What Does Fox Say?

Alex Rosenblum, Chris Figy, Peter Jachim

Contents

What Does Fox Say?

Alex Rosenblum, Chris Figy, Peter Jachim

Executive Summary

Text analytics or natural language processing NLP is the analysis of a corpus or a collection of documents. Our goal is to convert a corpus of documents into a matrix of numbers that can allow statistical analysis and modeling to help identify grouping or subjects of articles. We also look at a sentiment toolkit to help differentiate the documents.

Our corpus is over two years' worth of FOX newsletters/emails where a python script creates a CSV file for analysis below. The file or dataset does not have categorical variables, but a natural language processor toolkit NLP-TK is used to apply a sentiment calculation named VADER ranging from 1 to -1 where a positive article is more positive on the VADER index. We later also turn this into a likert score.

Data

The data had initially been collected using a designated email inbox. We downloaded the emails, and parsed them into a spreadsheet. Once we had the data in a spreadsheet, we used a variety of techniques to clean the text itself to remove the HTML and embedded JavaScript from the newsletters.

Once we had just the text from the emails, we used something called a lemmatizer to combine different forms of the same word, like "running", "ran" and "runs" with "run".

Because we were working with text data, the first step in our analysis is to convert text into numbers. We used two different types of strategies. For the first strategy, we vectorized the data, which means we created encoded the text into numbers. To do that, first, we used tokenizers. Tokenizers divide the text into tokens of a given size. A sentence or a word are two examples of a token size. For this research we will be dealing with tokenization at the word level. This is our starting point.

In NLP we commonly refer to the word tokenization as a "bag of words" or BOW for short. We have no context of order and we have lost a lot of information in the ordering or context of the words. This is OK as we will see this is still powerful. We also tried using a TF-IDF vectorizer that weighs the importance of each word based on how much each word appears in a newsletter with how many newsletters contain each word.

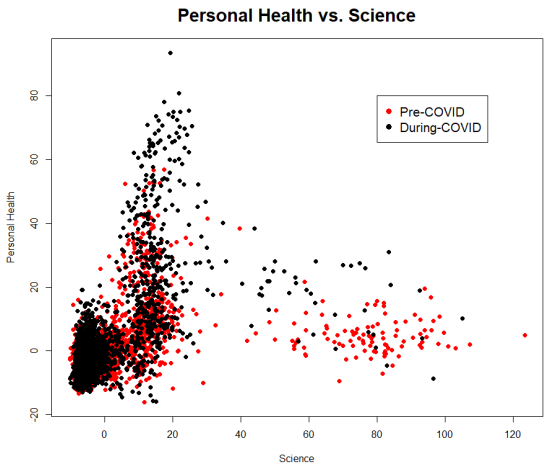
For a second strategy, we used the VADER sentiment analyzer (Hutto & Gilbert, 2015), which takes a piece of text and estimates how positive, negative, and neutral it is based on the words in the piece of text.

The wordcloud below shows the term frequency count from some of the most negative newsletters:

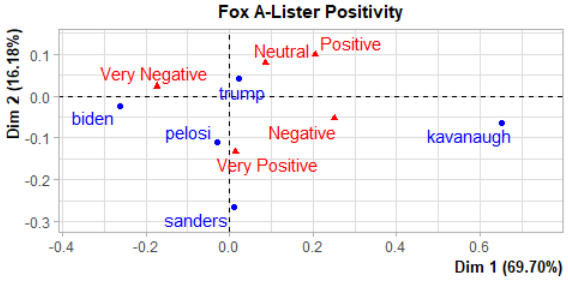
For a more thorough, in-depth discussion of the data cleaning, see Appendix A.

Analysis

One of the primary goals of this analysis was to model the change in messaging regarding various topics enacted by Fox News after the COVID-19 pandemic began in the United States. Consider the scatter plot presented here, which was generated by applying the analytical technique Principal Component Analysis to the tokenized text described above. This technique results in several groupings of terms used by Fox News separated by subject. The points colored red represent newsletters sent before the pandemic began, and the black points sent after. This plot shows what our analysis here has confirmed: Fox News has sharply curbed reporting surrounding science, while increasing coverage around lifestyle and health without using much science terminology. Other trends indicated by our analysis (not shown here) include US Politics coverage dropping considerably, while average sentiment in Fox's messaging has strayed from negative verbiage in favor of stronger positivity.



