ISyE 6740 – Spring 2021
Final Report

Team Member Names: Daniel Schauder (dschauder3 – 902362614)

Project Title: Classifying and Clustering US TV News Using the Global Database of Events, Language, and Tone

Problem Statement

While the ideal of impartiality has long been regarded as an integral principle of the news media, in the United States, most major TV news outlets are incentivized primarily to produce content that drives viewership (and thus increases advertising sales). These two aims appear to be at odds, and many mainstream news stations are known to impart a political or sensational slant to their coverage.

This analysis employs automated machine learning methods to categorize, quantify, and explore the similarities and differences between major television news stations broadcasting in the US based on English transcriptions of their broadcasts. Using transcription data on 7 of the largest American TV news stations from the Global Database of Events, Language, and Tone, several analytical techniques are explored, including Naïve Bayes Classification, Latent Semantic Analysis, and Clustering via Gaussian Mixture Modeling. Ultimately, this paper demonstrates that machine learning methods can accurately distinguish between news stations and predict the broadcast source from anonymized transcript data, quantify the linguistic similarity between news stations, and automatically extract key "headlines" or topics in an unsupervised context.

Data Source

All data for this analysis are drawn from the Global Database of Events, Language, and Tone (referred to henceforth as "GDELT") [1]. This publicly available database "monitors the world's broadcast, print, and web news from nearly every corner of every country in over 100 languages and identifies the people, locations, organizations, themes, sources, emotions, counts, quotes, images, and events driving our global society every second of every day, creating a free open platform for computing on the entire world" [2].

Specifically, this analysis makes use of the "Television News Ngram 2.0 Dataset" [3]. The dataset is generated through an automated process which monitors the Internet Archive's Television News Archive, which provides the full-text "closed caption transcriptions from US TV News shows" [4]. The GDELT Project parses these transcriptions into a set of n-grams (unigrams, bigrams, trigrams, 4-grams, and 5-grams) with associated frequencies at 10-minute intervals for the following news stations: ABC, Al Jazeera, BBC News, Bloomberg, CBS, CNBC, CNN, CSPAN, CSPAN2, CSPAN3, Deutsche Welle, FOX, Fox Business, Fox News, LinkTV, MyNetworkTV, NBC, MSNBC, PBS, Russia Today, Telemundo, and Univision.

A small sample of the unigram data is shown here:

Row	TIMESTAMP	DATE	STATION	HOUR	NGRAM	COUNT	SHOW
3601	2021-02-25 18:10:00 UTC	20210225	DW	1810	have	3	Shift
3602	2021-02-25 18:10:00 UTC	20210225	DW	1810	on	7	Shift
3603	2021-02-25 18:10:00 UTC	20210225	DW	1810	wanted	1	Shift
3604	2021-02-25 18:10:00 UTC	20210225	DW	1810	when	2	Shift
3605	2021-02-25 18:10:00 UTC	20210225	DW	1810	facebook	2	Shift
3606	2021-02-25 18:10:00 UTC	20210225	DW	1810	will	1	Shift

Well over a terabyte of data is available, with n-grams going back to 2009 for most of the available stations. Over a million records are generated each day across all stations. The dataset is published in an easily accessible format hosted by Google BigQuery.

Methodology

Data Treatment and Preprocessing

The results discussed in this work were derived from one week of unigram data covering the period from 4/8/2021 to 4/14/2021 (inclusive), but the same methods described here were applied to multiple other weeks in 2021 and all yielded similar results. While the source data included 22 stations, only 7 stations were included in this analysis. The remaining 15 stations were excluded either because they did not broadcast on a 24-hour schedule, they were redundant of other stations (e.g. CSPAN2, CSPAN3), they were local affiliate stations and skewed towards local topics, and/or their primary focus was business news, featuring a lexicon oriented towards updates on the financial markets. The 7 stations ultimately chosen for analysis were: Al Jazeera, BBC News, CNN, CSPAN, Deutsche Welle, FOX News, and MSNBC.

