\n\nNewspaper={}\nTitle={}\n\nText={}"
            row = {'url':link,
                    'author':author,
                    'text':a.text,
                    'title': title}
            rows.append(row)
        except Exception as e:
            # print(e)
            row = {'url':link,
                    'author': 'N/A',
                    'text': 'N/A',
                    'title': 'N/A'}
```



DASHBOARD

Monitoring and Mitigating Misinformation:

Climate Dashboard



SUMMARY

We compiled the work from the previous capstone iteration and turned it into a DASH multi page application with analytical functions and visual aids in the form of dynamic graphs.

We used BERTopic to cluster the articles in meaningful groups to find relevant documents to the user's searched term.

All the Newswhip data static files that were being used as a reference for the scripts were put into a Postgres database. We tested the database by building a proof of concept using optimized queries.

Created a scraper that used Multithreading

Added the whole UI

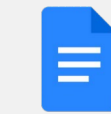
Optimized application*

Implemented BERTopic

Deployed to Github Pages

Deployed Database- next capstone needs to continue

IMPLEMENTATION



Google Docs - Used to keep track of everyone's contributions to the project. Multiple meetings and their information has been recorded through Google Docs.



Github - A site we used to access our code. For the group to test and add changes to the project.



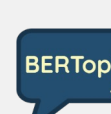
Plotly - Allows a point & click interface. Assists in creation of our dashboard.



PostgreSQL - Relational database management system.



Python - The main language that we used for this project.



BERTopic- Topic modeling technique we used to cluster articles based on their text and headlines.

REFERENCES



ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Sam Malloy. We thank Sam Malloy, Mowafak Allaham, and Ayse Lokmanoglu for their mentorship and assistance that we received throughout the senior design project.