Another way that we continued our analysis of what Fox speaks positively or negatively about is through our use of a correspondence analysis. A correspondence analysis is a way to compare two distinct categorical variables. We used it to help look for how positive emails that mention a specific person are. In the graph below, the people (in Blue) are closest to the positivity that Fox most frequently uses to describe them.



Conclusions

Through our analysis, we accomplished several things. We were able to apply several different text analysis algorithms to quantify sentiment (positivity vs. negativity) in Fox newsletters, and identify which people and topics Fox writes about negatively or positively, and to what degree. We were also able to apply analytical techniques to group certain newsletters together based on subject, and used this to discover that Fox's coverage regarding both science and US politics has been largely reduced since the start of the COVID-19 pandemic in favor of reporting surrounding the subjects of health and lifestyle. Additionally, Fox has adopted a generally more positive and less negative average sentiment since the pandemic began.

Technical Summary

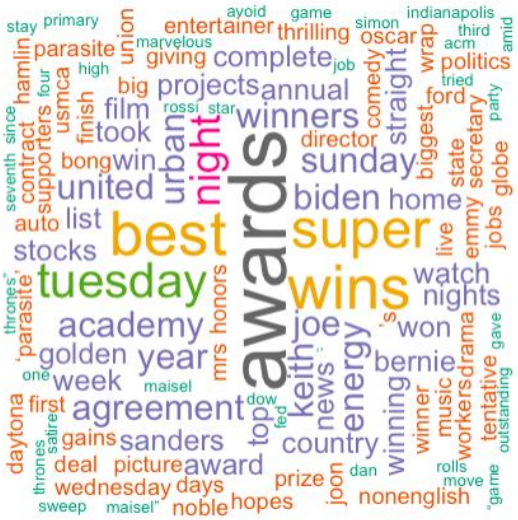
Exploratory Data Analysis

The Vader index is part of the NLP-TK toolkit providing a composite sentiment scoring of positive one to negative one scaling for positivity / neutrality / negativity. This metric can be used for our exploratory analysis. We include two word clouds to this exploration to provide a nice visual into the corpus.

A word cloud chart sizes the words in the chart based on their count in the sample. The larger the word on the chart, the higher count in the emails. We will take a sample of emails with the highest negative and positive Vader scores to produce the word clouds.

The first smaller sample produced the below two word-clouds. Scaling up the sample size produced interesting results in the positive word cloud. Two major words of "President" and "Trump" are removed when looking at the smaller sample. Those words are appearing in every email in the small sample caused them to be penalized and removed from our analysis. The appendix will explore why this happens when we explain the TF-IDF calculation.

High Positivity Vader Word Cloud



A lot of the above is about "Winning" or "Win" once you look at the articles that created this graphic. Articles with sentences using the words win, award, prize, gains, best, plus others are found above. We see signs of grouping too, for example, we might have groups like Financial, Politics, & Entertainment.

High Negativity Vader Word Cloud

The negative word cloud above is clearly crime related. Overall, our sentiment index or VADER feature we added using NLP-TK seems to be working well at ranking the articles. The only issue maybe the word "coronavirus" appearing in the positive cloud when including a large number of articles. This seems like a miss-classification.

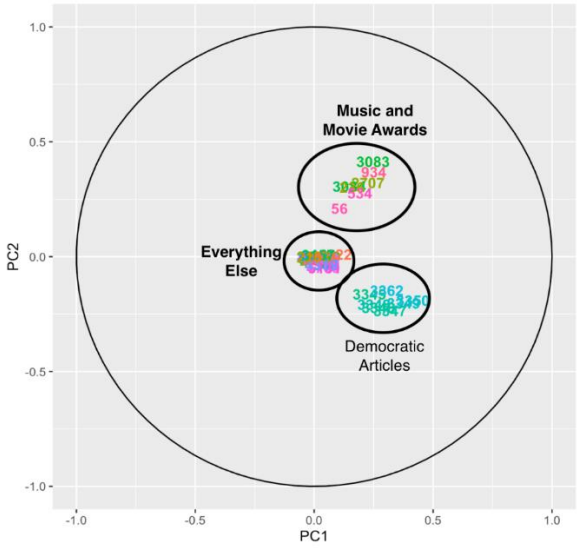
Principal Factor Analysis

Once we created the TF-IDF matrix during the data cleanup and preparation work, we can now use this matrix in various models. We begin with using the same two samples from the VADER exploratory analysis and word clouds above. The two datasets will be merged and used as a single dataset with two groups – negative and positive articles.