The GDELT project converts all words to lowercase by default and strips punctuation. The data were further treated by removing records with missing unigram values, duplicate records, and unigrams which only consisted of "special" non-alphanumeric characters. The data was anonymized by removing unigrams containing the specific station names and removing any unigrams which appeared on only one station. Additionally, "stop words" (common words which contribute little meaning) were removed. The list of stop words was derived from the Natural Language Toolkit for Python and supplemented with a manually-compiled list of common words in the news context which were creating noise (e.g. "people", "think", etc) [5].

Bag-of-Words

To prepare the data for further analysis, the unigram data was converted into a normalized matrix using the pandas package in Python [6]. Each row of the matrix corresponded with a 10-minute interval on a particular station and was considered to be a "sample". Each column of the matrix corresponded with a specific unigram or "feature", and the values in the matrix denoted the unigram's frequency (number of times the unigram was said) within a specific 10-minute interval on a specific station. After conversion, the resulting matrix consisted of 6,965 samples, each of which containing 39,780 unigram features. Since this high-dimensional matrix was

extremely sparse, it was then converted to scipy's csr_matrix format for more efficient processing [7].

Next, Term-Frequency Inverse-Document-Frequency ("tf-idf") weights were applied to the raw frequency values in the matrix using scikit-learn's TfidfTransformer package [8]. The effect of this transformation is to scale down the values of terms which are highly common across the corpus while inflating the values of terms which only occur on a small subset of samples, and thus may be more useful in statistical analysis [9]. Smoothing was applied in generating the tf-idf weights to ensure that unigrams that appeared in every sample were not completely removed from consideration.

This normalization process is common in the semantic analysis literature and is known as "bag-of-words", since all information related to the ordering of the words is lost and unigrams are assumed to be independent. While this assumption is not actually correct, this normalization process opens the door to a host of useful analytical techniques rooted in linear algebra which can be implemented to derive interesting insights.

Multinomial Naïve Bayes Classification

A Multinomial Naïve Bayes classifier was trained using the MultinomialNB package of scikit-learn on 80% of the samples, while 20% of the samples were reserved for evaluating classification performance [10]. Posterior log-probabilities were calculated for each unigram for each station, and the top 10 highest posterior log-probabilities for each station are shown in the following figure. These probabilities provide a novel perspective in themselves in characterizing the language of each station.

Al Jazeera	
Unigram	Posterior Probability (Log)
iran	-7.424975
1st	-7.597515
nuclear	-7.660537
police	-7.66873
says	-7.679791
president	-7.716069
government	-7.717357
u.s.	-7.728808
israel	-7.74055
vaccine	-7.843299

BBC News	
Unigram	Posterior Probability (Log)
duke	-7.089097
queen	-7.286399
prince	-7.323617
royal	-7.37596
philip	-7.502319
uk	-7.511019
edinburgh	-7.621932
england	-7.635947
showers	-7.699171
vaccine	-7.724762

CNN	
Unigram	Posterior Probability (Log)
police	-7.309221
reporter	-7.342451
floyd	-7.501634
gaetz	-7.591702
vaccine	-7.671978
officer	-7.7292
death	-7.831836
george	-7.833347
mr	-7.936646
case	-7.941268

CSPAN	
Unigram	Posterior Probability (Log)
host	-7.677615
thank	-7.786644
work	-7.910037
children	-7.957598
caller	-7.975793
health	-7.981065
president	-7.987498
question	-7.991006
vaccine	-7.994989
important	-8.03969

Deutsche Welle	
Unigram	Posterior Probability (Log)
1st	-7.481738
world	-7.765946
germany	-7.815178
pandemic	-7.948816
year	-7.95307
years	-7.972568
world	-8.003917
government	-8.015184
still	-8.016784
u.s.	-8.0204

