**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: <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: <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**

**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)

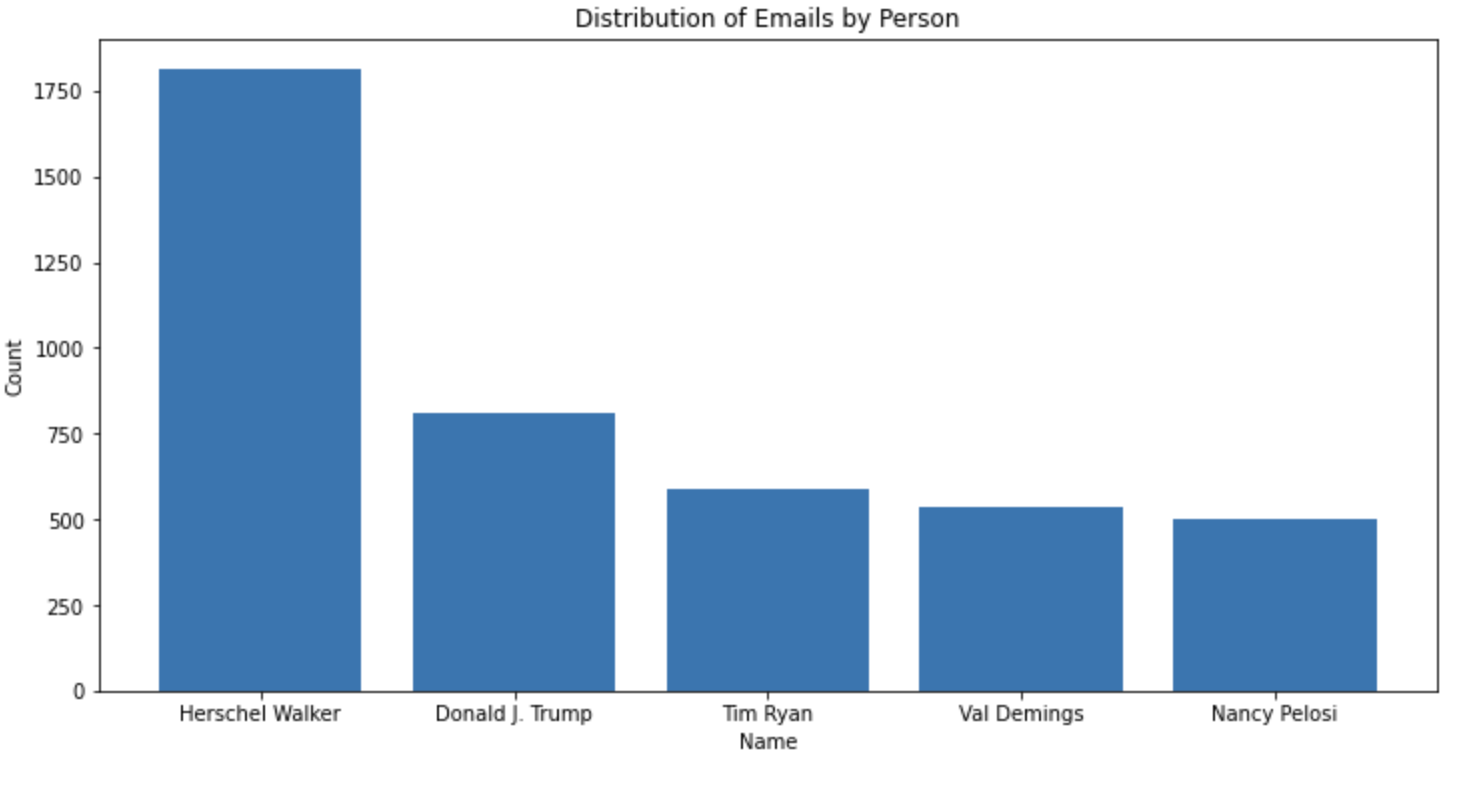
Commands used to load tv data

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
3. **Read the Research Paper**

