**Datasets**

**Podcasts:** (transcribe folder) <https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Frutgers.box.com%2Fs%2Fxdgzu9p54dsn81g8ofm08g4uegcr2y9n&data=05%7C01%7Crm1667%40scarletmail.rutgers.edu%7Ccc503d7b29814dea6a4f08dba8bc18b1%7Cb92d2b234d35447093ff69aca6632ffe%7C1%7C0%7C638289298032978620%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=rf%2B5YrMCRuFY%2Bdh16E3o2mmr2NTJrfPCpEHSD%2FXmlRc%3D&reserved=0>

**Email:**  (very first link to download the db) <https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1zGrmTyGxRPrrwX4LXumJmnd_-oak_KA5vXZzlkD7CPc%2Fedit&data=05%7C01%7Crm1667%40scarletmail.rutgers.edu%7Ccc503d7b29814dea6a4f08dba8bc18b1%7Cb92d2b234d35447093ff69aca6632ffe%7C1%7C0%7C638289298032978620%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=l%2F2SWWfraAWxL5QY2A%2FN87M5cRsgR2GppMM4w%2F3PBIA%3D&reserved=0>

**TV data:** (download all the 6 archive-cc-2022.csv.gza\* files from Harvard dataverse)

Commands used to load tv data

1. cat archive-cc-2022.csv.gz{aa,ab,ac,ad,ae,af} > combined-archive-cc-2022.csv.gz
2. gunzip combined-archive-cc-2022.csv.gz

<https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fnotnews%2Farchive_news_cc%23data&data=05%7C01%7Crm1667%40scarletmail.rutgers.edu%7Ccc503d7b29814dea6a4f08dba8bc18b1%7Cb92d2b234d35447093ff69aca6632ffe%7C1%7C0%7C638289298032978620%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=hVUg2q8Vs9UDal753XEjz%2Bq8Zi5wGVlnqtMf87G%2F1wo%3D&reserved=0>

Research paper related to the project <https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Farxiv.org%2Fpdf%2F2308.02068.pdf&data=05%7C01%7Crm1667%40scarletmail.rutgers.edu%7Ccc503d7b29814dea6a4f08dba8bc18b1%7Cb92d2b234d35447093ff69aca6632ffe%7C1%7C0%7C638289298032978620%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=4TqYcdxvYRAM%2B1b39R43SZ8vixVedmldw9C5CqfYLmY%3D&reserved=0>

Github link for the project:

<https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling>

**Week 1, 2 : Sep 1 - Sep 11**

1. **Load the Datasets -** Done

Tv data was only available till October so only considering the data till October for the rest of the sources (shown on later pages)

**Emails:**

Total Emails count - 186618

2022 Emails count - 80100

2022 Emails count (till October) - **70765**

**Podcasts:**

Total Episodes - 10245

2022 Episodes - 6949

2022 Episodes (till October) - **5687**

**TV Data:**

Total shows - 917221 (32 GB)

2022 Shows of selected 5 broadcasters - **43284** (1.7GB)

Files created at the end of this step as raw files:

1. emails\_data\_raw.csv
2. podcast\_data\_raw.csv
3. tv\_data\_raw\_2022.csv
4. **Read the Research Paper -** Done

Read the paper and understood the process and steps involved in the project. Some of the steps involved were, extracting data using web scraping and crawlers, pre-processing the data to remove html tags, emojis etc, converting the text into passages of certain words length, passing it to MPNet LLM to obtain the embeddings, performing DP-Means Clustering to obtain narrative topics, and finally analyzing the results and optimizing the performance.

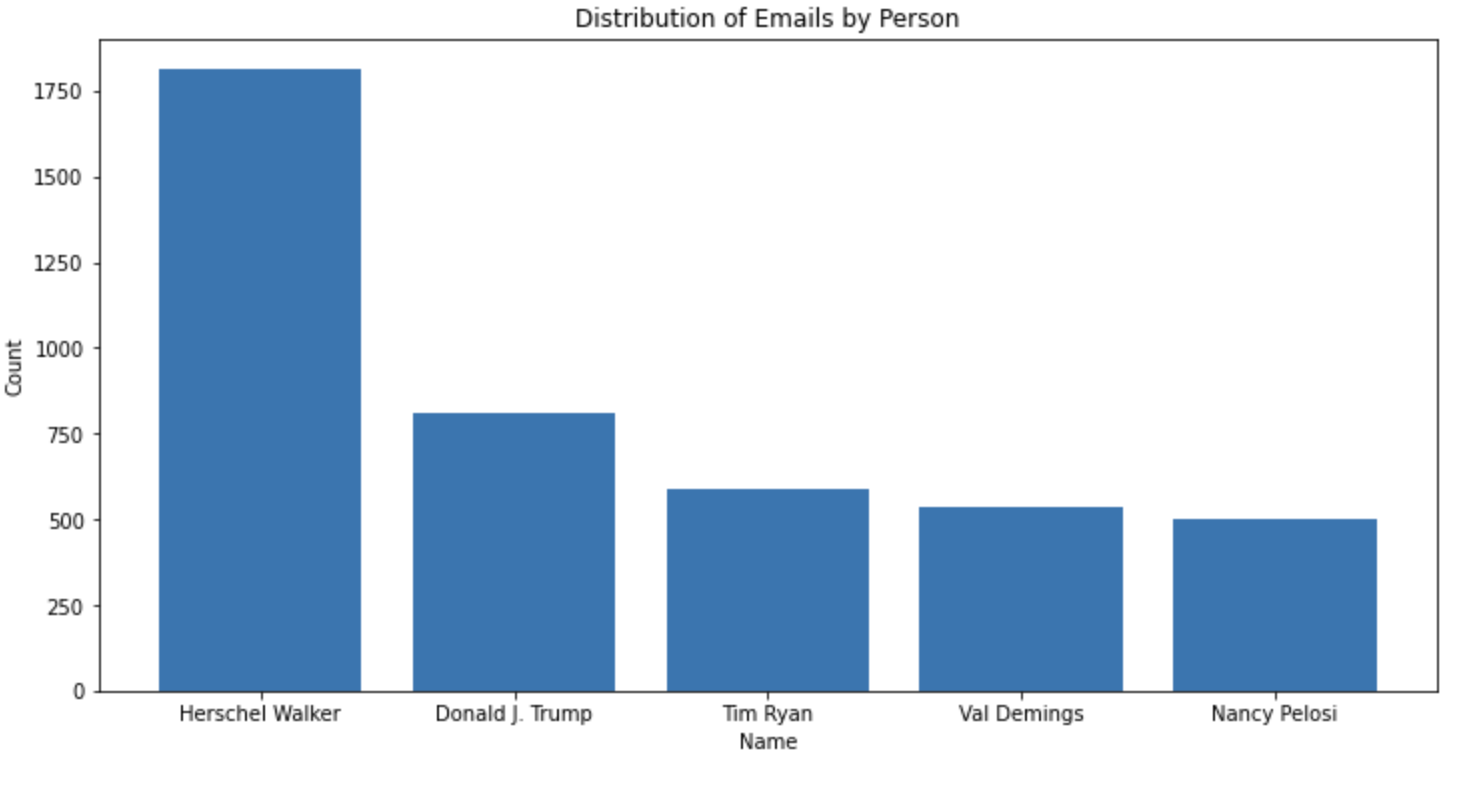
Note: Primary code run was done using Amarel cluster compute power.

**Week 2: Sep 12 - Sep 20**

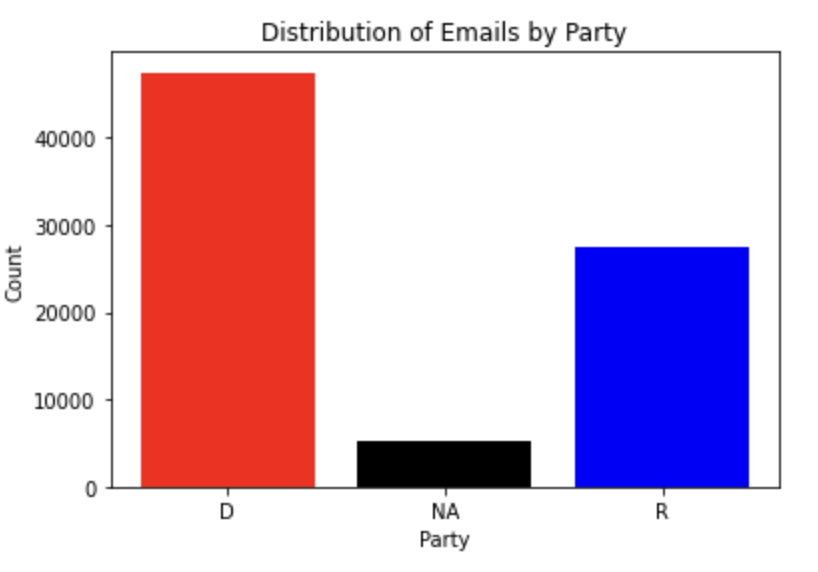
1. **Clean and perform EDA** ([data-standardization-eda.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/data-standardization-eda.ipynb)) - Done

Only 2022 data was considered for this analysis. Data was filtered for the same from the original datasets.