FOX News	
Unigram	Posterior Probability (Log)
border	-7.177015
biden	-7.345561
police	-7.524162
president	-7.527588
greg	-7.618589
infrastructure	-7.866845
administration	-7.930169
joe	-8.025566
pete	-8.046214
dana	-8.049386

MSNBC	
Unigram	Posterior Probability (Log)
police	-7.252379
gaetz	-7.610934
floyd	-7.65317
trump	-7.789967
officer	-7.824591
george	-7.829466
mr	-7.836496
president	-7.883809
defense	-7.903242
case	-7.931931

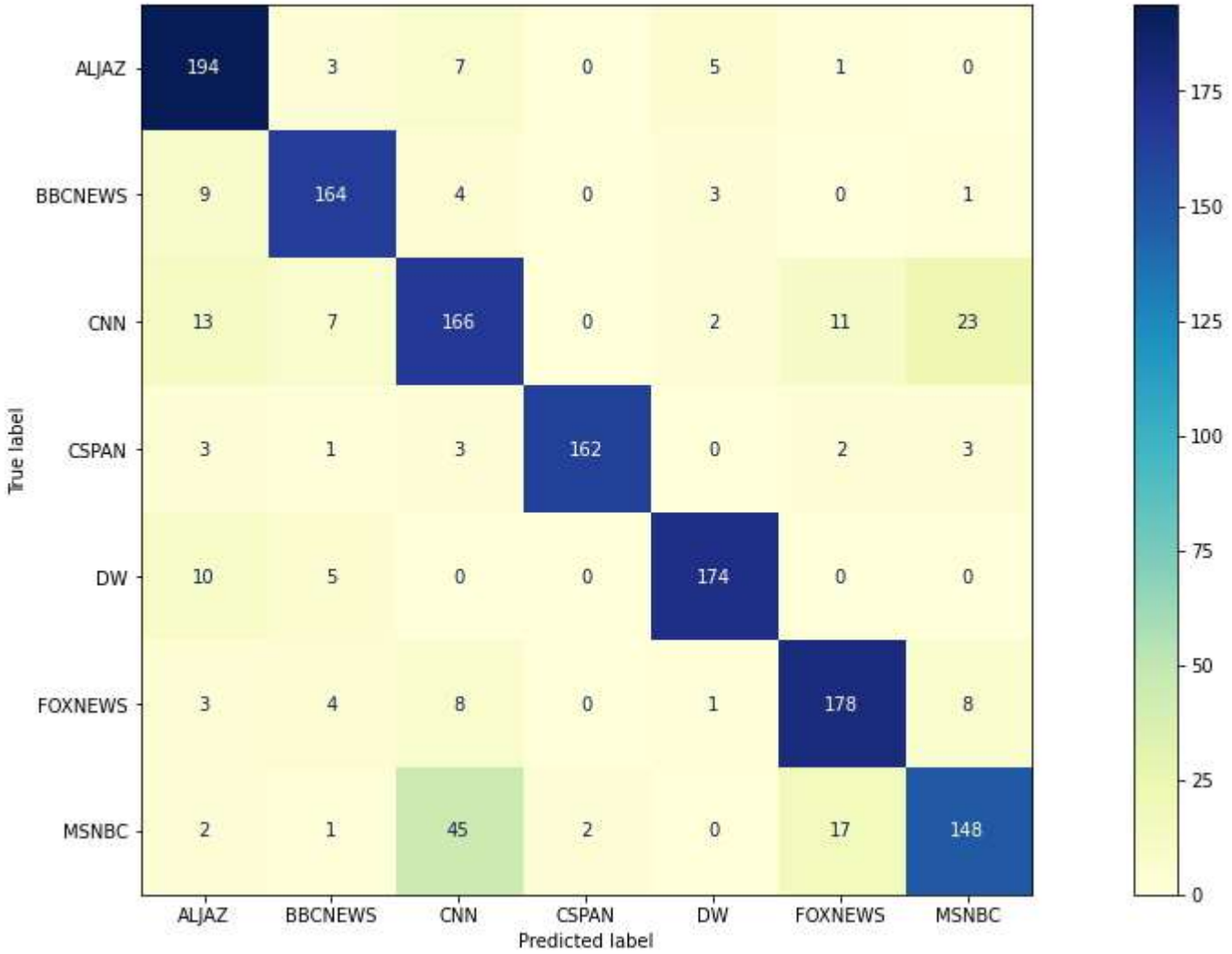
At a glance, Al Jazeera appears to have a greater focus on the political situation in the Middle East, BBC's top terms revolve around the passing of Prince Philip, CNN and MSNBC focus on George Floyd and the Matt Gaetz scandal, FOX News appears to focus on border regulations and President Biden, while Deutsche Welle and CSPAN unigrams revolve around the global pandemic and/or vaccines.