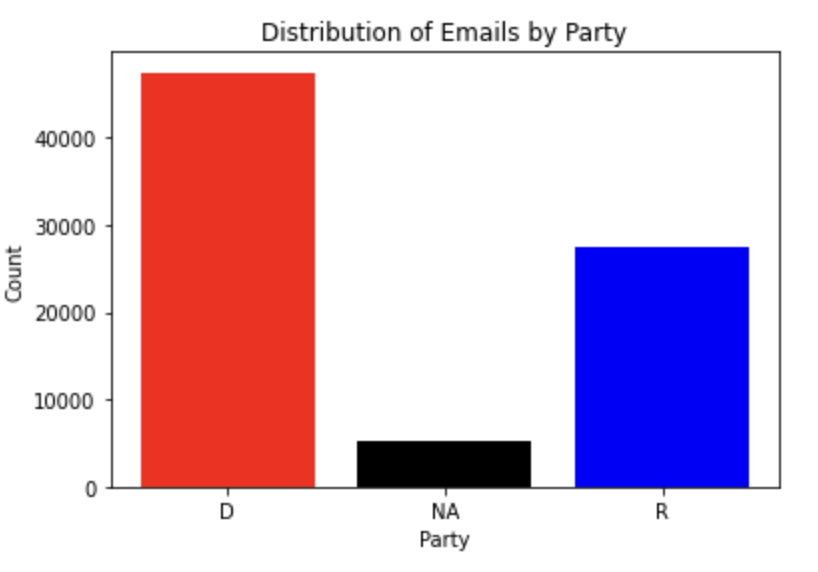
Read the paper and understood the process and steps involved in the project. Some of the steps involved were, extracting data using web scraping and crawlers, pre-processing the data to remove html tags, emojis etc, defining the narrative and passing it to MPNet LLM to obtain the embeddings, performing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

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

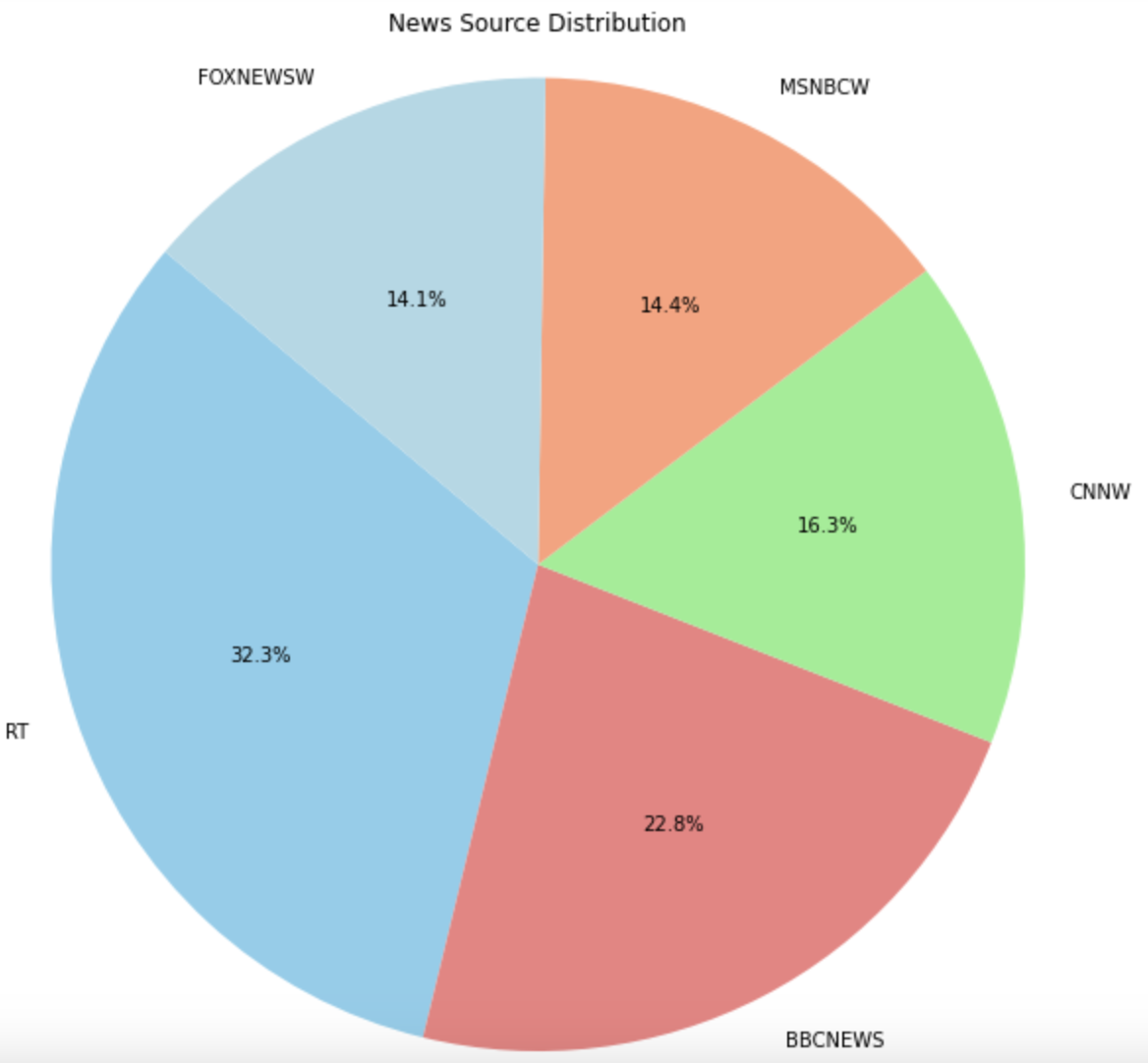
1. **Clean and perform EDA**
2. Most emails by person