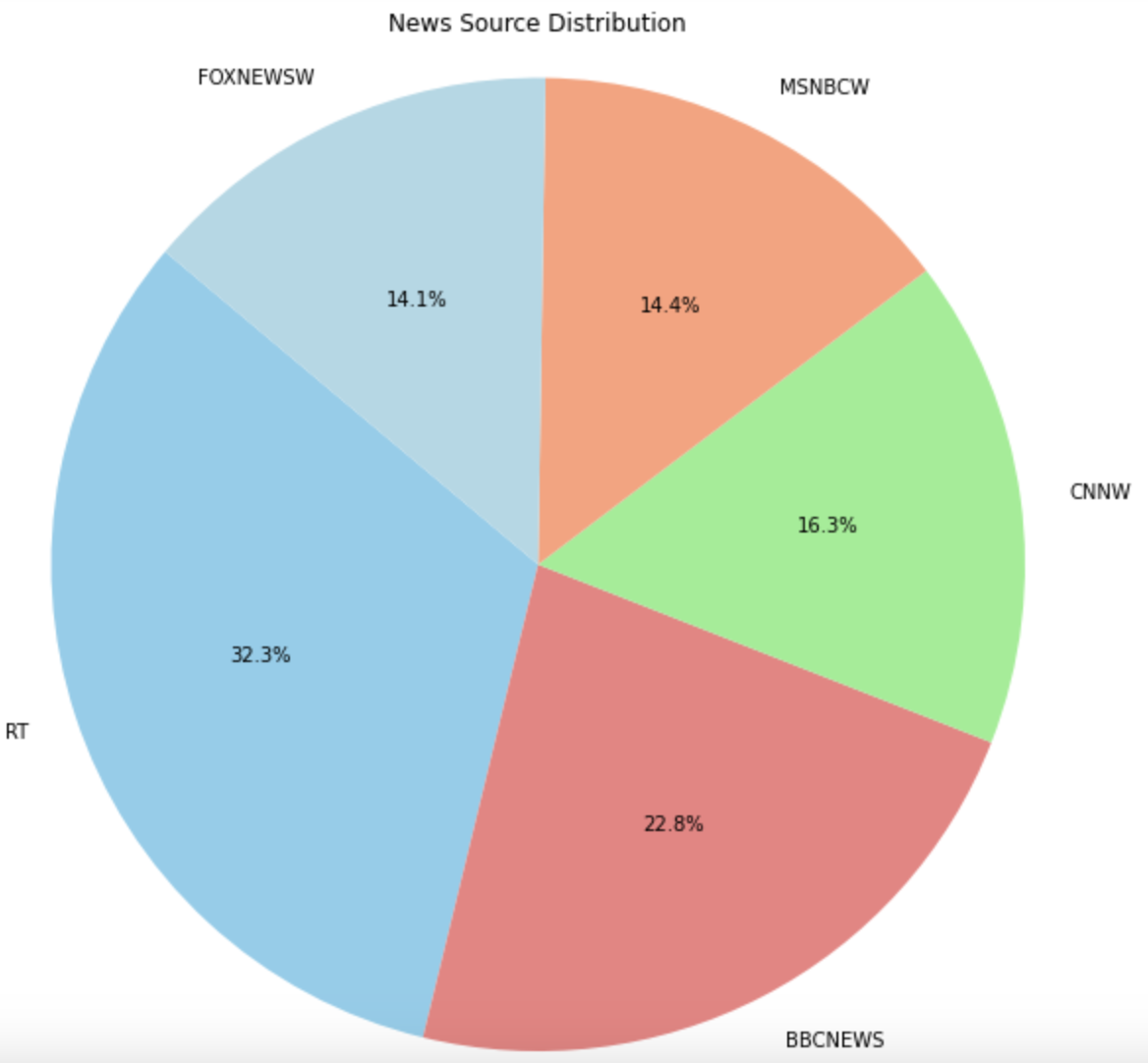
1. Most emails by person



1. Emails by Party



1. TV Broadcasters distribution

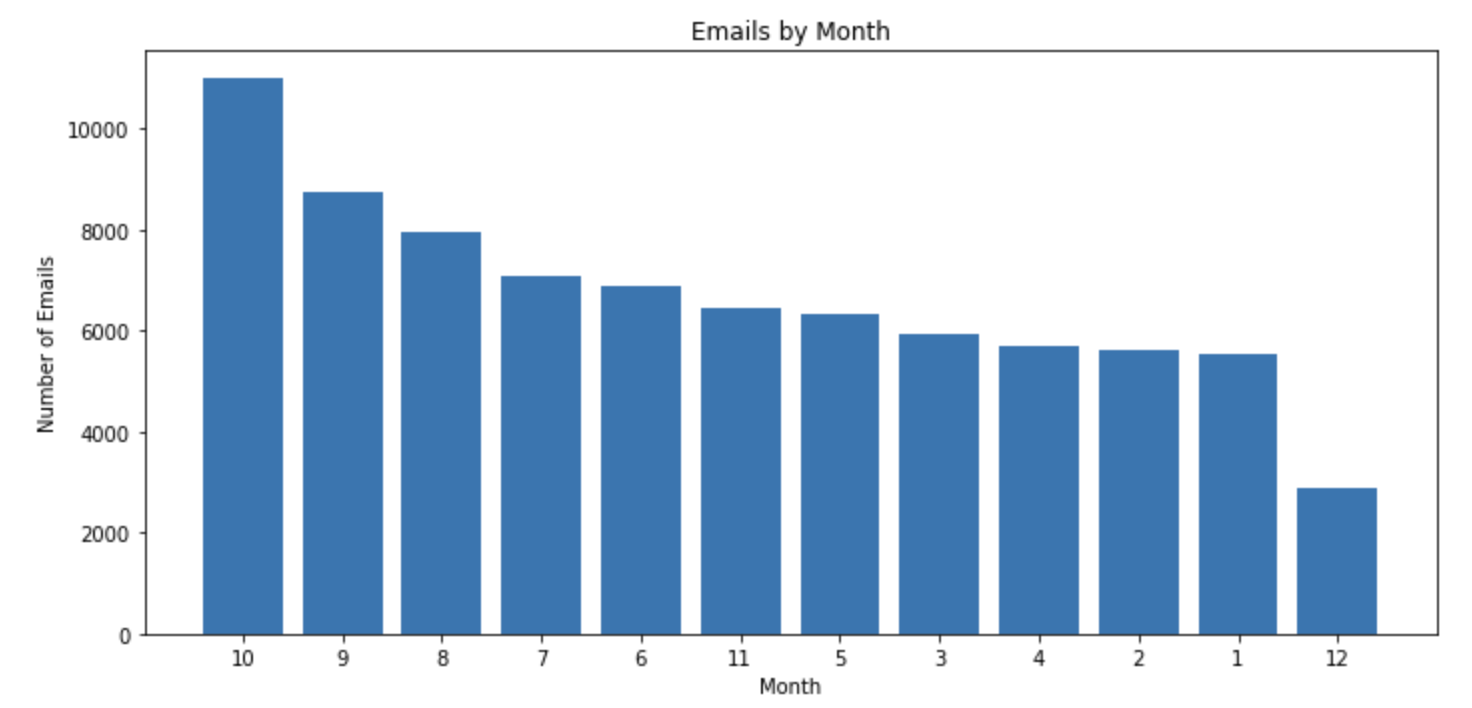


1. Top 5 podcasts with most episodes



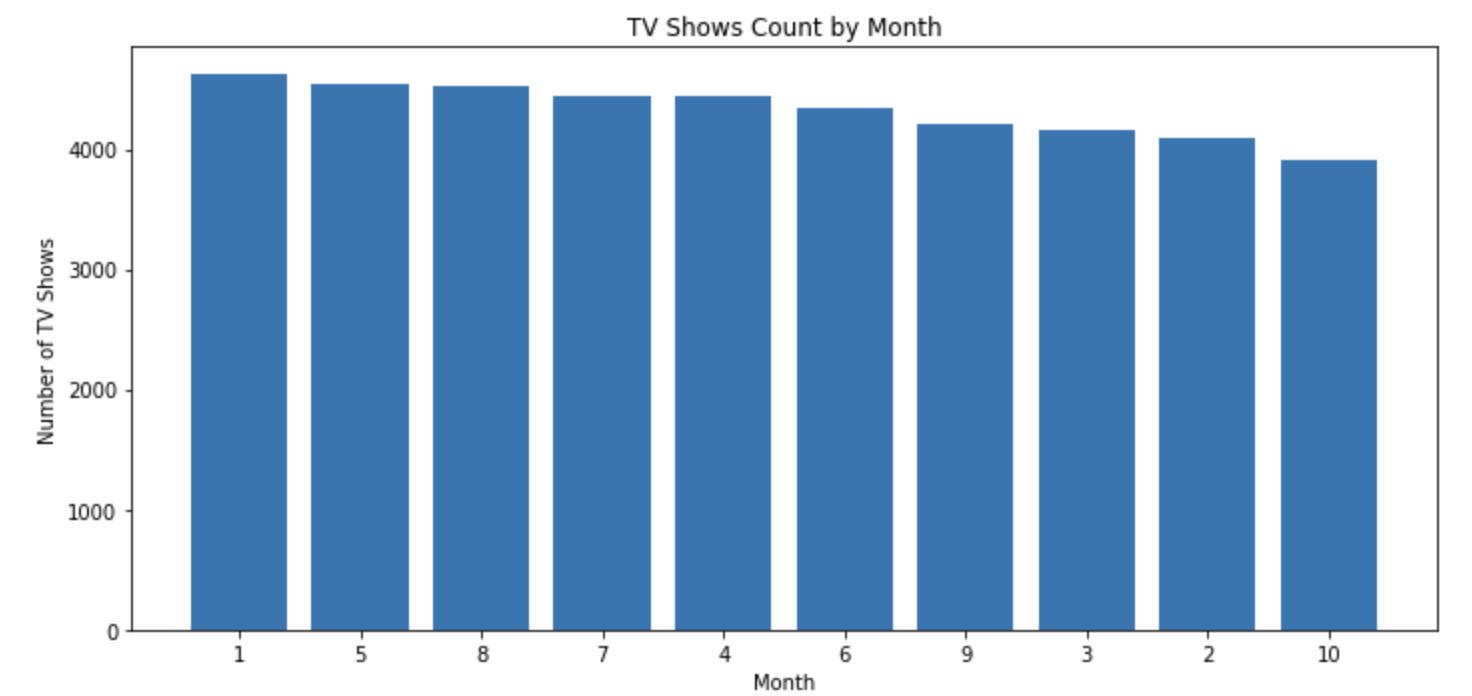
1. Data comparison by month for all the sources

Email:

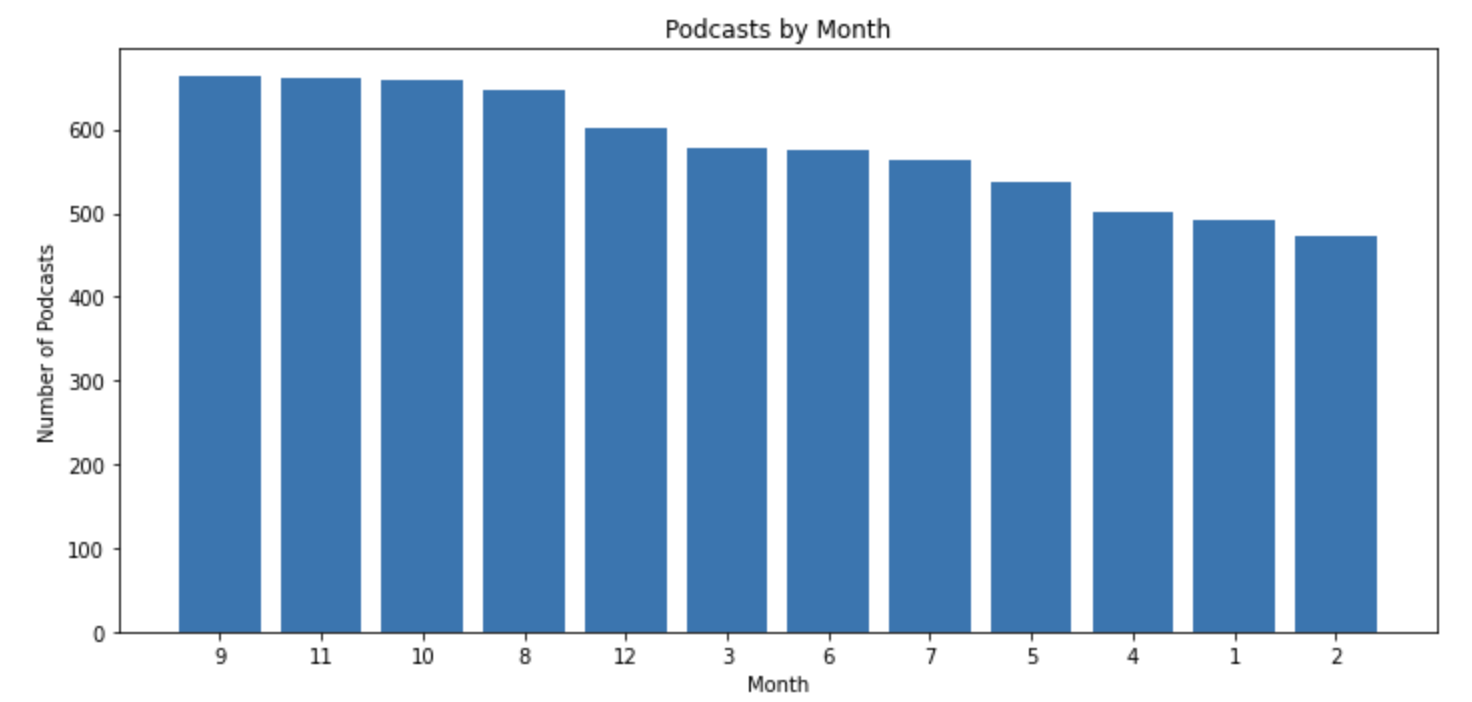


TV: Missing for the month November and December

Note: Considering only 10 months data for the rest of the sources as well



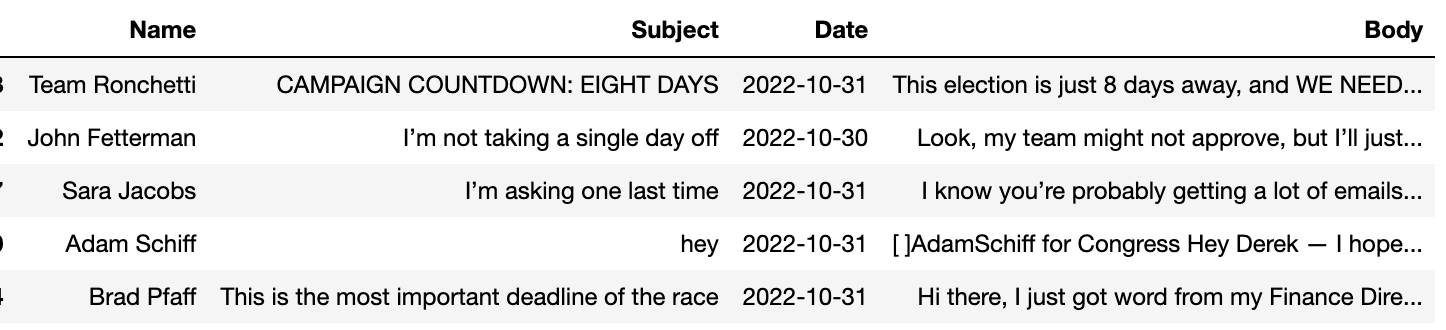
Podcasts:



1. **Standardize the data -** Done

Final Columns were identify for each source and are as follows:

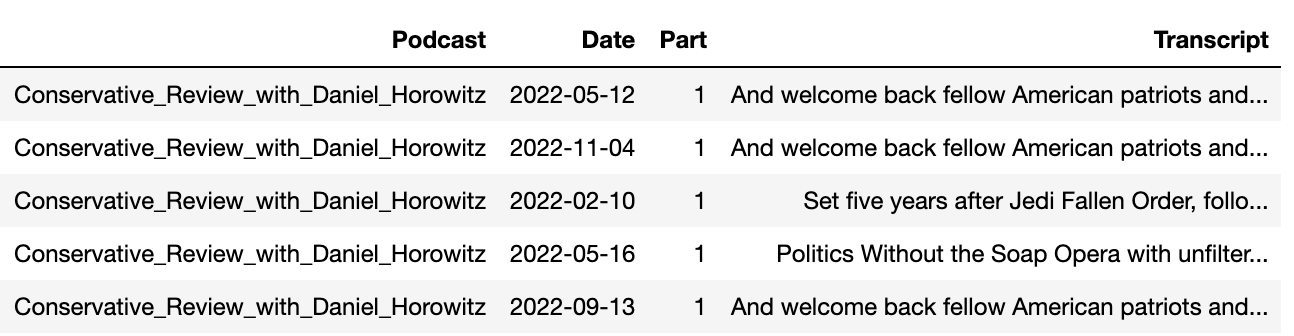
1. Email:



1. TV:



1. Podcast:



1. **Perform Pre-processing** ([pre-processing.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/pre-processing.ipynb)) - Done

As part of pre-processing, the below steps were performed for each source in reference to the steps mentioned in the research paper:

1. Remove URL, HTML tags, and Emojis
2. Convert the text into paragraphs (not done as most of the data is single paragraphs for the most part)
3. Convert the data into passages of maximum length 100 words (only complete sentences)

Files created at the end of this step as processed files:

1. emails\_data\_processed.pkl
2. podcast\_data\_processed.pkl
3. tv\_data\_processed.pkl

Note: Pickle files format is used to maintain the data type intact. The text is split into a list of passages, when the data is saved to csv and reloaded, the list changes to string. Hence pickle format. Alternatively, parquet format can also be used.

To convert the text into passages, another approach was also used to split the text based on similarities and minima but rejected it as the split was very uneven.

* [email\_passages.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/email_passages.ipynb)
* [podcast\_passages.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/podcast_passages.ipynb)

**Week 3, 4: Sep 21 - Oct 2**

1. **Get Embeddings for the data -** Done

* Obtain embeddings for email and podcast data from MPNet LLM
  + [email-embeddings.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/email-embeddings.ipynb)
  + [tv-embeddings.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/tv-embeddings.ipynb)
* Tv embeddings - pending (due to Amarel cluster issue)
  + [tv-embeddings.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/tv-embeddings.ipynb) - Code never ran due to memory/compute issues, embedding not obtained

Files created at the end of this step as numpy zip files that stored the embeddings:

1. email\_embeddings.npz
2. podcast\_embeddings.npz
3. **Run the DP-Means Clustering to identify narratives**

Unable to use it due to installation problems on Amarel Cluster and memory problems on local machines.

1. **DBSCAN and K-Means -** Done

* Implemented K-Means clustering with 10 clusters for now (due to computation limitations) for email data
* Created visualization and word cloud to understand the distribution of the clusters and the topics associated with each cluster.
* [clustering-kmeans-emails.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/clustering-kmeans-emails.ipynb), [clustering-kmeans-podcast.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Archive/clustering-kmeans-podcast.ipynb)

Note: Not implemented for tv data as embeddings were not obtained

**Week 5: Oct 2 - Oct 12**

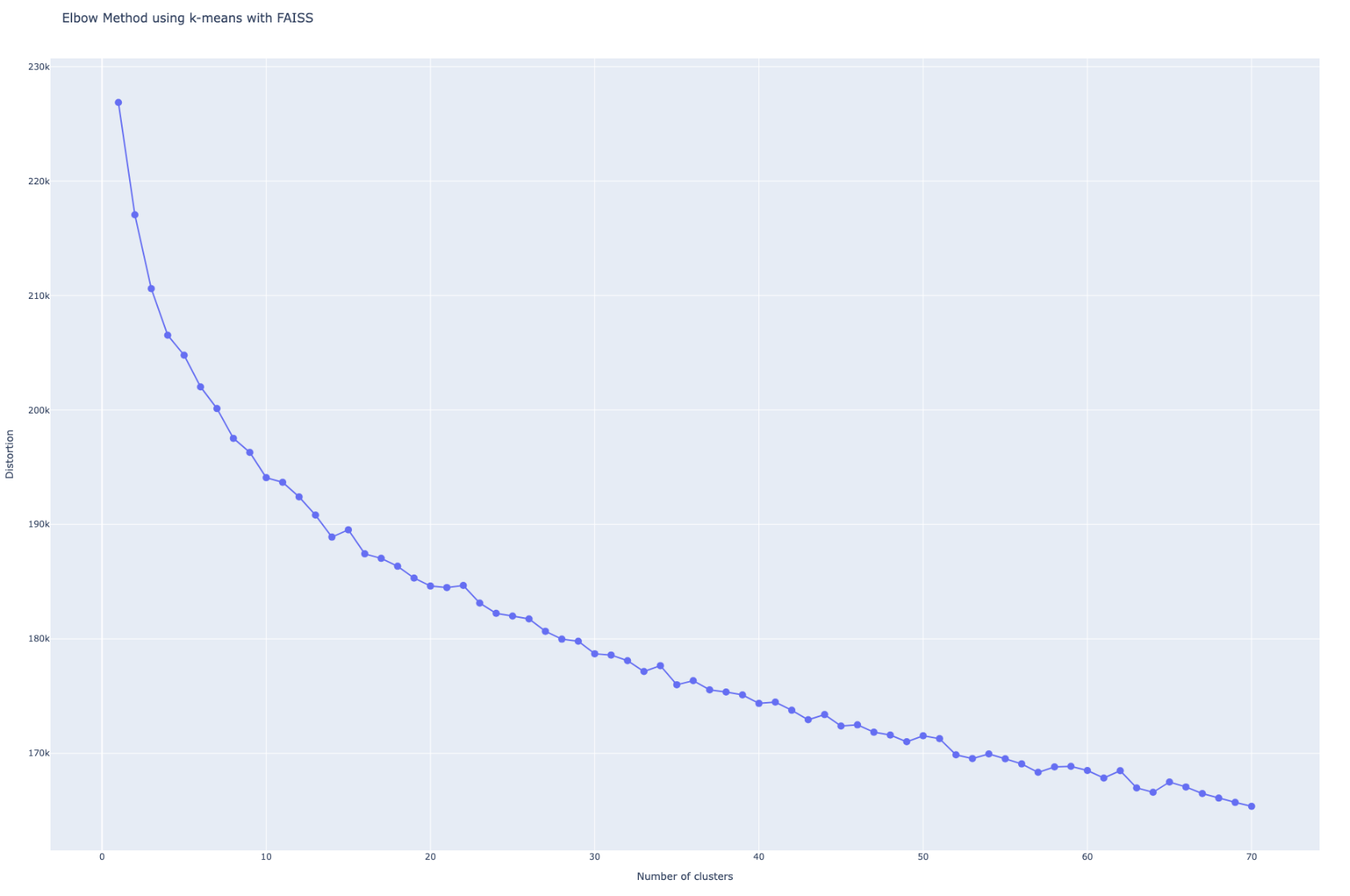
1. **Amarel Cluster issue**

* Get the Cluster issue resolved and get embeddings for Tv data - raised tickets for both job issues and package installation support.