When evaluated using the 20% of the data reserved for testing, the Multinomial Naïve Bayes classifier achieved 85% accuracy overall. Detailed classification performance metrics are provided in the following figures.

	ALJAZ	BBCNEWS	CNN	CSPAN	DW	FOXNEWS	MSNBC
Precision	0.83	0.89	0.71	0.99	0.94	0.85	0.81
Recall	0.92	0.91	0.75	0.93	0.92	0.88	0.69
F1	0.87	0.9	0.73	0.96	0.93	0.87	0.74

Accuracy	Macro Avg	Weighted Avg
	0.86	0.85
	0.86	0.85
0.85	0.86	0.85

Multinomial Naïve Bayes Confusion Matrix



Other classification models were considered including Support Vector Machine, Logistic Regression, and/or a Multilayer Perceptron Neural Network. While it is possible other models may achieve greater accuracy than the Multinomial Naïve Bayes model, they may do so at the expense of explainability. The posterior probabilities associated with specific unigram/station combinations derived from the Naïve Bayes approach were of direct intrinsic interest, so the Naïve Bayes classifier was deemed to be sufficient for this portion of the analysis.

Comparing Similarity of Stations

A second goal of this work was to find a quantitative method of comparing stations with one another to uncover which stations were most similar based on the language used in their broadcasts. Since the focus of this and all the remaining analyses was no longer on predicting a response on novel data, the full dataset was used (covering 4/8/2021 through 4/14/2021). First, the frequency matrix was aggregated to the station level such that each station was described by a 39,780-dimensional vector of total unigram frequencies over the entire period. Next, scikit-learn's cosine_similarity function was used to produce a 7x7 pairwise similarity matrix [10]. In general, the cosine similarity score, also known as the *Pearson Correlation Score* or the *normalized dot product*, describes the extent to which two vectors point in the same direction. This similarity metric was chosen for its computational simplicity and ubiquity in the literature for comparing documents or corpuses. The station similarity matrix is depicted in the following figure. Since the matrix is symmetric and the diagonal entries are all 1, only the lower triangular portion of the matrix is shown.