Running the first PCA splits the articles up nicely. Looking at the first two principal components we were able to split out two

major groups found in the corpus. All the articles not split out seemed to remain at the origin or (0,0) points in PCA1(x) and PCA2(y).

Articles from a "Democratic Articles" group has words like "Joe", "Bernie", "Biden," and "Sanders." The other group has words like "Awards", "Keith", "Urban", "Golden" and "Globe." We labeled as "Music and Movie Awards." Both groups moved positively in PC1 away from the origin. The "Democratic Articles" group had negative PCA2 values while the "Movie and Music Awards" had positive PC2 values.



Examining further with PC3 and PC4 we saw articles with sports theme in a positive PC4 and crime related articles in a negative PC4 direction. The two directions the articles moved away from the origin appear to be 90 degree from one another.

This was a positive outcome for PCA. We next wanted to see how this PCA would handle a larger corpus of articles. We performed this same analysis on 70% of the data – leaving the remaining 30% for some training later on.

Our results with 70% of the articles was able to pull out two groups similar to above. Everything is in the center and we have "Stock Market" emails as positive in both PC1 and PC2. The other group was about "Crime and Murder" in the positive PC1 and negative in PC2.

One major correction is needed after looking at the larger sample. Our "Democratic Articles" from the small group are really "Politics" in general and not so party specific. For example, running the word cloud on a larger set of emails produced similar results but the largest words in the cloud were "President" and "Trump."

If you read the section on TF-IDF you might understand why this happened. In our small sample size for the first exploration every article must have had "President Trump" in it and again because of the IDF part of TF-IDF we penalized President Trump as it was found in all articles/emails in the small sample.

Overall PCA showed positive signs of being able to group some articles.

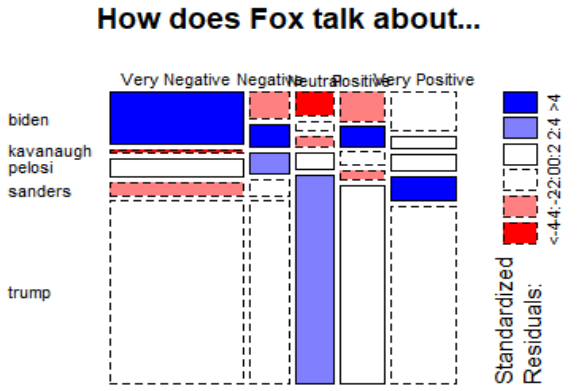
Correspondence Analysis

The next direction that we explored was a correspondence analysis of specific words in emails. To do this, we experimented with creating a new metric from the VADER Compound scores called the "Positivity" that shows how positive a text sample is. For the Positivity, I am considering anything less than -.75 as "Very Negative", any scores between -.75 and -.25 as "Negative", anything between -.25 and .25 as "Neutral", .25 to .75 is "Positive", and scores greater than .75 are "Very Positive". While these values are technically ordinal, we have found promising initial results using correspondence analysis.

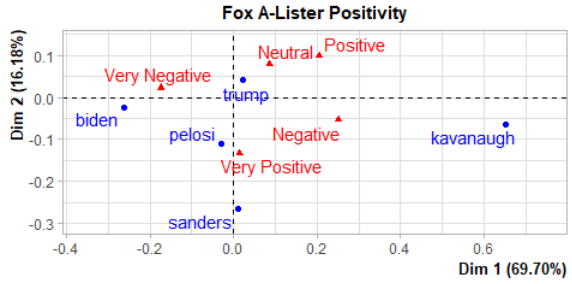
To start out with, we created lists of words to compare with the positivity. Initially we started with a list of politicians, but found that specific people dominated our mosaicplot, so we split them into a listers and b-listers based on how frequently they appear in Fox newsletters (out of curiosity, I check "Kardashian" and "Kanye", and found that they were both solidly B Listers).

A-Listers

The A-Listers I identified were (Joe) "Biden", (Brett) "Kavanaugh", (Nancy) "Pelosi", and (Bernie) "Sanders", and (Donald) "Trump". Of these, because he's President, Trump dominated this list.