1. **Try FAISS on K-Means -** Done

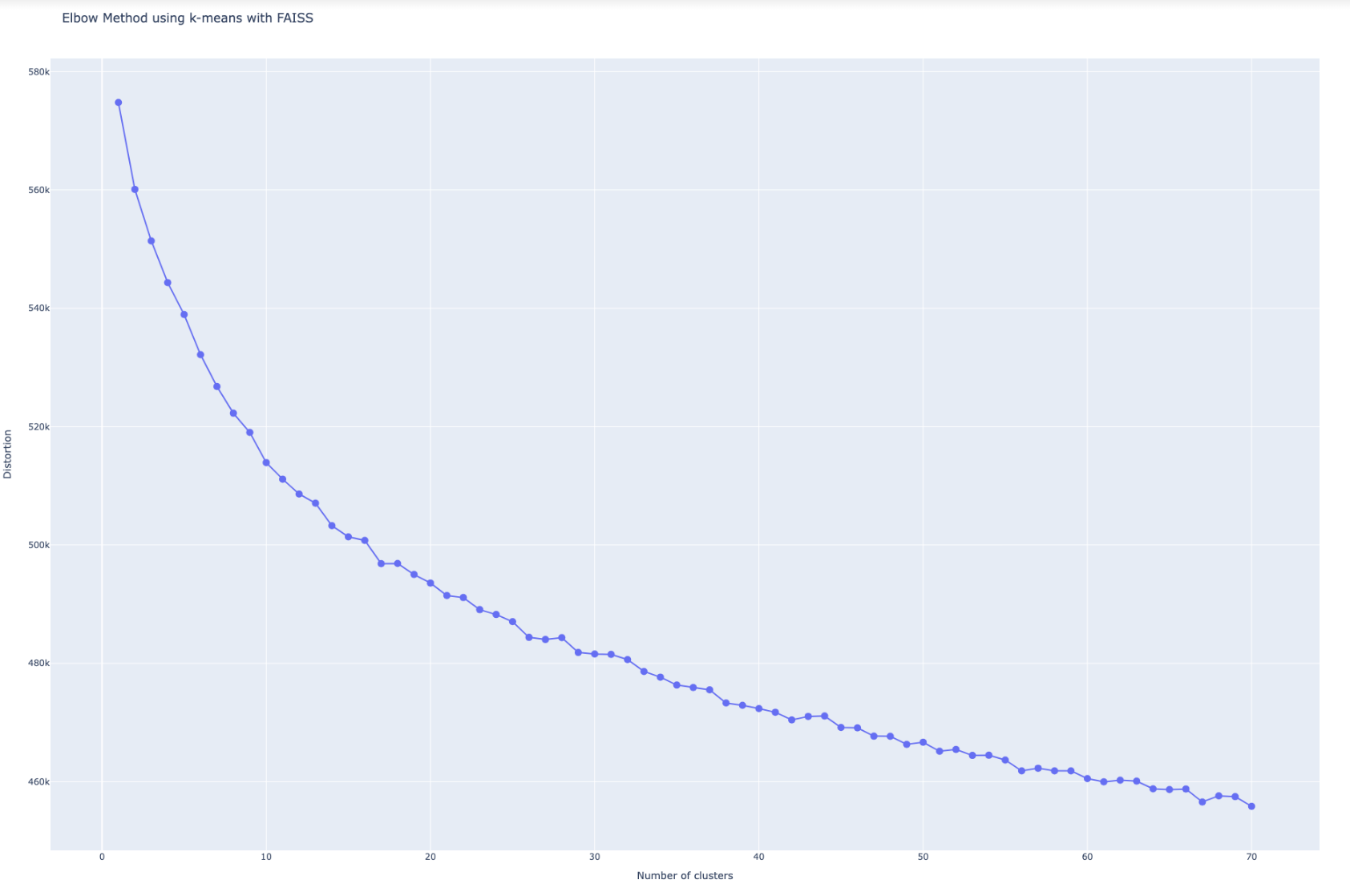
* Obtain the ideal number of cluster from elbow method
* Use FAISS to improve the speed of K-Means clustering

Email Data



No Clear elbow pattern but identified potential # cluster to be 14, 20, 24, 33

Podcast Data:



No Clear elbow pattern but identified potential # cluster to be 17, 26, 37, 42

Note: Moving on from K-Means using FAISS to BERTopic as we need dynamic clustering without mentioning the cluster count.

1. **Try Link Transformer**

* Implemented the code for sample data, need to run the code after resolving the Amarel issue is cleared.

**Week 6: Oct 13 - Oct 19**

1. **Obtain Clusters for all data sources using BERTopic**
2. Experiment with hyperparameters to figure out the right once. - Done
3. Get around 150-200 clusters for each data source. - Done

Clusters obtained for Emails and Podcast data. Pending on Amarel to run it for TV data.

Code for the task:

* [clustering-bert-emails.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/clustering-bert-emails.ipynb), [clustering-bert-podcast.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/clustering-bert-podcast.ipynb), [clustering-tv.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/clustering-tv.py)

Files created at the end of this step store the cluster information:

1. email\_cluster\_bert.pkl
2. podcast-bert-clust.pkl
3. tv-bert-clust.pkl

Files created at the end of this step store the cluster ids for each passage (extension of processed files):

1. email\_cluster\_ids.pkl
2. podcast\_cluster\_ids.pkl
3. tv\_cluster\_ids.pkl

Additionally, the Bertopic models for email and podcast were also saved as bertopic\_model in respective folders (the model size of tv was too large to store)

**Week 7: Oct 20 - Oct 26**

1. **Amarel Issues -** Cleared
2. Get the Job running issues resolved
3. Get the required packages installed
4. **TV Embedding**- on Hold
5. Obtain TV Embeddings

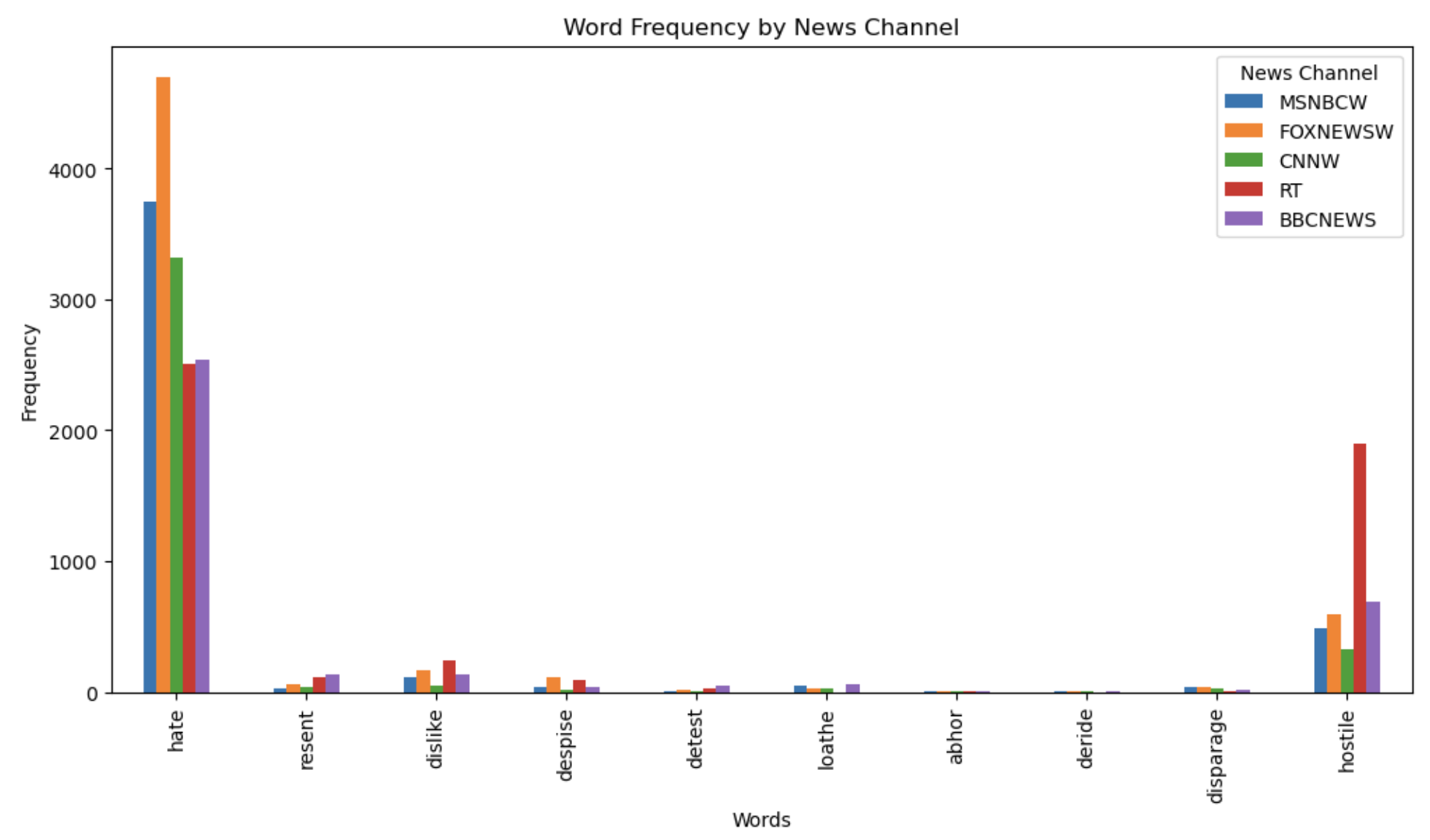
Note: As we are not using DBSCAN or K-Means, putting it on hold