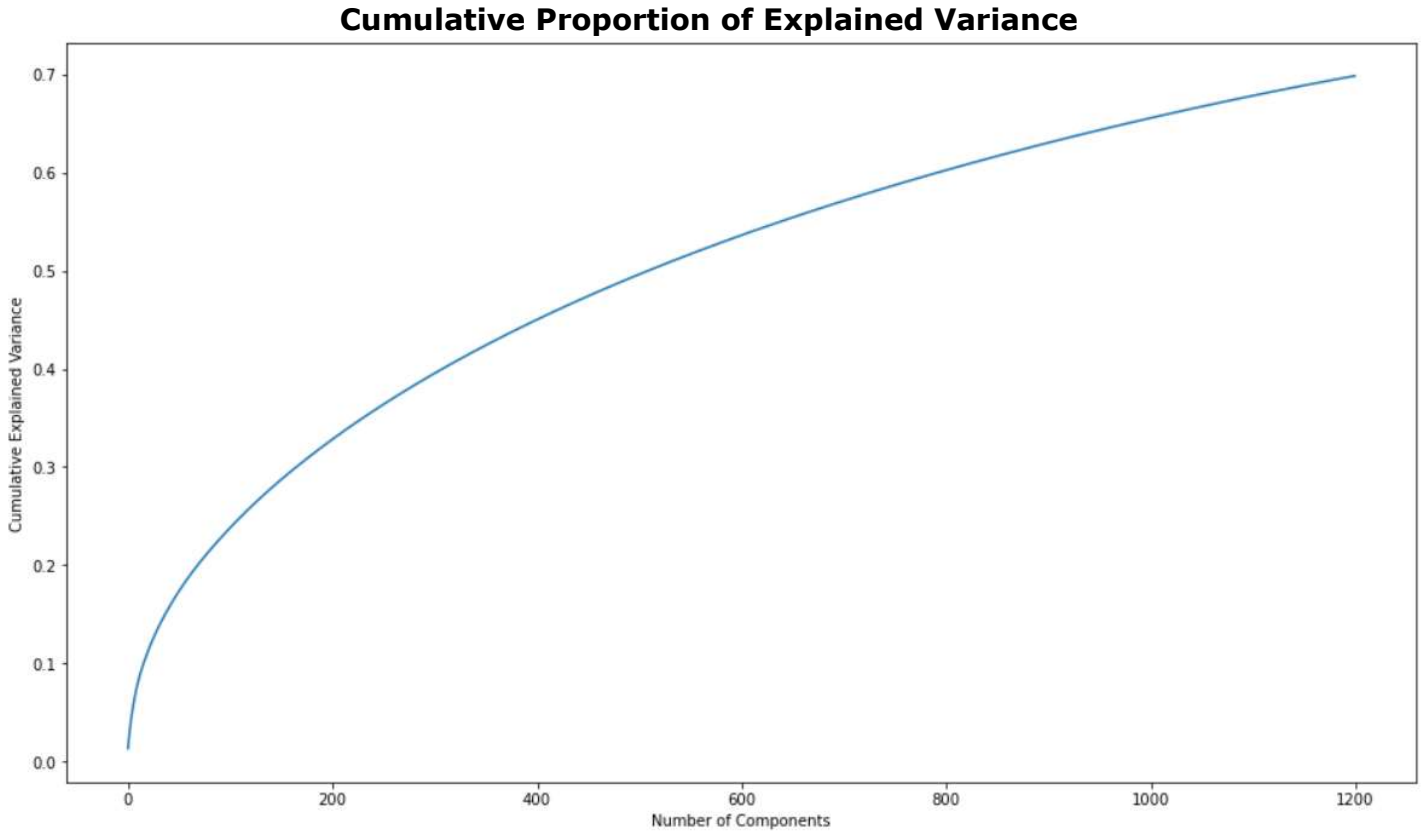
Station Similarity Matrix

MSNBC and CNN shared the highest similarity score (0.96) while CSPAN and BBC News appeared to be the most divergent with a score of 0.68. Somewhat surprisingly, MSNBC and FOX News shared a relatively high similarity score of 0.86 despite their reputations for portraying the news from opposing political viewpoints.

Latent Semantic Analysis for Dimensionality Reduction

The final goal of this project was to use unsupervised clustering methods to uncover the main headlines or topics in a given time period and to approximate the share of coverage for each topic by each station. While the frequency matrices used in the preceding sections were sparse and high-dimensional, clustering methods tend to work best when the number of samples far outweigh the number of dimensions. To overcome this challenge, Latent Semantic Analysis (LSA), a variant of Singular Value Decomposition (SVD), was used for dimensionality reduction.