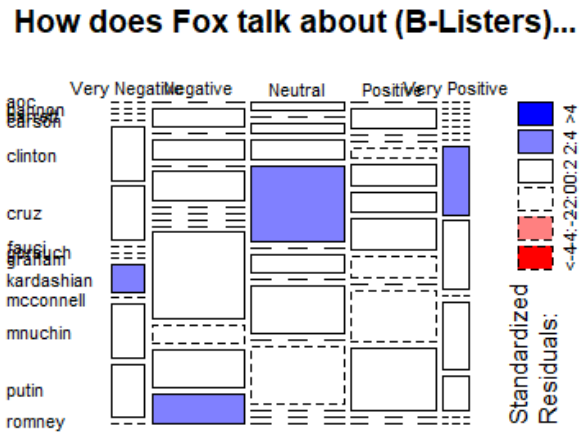


Notice how infrequently Fox mentioned Brett Kavanaugh in very negative terms, versus how frequently they speak negatively about Nancy Pelosi. This makes it seem like Fox news might use Nancy Pelosi as a scapegoat of sorts for barriers to conservative agendas.

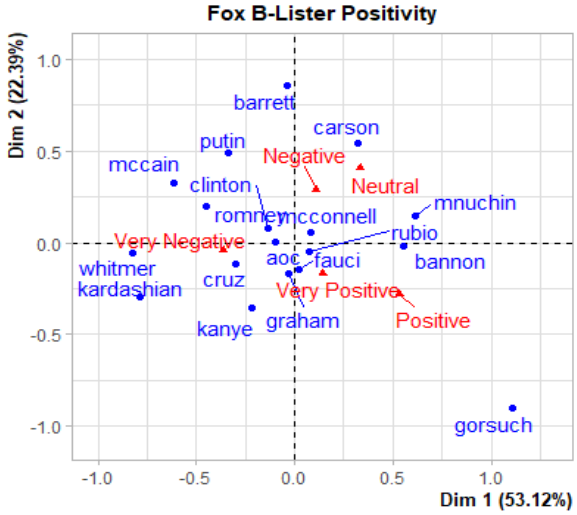


Interestingly, Sanders is generally spoken about very positively. Kavanaugh, while he is not generally spoken about using very negative terms, he is frequently described using negative terms, though these negative terms could refer to the negative terms used to describe the process of Kavanaugh's confirmation. Biden is also generally discussed using very negative terms, which seems to reaffirm Fox's position as a right-leaning news platform.

B-Listers



Together, the first two components of the correspondence analysis account for 71.6% of the variance in the data.



Notice how positively Fox reports on Neil Gorsuch (located in the bottom right) vs. how they speak about Amy Coney Barrett (located towards the top of the plot), who are both ostensibly Trump conservative picks for the Supreme Court.

Another interesting relationship is between Whitmer and Kardashian. Both of them are generally spoken about overwhelmingly negatively.

Linear Discriminant Analysis

In this section, we use the Linear Discriminant Analysis (LDA) classification technique to detect if a change in messaging took place in response to the COVID-19 pandemic, and then to model and interpret that change. The Principal Factor Analysis technique was also applied as a dimensionality-reduction tool to group terms that often coincided with each other into distinct messaging subjects.

The first step in this analysis was to process the data into a usable state. First, a column labelled "COVID.Status" was added to both the TF-IDF and Word Count datasets in Python. Values for this column were binary categorical, with "Pre-COVID" and "During COVID" being the factors assigned according to whether an email was sent before or after the first COVID-19 case was reported in the United States on January 20, 2020. Then, a script was run in R on the Word Count dataset to tally and store the total counts for every term as well as separate counts for the terms pre-COVID and during. Finally, terms that had word counts disproportionally on either side of the COVID.Status line were removed from the working dataset. The purpose for this was to remove any dead giveaways that would dominate as linear discriminants later in the analysis. For example, when LDA was run without this step, it showed that the terms "coronavirus" and "fauci" have been mentioned much more frequently since the pandemic began. While technically true, we did not need to run a complex analysis to arrive at this conclusion. The cutoff threshold for this step was 85%, as anything lower would have started to cut off key terms like "science", for which 81% of mentions were Pre-COVID.