1. Emails by Party



1. TV Broadcasters distribution

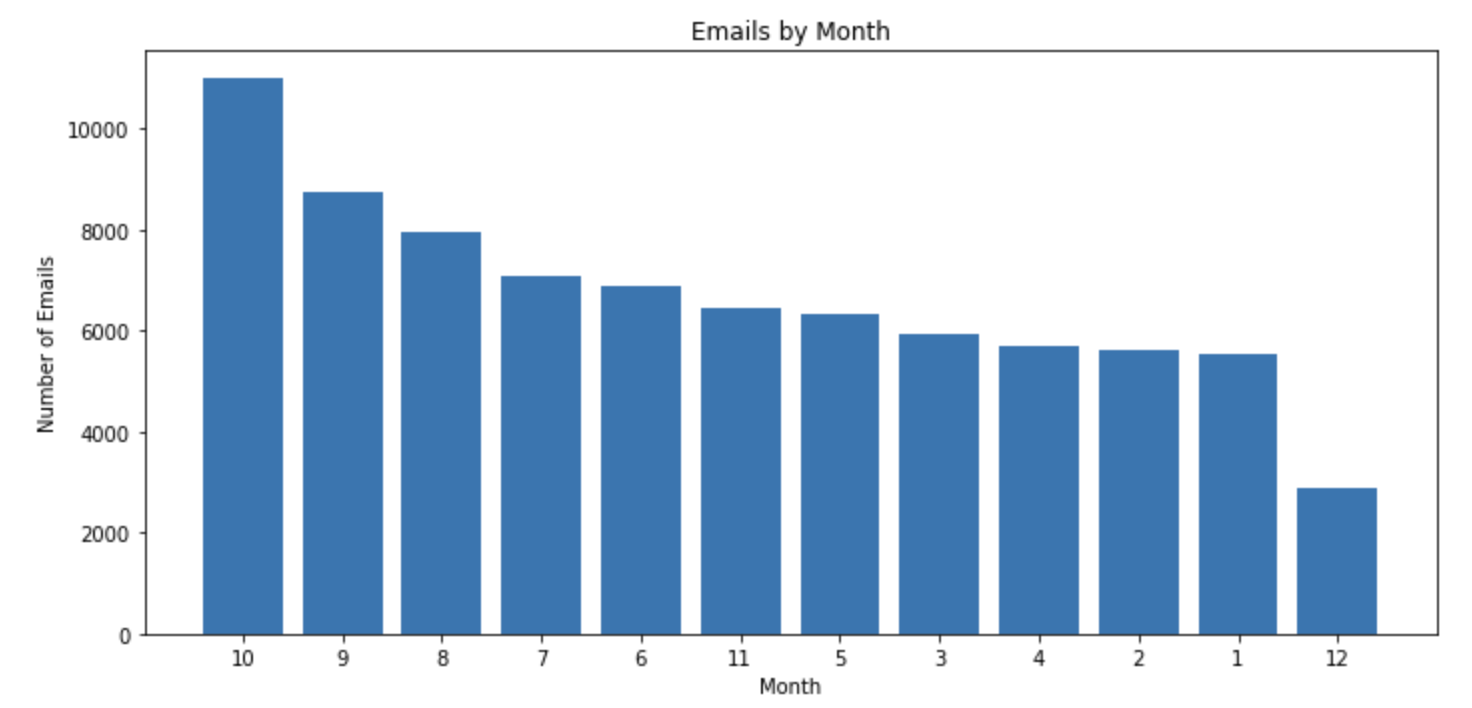


1. Top 5 podcasts