LSA uses SVD to find a low-rank approximation of the original high-dimensional matrix while preserving as much of the variation from the original matrix as possible [11]. Scikit-learn's TruncatedSVD function was used to transform the 39,780-dimensional matrix into a truncated 1,200-dimensional matrix [12]. This package is formulated specifically to efficiently process sparse matrices where pre-centering of the data is infeasible due to memory constraints. The choice of how many components to use was not straightforward – ultimately 1,200 components were chosen as they cumulatively preserved ~70% of the variance from the original dataset as shown in the following figure.



Topic Discovery via Gaussian Mixture Model and Expectation Maximization
After the dataset was projected to a 1,200-dimensional space, a Gaussian Mixture Model (GMM) was fit via Expectation Maximization using scikit-learn's GaussianMixture package [13]. The motivation for this exercise was to use unsupervised clustering methods to detect the most prevalent topics being covered in the news and to show the allocation of coverage from each station to each topic.

One shortcoming of this approach is that the "optimal" number of clusters is not automatically detected by the GMM algorithm – it must be provided as an input parameter. Tuning this parameter is not straightforward as quality-of-fit metrics will generally improve monotonically as more clusters are added, and no ground truth was available in this case for establishing metrics like purity, homogeneity, or completeness [14]. Different numbers of clusters were tried and evaluated somewhat subjectively based on how well the detected topics aligned with real recent headlines, how balanced the resulting clusters were, and how well the number of topics generalized over several different weeks. Ultimately, 10 clusters were chosen, which aligned somewhat intuitively with the number of major headlines one may expect to be covered over the course of a week.