The next portion of the analysis was to run Principal Component Analysis (PCA) and ultimately Principal Factor Analysis. This is useful as we are more interested in the general change in messaging surrounding larger subjects than we are in usage change of specific words. For example, one of the first principal factors calculated encompasses many terms related to the subject of Science, like 'study'. Rather than investigating the difference in the usage of the word 'study', it is far more reliable to chunk this term in with others that it is highly correlated with and analyze them as one larger subject. So, the next step was to run an exploratory PCA on the TF-IDF dataset to determine the number of components to include in later steps. Because the number of components calculated by this procedure is $n - 1$ and in this case n approaches 5000, the first components do not cover a significant amount of the variance and so a review of the scree plot shows no obvious cutoff point. Additionally, including sufficient components to reach 70% of variance explained required too many components to be reasonably interpreted. Ultimately, the number of components appropriate to include was decided to be 20. The variance explained by these 20 components was a subpar but acceptable 55%, with the components themselves remaining highly interpretable, especially after undergoing VARIMAX factor rotation in the following step. There was also the concern that too many components would split the messaging subjects into specific, possibly time-dependent subtopics that may fall into the dead-giveaway territory described in the previous paragraph. Indeed, this effect is apparent even in the 20 components generated, but it was felt that reducing the number of components further would damage an already low amount of explained variance.

As indicated above, the next step was to run the TF-IDF dataset through Principal Factor Analysis using the predetermined 20 components (referred to as factors, going forward). As interpretability is key for this analysis, VARIMAX rotation was also applied in this step. The result from this output was mostly highly interpretable factors, but there were also a few factors containing unfamiliar terms that were deemed uninterpretable as well as factors loaded with nonsense residual

HTML leftover from the data cleaning process. Concerning the latter, much time was spent trying to spot clean the leftover nonsense terms, but ultimately it was decided that the best course of action would be simply to exclude the factors loaded with these terms from analysis. Additionally, those factors deemed uninterpretable were also left out from the analysis moving forward.

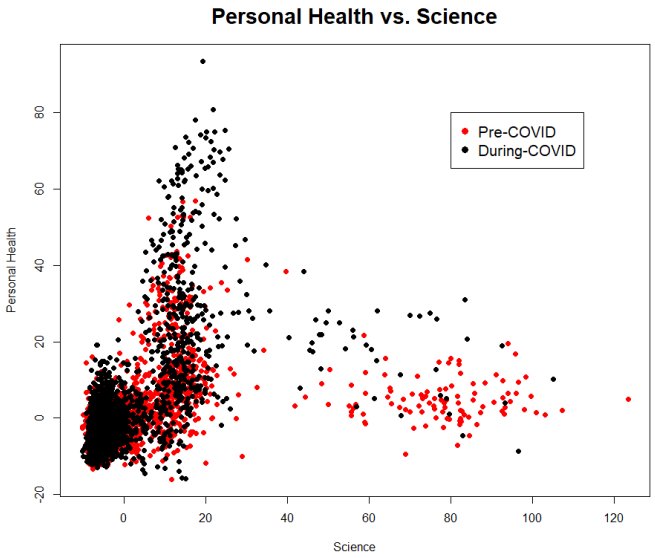
Science		US Politics in Media		Personal Health	
Term	Loading	Term	Loading	Term	Loading
science	0.84401	dems	0.56957	lifestyle	0.66480
latest	0.73943	today	0.53452	late	0.47113
having	0.59166	blast	0.50712	treat	0.40342
discover	0.56862	latest	0.46534	need	0.38907
daily	0.46846	tucker	0.42432	stay	0.38800
scientist	0.46369	gutfeld	0.38433	crisis	0.36238
ancient	0.43512	dem	0.35799	today	0.35766
nasa	0.42978	reacts	0.35563	date	0.34708
study	0.41348	channel	0.33819	important	0.33720
mysterious	0.40285	left	0.32099	news	0.33270

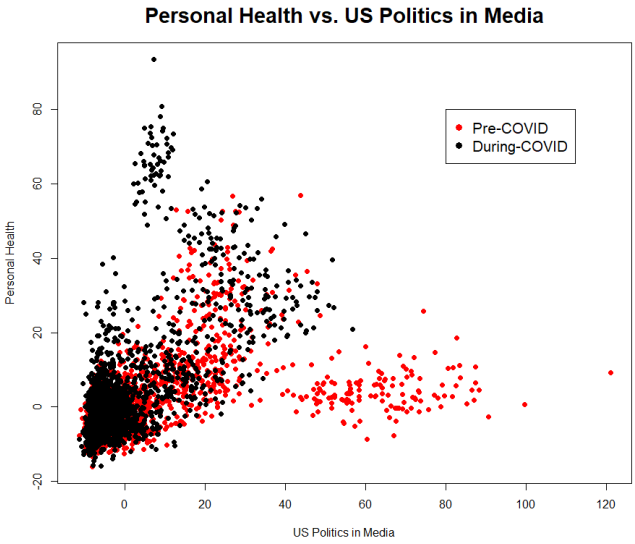
Table 1: Top 3 Principal Factors with loadings reported for their top 10 terms