1. **Keyword Analysis**

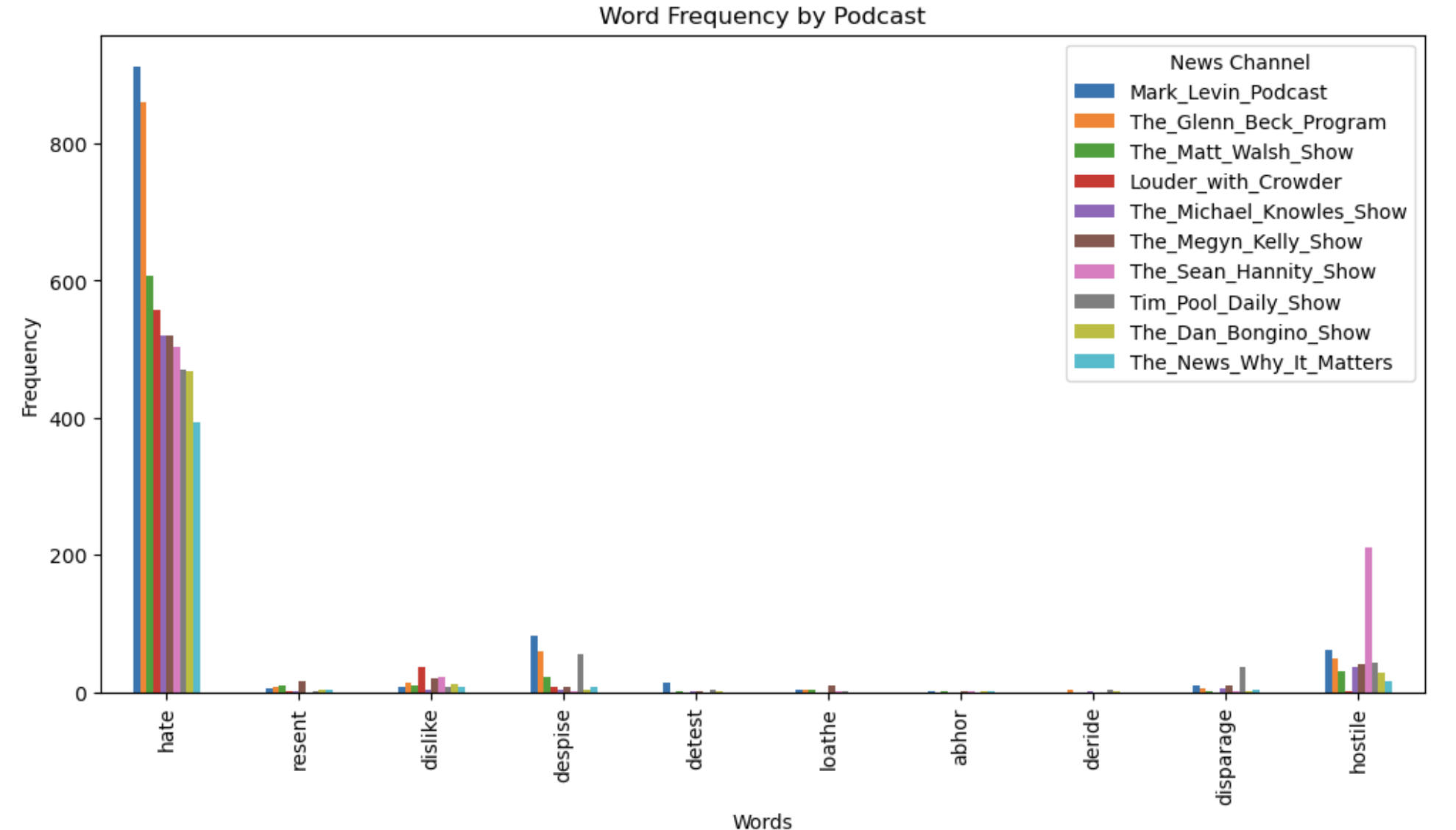
Refer to the paper to perform keyword analysis across the data sources (example - “HATE”, “RESENT”)

Paper - <https://cdn.theconversation.com/static_files/files/1255/Hate_on_Fox_News_draft_report_9-28-20.pdf>

1. Top 10 Channels using defined keywords - Done

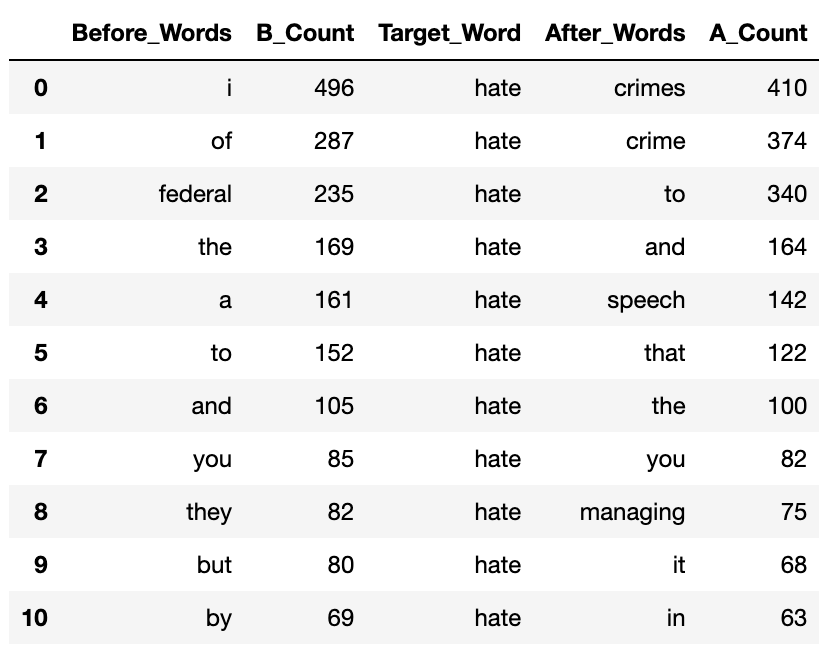


1. Top 10 Podcasts using defined keywords - Done



1. Context words analysis for TV data - Done

Note: The screenshot below shown is only for the CNN data



1. Context words analysis for Podcast data - Done



Code for the task:

* [Keyword-Analysis-Email.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/Keyword-Analysis-Email.ipynb), [Keyword-Analysis-Podcast.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/Keyword-Analysis-Podcast.ipynb), [Keyword-Analysis-TV.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/Keyword-Analysis-TV.ipynb)

**Week 8, 9: Oct 27 - Nov 8**

1. **Obtain Clusters for TV data using BERTopic**- Done

Run the code for obtaining cluster on Amarel for TV data (srun job)

1. **Similarities** ([cluster-similarities.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/cluster-similarities.py)) - Done
2. Get Jaccard Similarities for the clusters to compare the data sources based on representative words(top 10 words of each cluster)
3. Compare the clusters with >0.4 similarity scores across different data sources

1. **Link Transformer Similarities** - Done

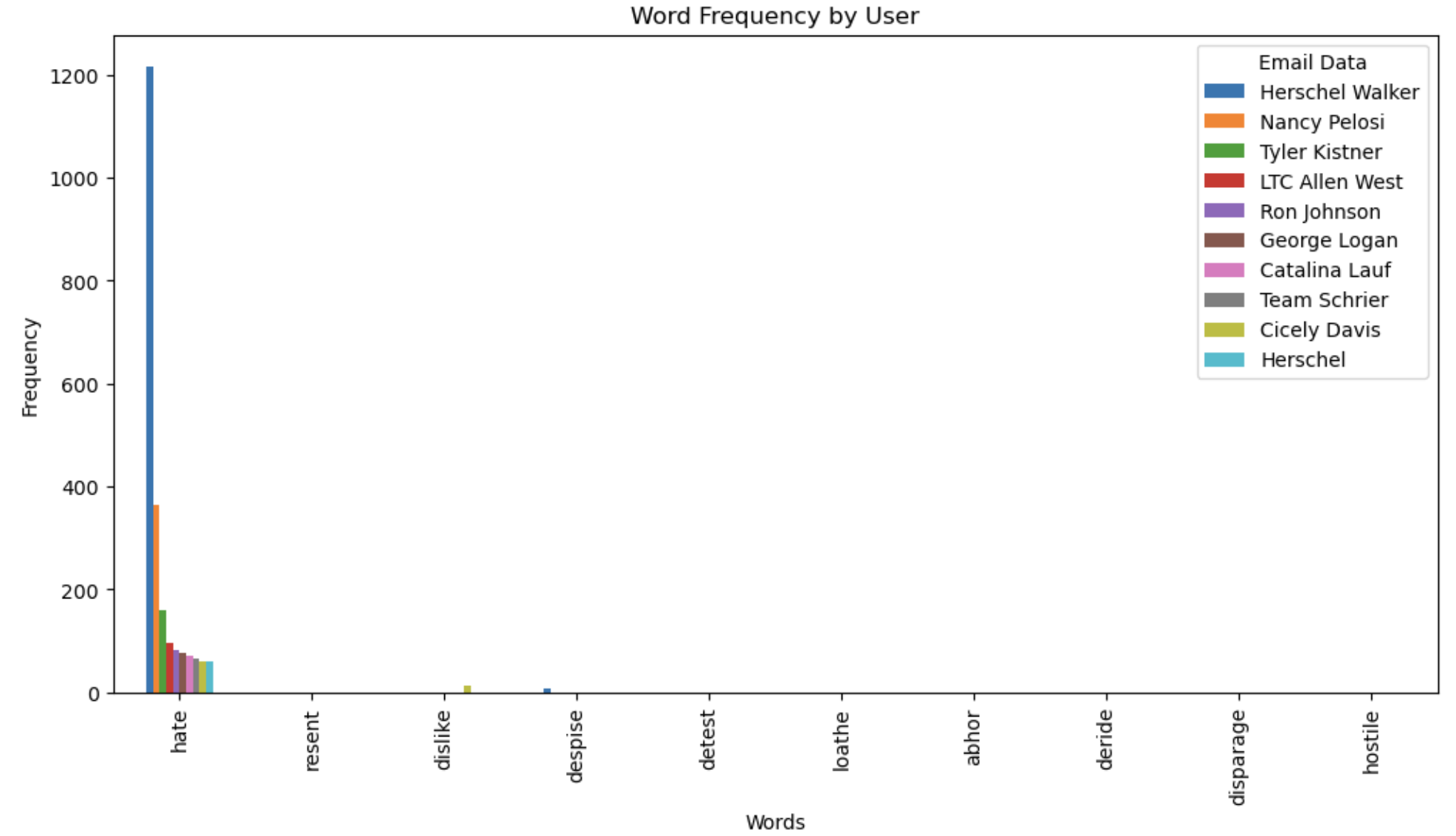
Run the code on the entire data to get the combined dataframe to compare and verify the jaccard similarities. (same code and files as mentioned above)