with most episodes



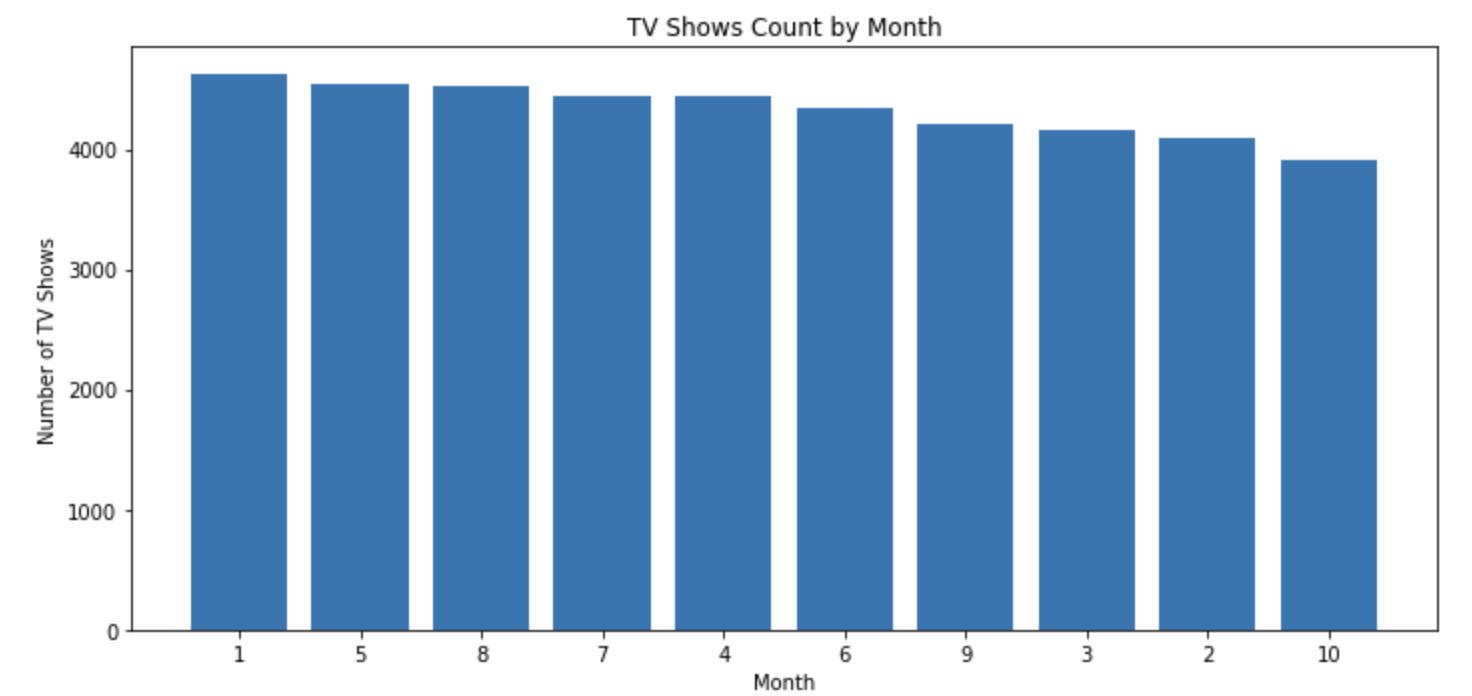
1. Data comparison by month for all the sources

Email:

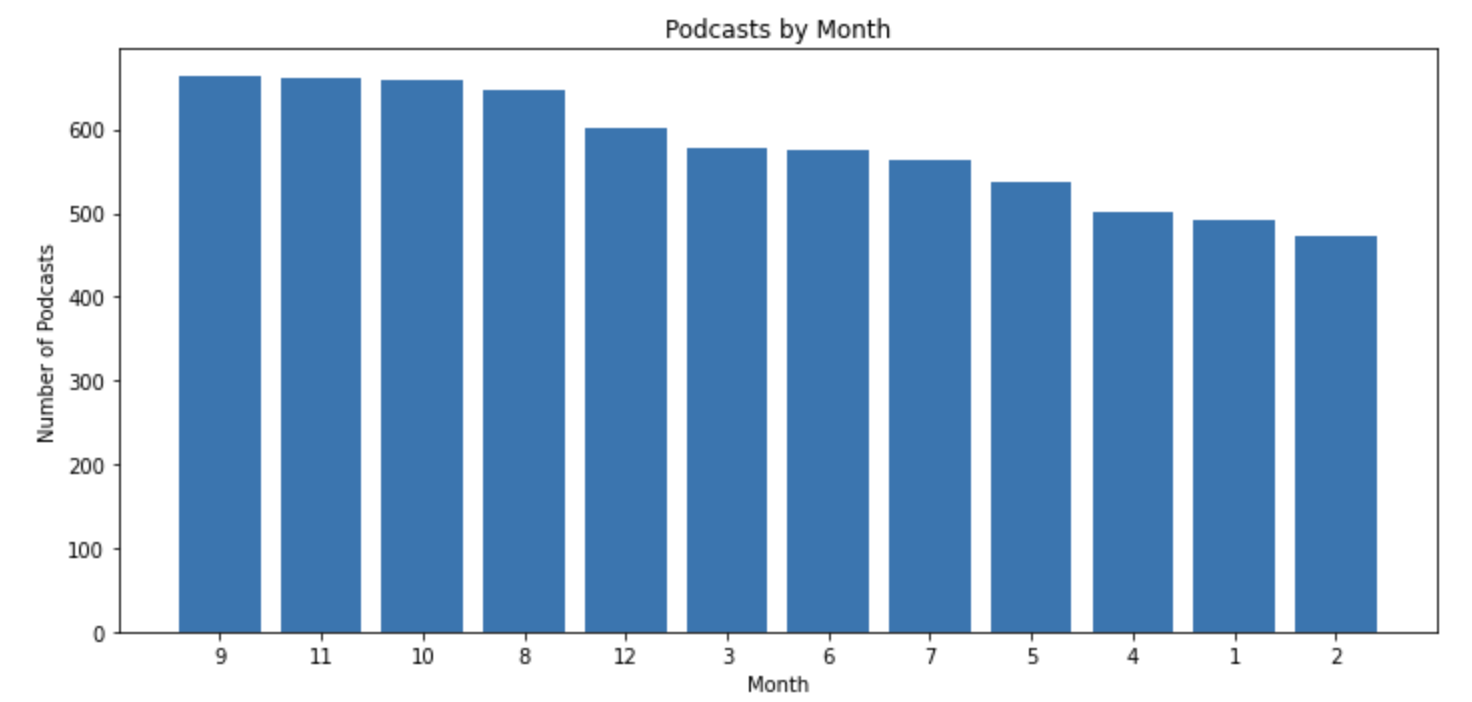


TV: Missing for the month November and December

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



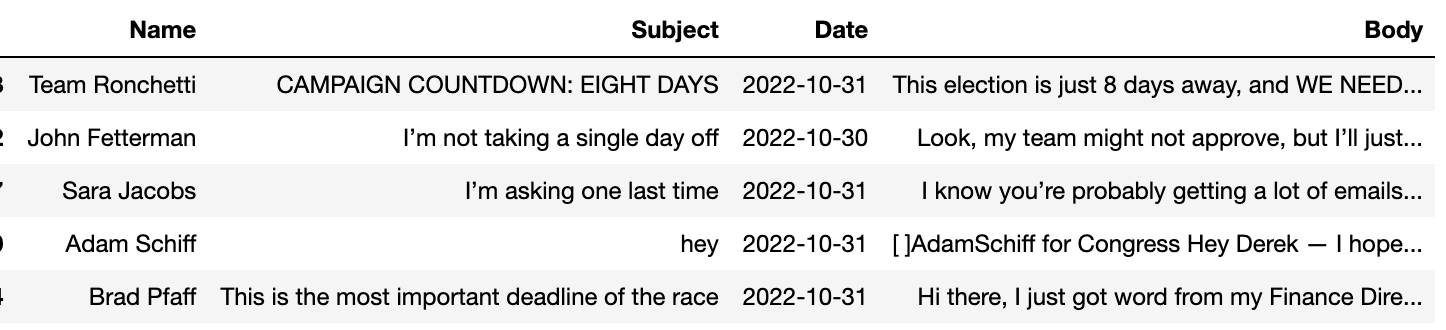
Podcasts:



1. **Standardize the data**

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

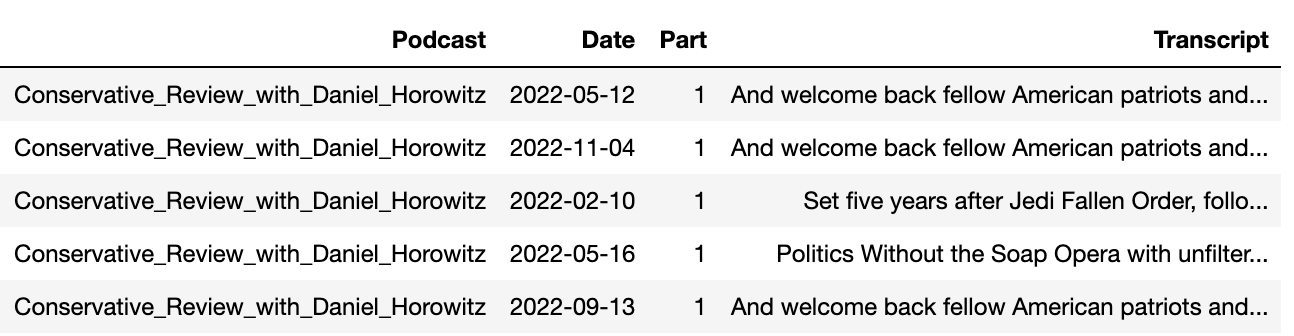
1. Email:



1. TV:



1. Podcast:



1. **Perform Pre-processing**

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 para)
3. Convert the data into passages of maximum length 100 words (only complete sentences)

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

1. **Get Embeddings for the data**

* Obtain embeddings for email and podcast data from MPNet LLM - Done
* Tv embeddings - pending (due to Amarel cluster issue)

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

* 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.