That left 14 key factors with meaningful interpretations. While all of these will be included as inputs to the LDA model later, focus should be kept on the first three factors for reasons which will be made clear further in the analysis. These factors are shown in more detail in Table 1. The first factor is, notably, Science. This factor is rather self-explanatory, with some top terms like "science", "discover", and "scientist" giving context to some of the vaguer terms, like "latest". The next factor is labelled as "US Politics in Media", including different terms for the political left like "dems" and "left" as well as the sensationalist verbs reporters love to use these days, like "blast", and finally Fox personalities "tucker" and "gutfeld", referring of course to Fox News TV favorites Tucker Carlson and Greg Gutfeld. This factor is interpreted as being coverage of the interactions between political entities in the US and the media. The third factor has been labelled Personal Health. The "Personal" part of this label largely comes from the top term "lifestyle", with the rest of the terms like "crisis" and "treat" indicating that this component is the closest we will get to discussing the pandemic directly since all references to COVID would have been filtered out earlier. One more factor that is not analyzed in detail is the dichotomous factor labelled "Supreme Court vs. Catholicism", which has several terms related to the supreme court (including a strong contribution from the late Ruth Bader Ginsburg) with strong positive loadings, and terms related to the catholic church on the negative end. This factor is not investigated in this analysis, but it was a funny quirk in the data that bears mention.

Once the factors were interpreted, the data was ready for LDA. The independent variables for this model include the 14 interpretable factors, as well as the VADER.Positivity, VADER.Negativity, and VADER.Neutrality scores (since the VADER.Compound score is calculated by subtracting the Negativity score from the Positivity score – both of which are already included – the Compound score was excluded from the model). The response variable is, of course, COVID.Status.

Since this response is binary, the output from LDA includes only one linear discriminant. The coefficients of this discriminant can be seen to the right. When considering this output, it is important to note that these coefficients are characteristic of the Pre-COVID class. This can be determined by comparing the signs with the group means also calculated by the model. The positive coefficients for both Science and US Politics in Media indicate that these subjects were covered more explicitly prior to the pandemic, while the negative coefficient for Personal Health shows the opposite for this subject – Personal Health coverage largely increased after the pandemic began. Both of these trends are illustrated in the scatter plots below. Concerning sentiment change,





the coefficients for the VADER scores indicate a shift towards more positive and less negative messaging since the pandemic began, as well as an increase in neutrality.

It should be noted that while there are factors that have coefficients with magnitudes higher than the three key factors, they tend to fall into the pitfall of specificity described previously. For example, the factor labelled "British Royal Family" consists almost exclusively of terms surrounding Prince Harry and Meghan Markle, whose marriage was big news in 2018 but have not made many big splashes since then (and evidently, fewer still after COVID hit). For this reason, the coefficients for this factor along with those related to impeachment and the Democratic Primaries were not analyzed closely.

```
Call:
lda(COVID.Status ~ . - RC1 - RC3 - RC15 - RC17 - RC14 - RC20 -
    VADER.Compound, data = ds)

Prior probabilities of groups:
During COVID    Pre-COVID
   0.4752494    0.5247506

Group means:
              Science US.Politics.in.Media Personal.Health
During COVID -0.6244915           -1.506940        1.690586
Pre-COVID     0.5655814            1.364786       -1.531108
Coefficients of linear discriminants:
                                                        LD1
Science                                        0.018646014
US.Politics.in.Media                           0.040374870
Personal.Health                               -0.081309365
Congress.Surrounding.Impeachment.Hearings      0.028310456
Police.Violence.and.Protests                  -0.004908225
Economy.and.Stock.Market                      -0.001782406
Impeachment.Investigations                     0.032850345
Democratic.Primary                            -0.049489865
Tech.Companies                                -0.014945146
Nature                                        -0.007440084
Natural.Disasters.and.Extreme.Weather          0.004643815
British.Royal.Family                           0.033869716
Space.Exploration                              0.029988874
Supreme.Court.vs.Catholicism                   0.050383503
VADER.Negativity                               0.378064402
VADER.Neutrality                              -0.760602881
VADER.Positivity                              -1.118786367
```

Before these conclusions can be trusted, model diagnostics are needed. For an LDA model, the best diagnostic is a confusion matrix. Calculating this matrix gives a correct classification rate of about 65%. Although it is clearly not a perfect model, this rate is significantly higher than chance and indicates that the conclusions drawn are valid. In response to feedback from the course instructor, another analysis was performed without applying VARIMAX rotation to the principal factors to improve the classification rate while sacrificing interpretability. However, classification rate was only improved by about 1% so the original factors with VARIMAX rotation were kept as is.