Code for the task

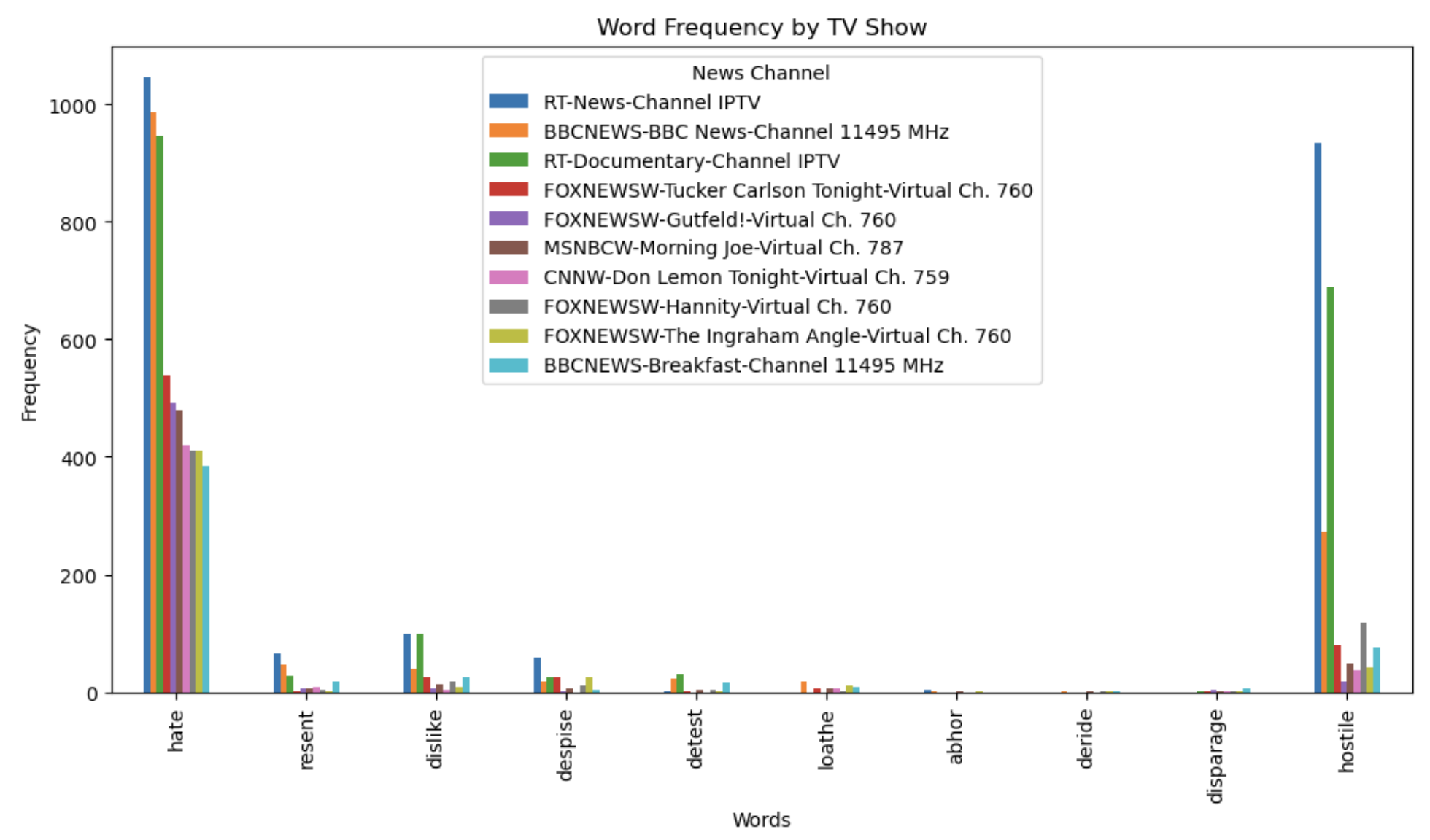
* [link-transformer-tv-em.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/link-transformer-tv-em.py), [link-transformer-tv-pod.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/link-transformer-tv-pod.py), [link-transformer-pod-em.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/link-transformer-pod-em.py)

Files created at the end of this step store the similarity scores for each pair of passages:

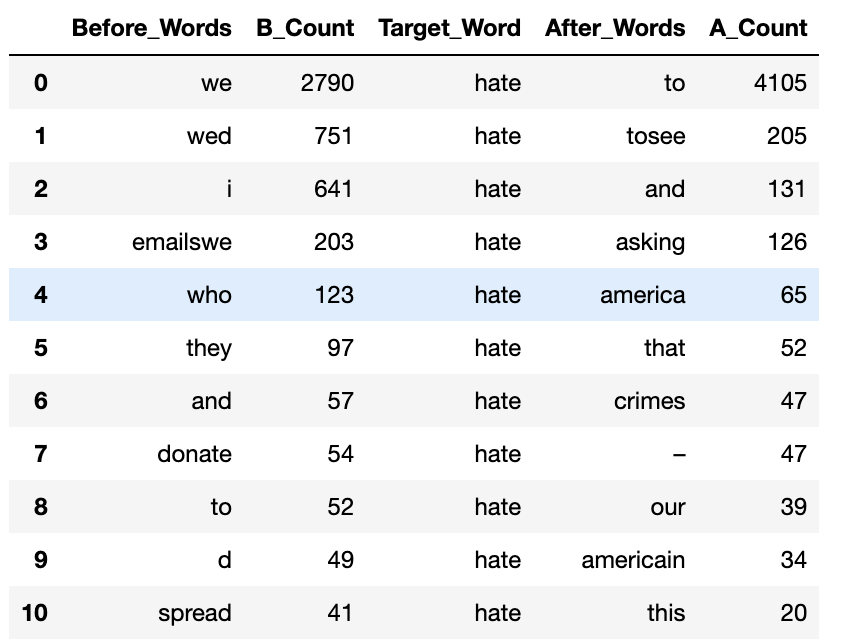
1. tv-email-passage-sim.pkl
2. tv-podcast-passage-sim.pkl
3. podcast-email-passage-sim.pkl
4. **Keyword Analysis**
5. Top 10 users(emails data) using certain keywords - Done