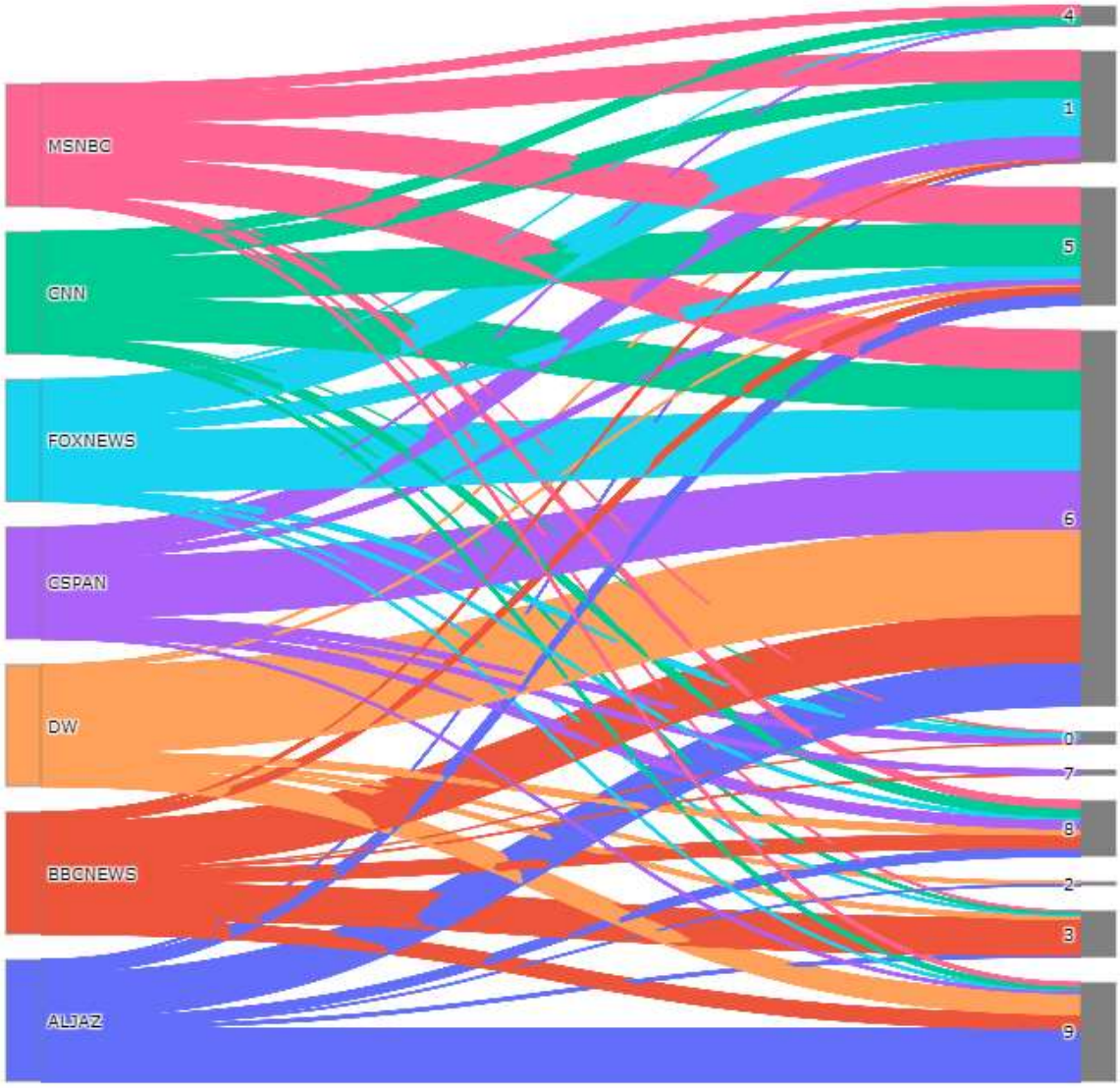
After the GMM model was fit, the 10 resultant mean vectors were transformed back into the original 39,780-dimensional space. The unigram features with the greatest magnitude in each of the mean vectors were the most "important" unigrams in defining the topic. Below are the top 10 unigrams for each of the 10 topics detected for the period from 4/8/2021 to 4/14/2021 (inclusive).

```
Cluster 0: court supreme president justice justices law biden packing property judges
Cluster 1: president biden gun republicans trump infrastructure border republican party house
Cluster 2: art digital market european president christie's e.u. crypto 1st artist
Cluster 3: prince queen philip royal duke family life edinburgh years world
Cluster 4: gaetz matt greenberg sex congressman investigation trump women he's trafficking
Cluster 5: police floyd officer george officers mr death case defense heart
Cluster 6: years year thank world still last take work first country
Cluster 7: speaker mr tempore gentleman pro house pursuant h.r. bill designated
Cluster 8: vaccine vaccines johnson health cases blood vaccinated doses million vaccination
Cluster 9: president government u.s. iran country military afghanistan says nuclear states
```

Interestingly, 8 out of 10 of the topics detected clearly align with distinct major headlines. Cluster 0 refers to proposed changes to the structure of the US Supreme Court, Cluster 1 appears to focus on the US southern border and/or gun control, Cluster 2 refers to a notable art sale making use of cryptographic technology, Cluster 3 refers to the death of Prince Philip, Cluster 4 refers to a sex scandal involving US Congressman Matt Gaetz, Cluster 5 focuses on George Floyd, Cluster 8 focuses on COVID-19 vaccinations, while Cluster 9 involves the US withdrawal from Afghanistan and/or nuclear talks with Iran. Of the remaining clusters, Cluster 6 appears to mostly involve generic language not clearly associated with a single topic, while Cluster 7 consists mostly of the formal language you may expect to hear from Congress on CSPAN.

The below Sankey diagram is useful in visualizing how the samples from each of the 7 stations were associated with each of the 10 topics. Cluster 6 was the largest of the clusters and consisted mostly of generic language not easily attributable to a single headline or topic. Regardless of the number of clusters chosen in tuning the algorithm, the largest cluster always corresponded with generic unigrams similar to Cluster 6. This may be due to the fact that many of the 10-minute samples consist of a mix of several different topics which cannot be easily characterized by a single headline. One potential avenue for further exploration would be to aggregate the samples to different increments such as 30 minutes or 1 hour to observe whether more clear distinctions can be drawn at this higher level of granularity (i.e. perhaps the first 10 minutes of all shows include a broad overview of all the major headlines, while the remainder of the show provides a deeper dive into a few specific topics).