Cluster Analysis

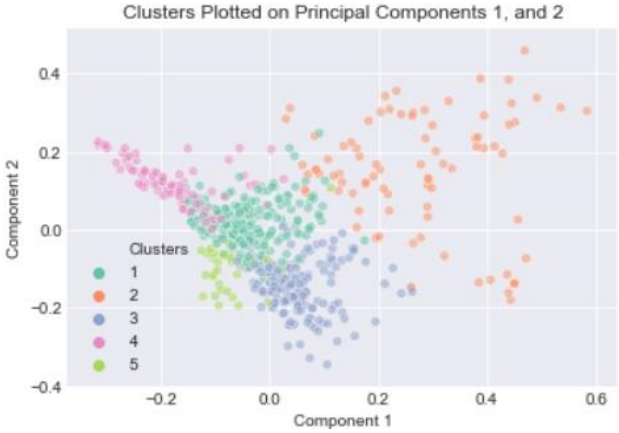
We complete a cluster analysis using KMeans, where we used five clusters to break the emails into five groups. To prepare for this analysis, the emails were filtered to just show the emails from the FoxNews.com sender name. We decided on this strategy because we figured that expanding this to include all newsletters would result in clusters that were not particularly useful, e.g. separating out the emails about sports from the Fox sports newsletter. With trial and error, we decided on five clusters, these clusters were:

Cluster Number	Cluster Name	Description
1	Grab Bag	Sentencings, Mueller Probe, Economy
2	2020	COVID, Presidential Election, Alec Baldwin[1]
3	Feuds	Fights between people that may or may not go to court, international spats.
4	Violence	Police shootings, mass shootings, terrorist attacks, car accidents
5	Supreme Court	Kavanaugh hearing, cases before the supreme court, etc.

There was a lot of overlap between the different groups, indicating a poor separation of the different clusters. For example, each of the five clusters contains some articles that concern coronavirus and legal cases.

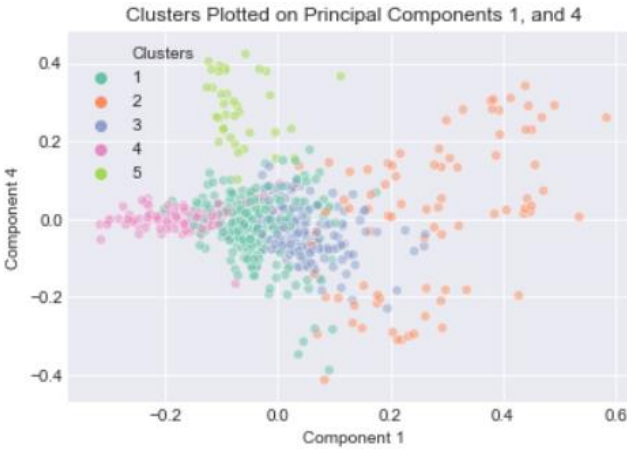
To get a better feel for the clusters, I put them into a few different graphs. To start out with, I tried using multidimensional scaling. Unfortunately, with a stress of 70,948.59, the graph wasn't very useful. This graph was not very useful for showing the clusters, this is likely due to the number of dimensions in the dataset. Next, I tried plotting the principal components of the data to show the shape of the data in just a few dimensions.

Using the first two dimensions, we can see some distinct clusters forming. Cluster 2, "2020", seems to be off to the right, and isn't quite as dense as any of the other clusters. Cluster 5, "Supreme Court", overlaps quite a bit with Clusters 1 "Grab Bag" which includes a lot of sentencings, as well as some discussion of the Mueller probe, as well as Cluster 3 "Feuds", which mainly includes fights between different groups of people. With some of the fights over the nominations of Neil Gorsuch and Brett Kavanaugh to the Supreme Court, these seem like it makes sense for that overlap to occur.