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

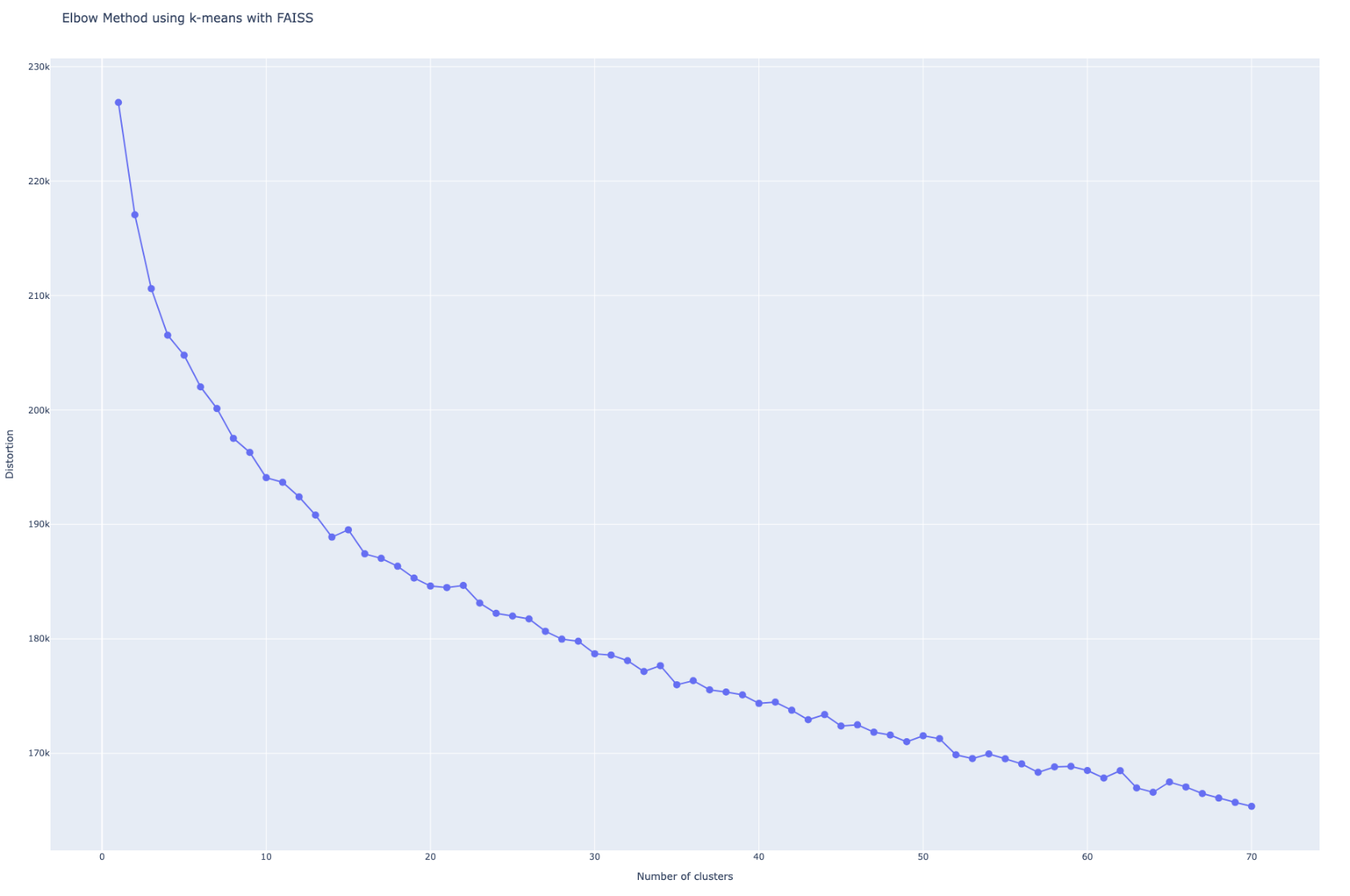
1. **Amarel Cluster issue**

* 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**

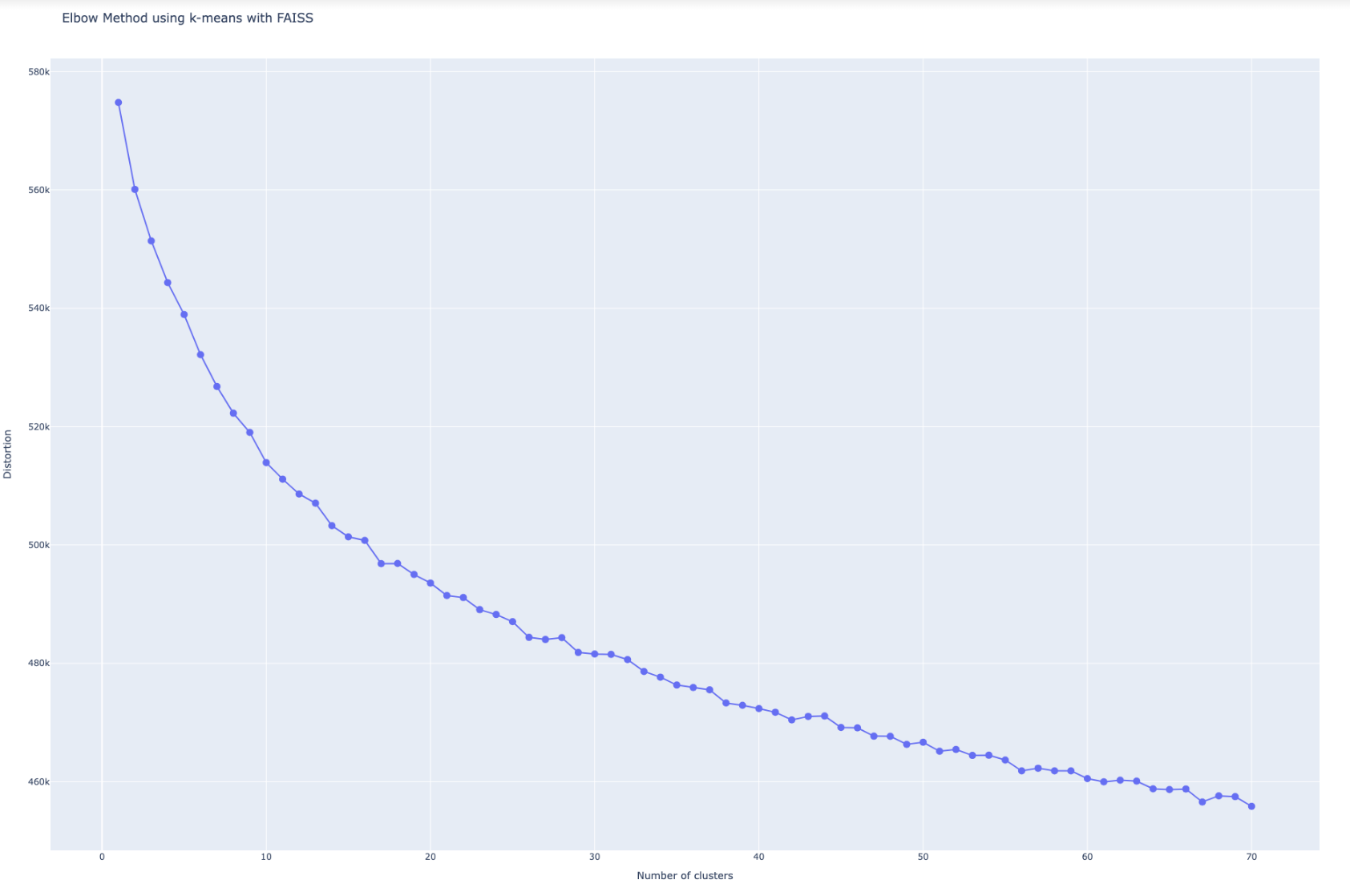
* Obtain the ideas number of cluster from elbow method
* Use below resources to improve the speed of K-Means clustering:

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

1. <https://towardsdatascience.com/how-to-speed-up-your-k-means-clustering-by-up-to-10x-over-scikit-learn-5aec980ebb72>
2. <https://www.blog.dailydoseofds.com/p/make-sklearn-kmeans-20x-times-faster#:~:text=To%20speed%2Dup%20KMeans%2C%20use,makes%20performing%20clustering%20extremely%20efficient>.
3. <https://github.com/facebookresearch/faiss>

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

1. **Try Link Transformer**

* The code is in place to do it, 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 for running it for TV data.

1. **Keyword Analysis**
2. Implement code for extracting unigrams and bigrams for TV data channel wise. - Done
3. Extract top antipathy words using sentiment analysis LLM and plot the comparisons for 3 TV channels - Done
4. Perform context-word analysis to identify before and after words for a particular keyword. - Done

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

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

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 certain keywords - Done
2. Top 10 Podcasts using certain keywords - Done
3. Context words analysis for TV data - Done
4. Context words analysis for Podcast data - Done

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

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

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

1. **Similarities - Done**
2. Get Jaccard Similarities for the clusters to compare the data sources.
3. Compare the clusters with >0.4 similarity scores

1. **Link Transformer Similarities**

Run the code to get the combined dataframe to compare and verify the cosine similarities. - Done for one scenario (podcast- email)

1. **Keyword Analysis**
2. Top 10 users using certain keywords - Done
3. Top 10 TV Shows using certain keywords - Done
4. Context words analysis for Email data - Done
5. General statistics, time series plots for all the data sources
6. POS tagging to identify the subjects, objects and causes for all the data sources
7. **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.

**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>

srun --cpus-per-task=16 --mem=16384 --time=600 /projects/community/anaconda/2022.10/bd387/bin/python clustering-tv.py

cp /home/rm1667/Documents/RA2/clustering-tv.py /scratch/rm1667/rm1667-outputs/