News Station Coverage of 10 Detected Topics



Evaluation and Final Results

This project sought to explore the hidden structures embedded in the language of 7 television news sources through the application of statistical analysis and machine learning methods on transcription data from the Global Database of Events, Language, and Tone. By training a Multinomial Naïve Bayes Classifier to categorize news sources with 85% accuracy, this work demonstrated that each of the 7 stations studied has a characteristic linguistic profile which differentiates it from the others, which is interesting considering that each station purports to present a fair and unbiased reporting of facts. The relationships between the 7 stations were explored to provide a quantitative comparison of similarities using Cosine Similarity scores, revealing a surprisingly high level of similarity on stations generally considered to be on opposite ends of the political spectrum. Latent Semantic Analysis was performed to reduce the dimensionality of the data, and finally, an unsupervised Gaussian Mixture Model was built and used to automatically detect the top headlines dominating the news cycle.

The techniques described here could be useful to researchers seeking to document the evolution of the news and to better understand the relationship between language, culture, and historical events. These methods could be easily extended to detect, categorize, and trace the top headlines at arbitrary points in time. While satisfactory performance was achieved to demonstrate the feasibility of classification and clustering tasks on this dataset, alternative algorithms could be explored and may attain superior performance.

References

[1] K. Leetaru and P. A. Schrodt, "GDELT: Global Data on Events, Location and Tone, 1979-2012," in *International Studies Association*, San Francisco, 2013.

[2] K. Leetaru, "Global Database of Events, Language, and Tone," [Online]. Available: https://www.gdeltproject.org/.

[3] K. Leetaru, "Announcing The Television News Ngram 2.0 Dataset," [Online]. Available: https://blog.gdeltproject.org/announcing-the-television-news-ngram-2-0-dataset/.

[4] "The Internet Archive," [Online]. Available: https://archive.org/about/.

[5] E. . Klein, E. . Loper and S. . Bird, Natural Language Processing with Python, ed., vol. , , : O'Reilly Media Inc, 2009, p. .

[6] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010.

[7] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. van der Walt, M. Brett, J. Wilson, J. K. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods,* vol. 17, pp. 261-272, 2020.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[9] e. a. Pedregosa, "scikit-learn TfidfTransformer Documentation," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html.

[10] e. a. Pedregosa, "scikit-learn Cosine Similarity Documentation," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html.

[11] T. Landauer, P. Foltz and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes,* vol. 25, pp. 259-284, 1998.

[12] e. a. Pedregosa, "scikit-learn TruncatedSVD Documentation," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.

[13] e. a. Pedregosa, "scikit-learn GaussianMixture Documentation," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html.

[14] C. D. Manning, P. . Raghavan and H. . Schütze, Introduction to Information Retrieval, ed., vol. , , : Cambridge University Press, 2008, p. .

[15] P. Prettenhofer and L. Buitinck, "Clustering text documents using k-means," [Online]. Available: https://scikit-learn.org/0.15/auto_examples/text/document_clustering.html.

[16] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software,* vol. 6, no. 60, p. 3021, 2021.

[17] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering,* vol. 9, no. 3, pp. 90-95, 2007.

[18] P. T. Inc., "Collaborative data science," Plotly Technologies Inc., Montreal, QC, 2015.

[19] C. R. Harris, J. K. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Rio, M. Wiebe, P. Peterson, P. Gerard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, "Array programming with NumPy," *Nature,* vol. 585, no. 7825, pp. 357-362, 2020.

Appendix

All of the code used for this analysis can be found at the following GitHub URL: https://github.com/danschauder/GDELT_News_Analysis