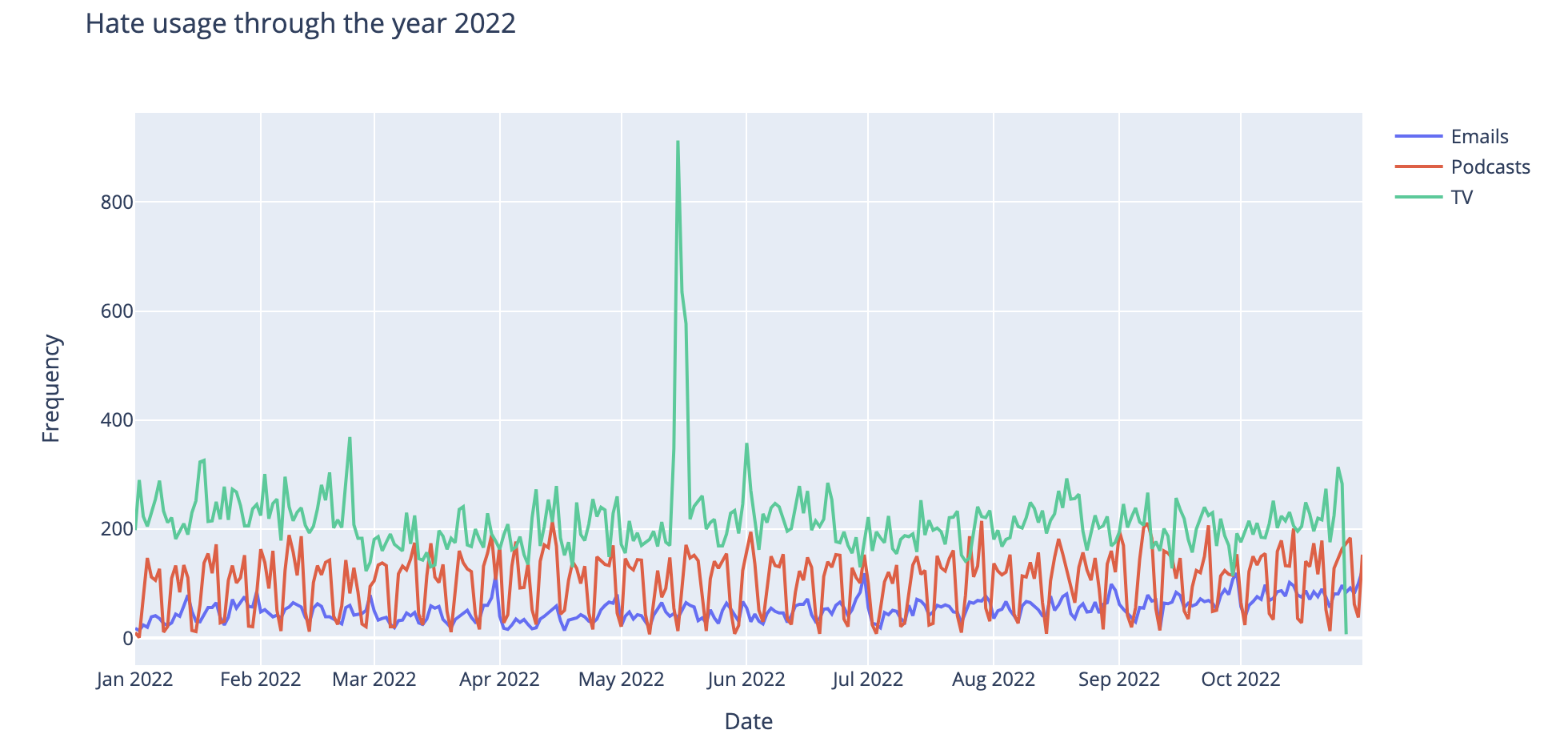
1. Top 10 TV Shows using certain keywords - Done



1. Context words analysis for Email data - Done



1. Time series plots for all the data sources - Done



Observation: Why is there a peak on May 15, 2022

Files created at the end of this step store the unigrams, bigrams, keyword analysis:

1. email-antipathy-analysis.pkl
2. emails\_uni\_bi\_neg.pkl
3. podcast-antipathy-analysis.pkl
4. podcast\_uni\_bi\_neg.pkl
5. MSNBCW\_uni\_bi\_neg.pkl
6. FOXNEWSW\_uni\_bi\_neg.pkl
7. CNNW\_uni\_bi\_neg.pkl
8. RT\_uni\_bi\_neg.pkl
9. BBCNEWS\_uni\_bi\_neg.pkl
10. TV\_uni\_bigrams.pkl
11. Tvshows-antipathy-analysis.pkl

Note: For TV data, analysis was done at channel level and show level hence files were created channel wise (5 channels).

1. **Get Top unigrams and bigrams for all data sources -** Done

Generate unigrams and bigrams and compare top of both of them across all data sources to identify unique unigrams/bigrams to certain forms of data source.

Code - [unigrams\_bigrams\_analysis.ipynb](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/unigrams_bigrams_analysis.ipynb)

**Week 10, 11 : Nov 9 - Nov 23**

1. **Cluster Similarity Analysis (Jaccard) -** [Cluster Analysis.docx](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Cluster%20Analysis.docx)
2. Manually look through the similar cluster passages (47), understand and document the similarities on a google doc.
3. Do a timestamp analysis of which source the narrative originated, identify trends if any, and plot a time series.

Update: (Same update shared via email)

Documented Narratives of 22 clusters for Email and Podcast clusters capturing:

1. Narrative title
2. Different topics discussed in the passages
3. Peak dates
4. Reasons for the peaks

However, I failed to identify actual similarities(Separate document created and shared via email and uploaded to git repository).

Next steps - Need to explore the accuracy of similarity scores, now that we have the labels, it can be easy to check the scores.

**Week 12: Nov 23 - Nov 30**

1. **Cluster Similarity Analysis (Jaccard)**
2. Manually look through the similar cluster passages (47), understand and document the similarities on a google doc.
3. Do a timestamp analysis of which source the narrative originated, identify trends if any, and plot a time series.

Update: From further investigation, it was identified that jaccard scores are computed based on representative words (top 10 words of each cluster) rather than the words of all passages of the clusters. This was also the reason why the scores for almost the same for many clusters there might be stopwords that might give the same and higher scores.

Updated the code to calculate the actual scores based on all the words (removing stop words as well). Currently documenting the clusters based on the new similarity scores.

1. **Keyword Analysis**
2. Why is there a peak on May 15, 2022?

Update: Took an initial stab at identifying the reasons by looking at the news from May 10- May 15 but could not identify clear reasons other than Russian invasion and Covid deaths count of America reaching 1 million.

1. POS tagging to identify the subjects, objects and causes for all the data sources

Update: Ran the code for the tv data but the results are not what we want, need to change the code.

**Week 13: Dec 1 - Dec 7**

* Run the code to get updated similarity scores
* Complete Project Documentation - Done
* Push everything to Github - Done
* Share the saved pickle files - to a box location or physical device ?

**NEXT STEPS**

1. **Cluster Similarity Analysis (Jaccard)**
2. Manually look through the similar cluster passages , understand and document the similarities on a google doc.
3. Do a timestamp analysis of which source the narrative originated, identify trends if any, and plot a time series.

Next Steps: run the code “[cluster-similarities.py](https://github.com/RohitMacherla3/narrative-detection-nlp-topicmodeling/blob/main/Code/cluster-similarities.py)” to get the jaccard similarities for clusters and manually document the narratives to check the actual similarities.

1. **Passage Similarity Analysis (Link-Transformer)**
2. Examine the passages, filter out the data and identify what each passage is referring to and if they are really related.
3. Do a timestamp analysis of which source the narrative originated, identify trends if any, and plot a time series.

Next Steps: Code run is already finished, data stored in pickle files as mentioned under link transformer section of week 8,9. Analyze the passages similar to the above task

1. **Keyword Analysis**
2. Why is there a peak on May 15, 2022? - No Progress
3. POS tagging to identify the subjects, objects for all the data sources - Need to implement the code
4. Dig deeper into unigrams and bigrams for each source and get a sense of it (what do the numbered unigrams indicate?) - unigrams and bigrams are obtained for all the data sources and stored in pickle files as mentioned under keyword analysis section of week 8, 9 . Need to analyze on it
5. **Tracking Largest Narrative**

Identify the narratives that are spoken a lot - Not started

**Resources**

1. Keywords Analysis Paper - <https://cdn.theconversation.com/static_files/files/1255/Hate_on_Fox_News_draft_report_9-28-20.pdf>
2. DP-Means - <https://github.com/BGU-CS-VIL/pdc-dp-means>
3. K-Means with FAISS
4. <https://towardsdatascience.com/how-to-speed-up-your-k-means-clustering-by-up-to-10x-over-scikit-learn-5aec980ebb72>
5. <https://www.blog.dailydoseofds.com/p/make-sklearn-kmeans-20x-times-faster#:~:text=To%20speed%2Dup%20KMeans%2C%20use,makes%20performing%20clustering%20extremely%20efficient>.
6. <https://github.com/facebookresearch/faiss>