When we examine the plot of the first and fourth principal components, there is a high degree of separation between Cluster 5 "Supreme Court" and the other clusters, though Clusters 1 through 4 are dramatically more jumbled, with little or no separation between those classes.

[1] While Alec Baldwin seems random to be featured in the 2020 cluster, he plays Donald Trump on *Saturday Night Live*.



Conclusion

Through our analysis, we identified several effective strategies for quantitative analysis of Fox newsletters, using numeric features we derived from the newsletter text, namely vectorized text using the Bag of Words vectorizer and the TF-IDF vectorizer, as well as VADER scores, which summarize the sentiment of each article.

Through these approaches, we identified relationships within groups of terms using factor analysis, which showed separation between emails discussing awards, emails discussing democrats, and everyone else. We used a correspondence analysis to show how positively Fox reports on different people, noting a couple of disparities between discussions of male and female subjects. Using a combination of principal component analysis and a linear discriminant analysis, we created a classifier to review the differences between what Fox spoke about before and after COVID. Finally, we performed a cluster analysis that identified a few distinct clusters, in particular showing how common of a theme violence is in Fox's newsletters.

Individual Summaries

Alex Rosenblum

My portion of the analysis covered the different messaging focuses from Fox News between Pre-COVID newsletters and those sent after the pandemic began. Originally, my focus was set on the messaging surrounding science. This was motivated by several factors – it had become apparent to me during the pandemic that many right-wing politicians and their supporters had an inherent distrust towards science. Of course, there was evidence of this prior to the pandemic in relation to the discussion surrounding climate change, but at that point it could largely have been written off as being due to simple self-interest. After all, what should I care about the ice caps melting when my house is miles from the coast? Why should I ride my bike to work when it's so much easier and faster just to drive? Once the COVID-19 pandemic hit, however, science and self-interest should have been in agreement – the science shows that COVID-19 is devastating to a person's health and in many cases deadly, and so subtly infectious that one infected individual can infect tens of others without feeling even a little unhealthy themselves. So why then am I still hearing horror stories from my nurse friends about their patients, sick with COVID, vehemently denying through labored breaths that the virus even exists? Why is there still so much resistance to mask-wearing when all major public and private health institutions have been recommending it for months? These questions, along with my background as a scientist in the biotech industry, were my primary motivations when deciding in which direction to take my analysis for this dataset.

At the time when I joined the team, the dataset was a fair bit larger and messier than its current form. While Peter was certainly the workhorse when it came to cleaning the data, I spent the early portion of exploratory analysis finding spots that still needed his attention. Towards the end of this phase, I

worked a little bit myself with the vectorizer he used to process the data every time a new fix had to be made, and got some hands-on experience with text-scraping and textual analysis in the process. Aside from organizing with the team to meet deliverables, my contributions towards the larger team effort ended there as we all split to work on our individual analysis lines.

I set out to apply PCA (and ultimately CFA) to the dataset to get it in such an order that LDA could be applied in an interpretable way, and in doing so I learned several things. The first was that these dimension-reducing processes can be a handy tool themselves in data cleaning. Since much of the messiness in the dataset was resultant from residual HTML code that was scraped along with the email text, and that error was repeated over many emails, several top principal components were primarily loaded with HTML text. After several times back and forth with Peter in attempts to remove all HTML from the dataset by adjusting the vectorizer code, I found the simple and effective solution of excluding those components from the analysis. Another lesson I learned during this time was that it was often best to just get the whole analysis into code and trouble-shoot it from there. I spent a lot of time trying to clean up that HTML text inside of R, when if I had just gone through with the LDA process I might have learned the previous lesson a little sooner.

Once all that was done and I had a chance to interpret the principal components and fiddle with the LDA inputs to maximize its classification accuracy, I was ready to draw my conclusions. Although I set out to see how coverage surrounding science changed during the pandemic, my approach was inherently flawed as a response to that question. My analysis could not answer the question of how science (or any subject) was covered, but rather how much it was covered. With this in mind, I was able to determine that the component containing the terms most clearly related to science was largely diminished since the start of the pandemic – indicating that science as the primary subject of a newsletter is now avoided. While not definitively answering my initial question of how Fox News treats science these days, it does indicate one possible reason why Fox's audience of right-wing voters do not trust science: the work being done by scientists simply isn't talked about anymore. Additionally, the contributions of the VADER scores to the LDA model showed that Fox News tended generally towards more positive and less negative messaging during the pandemic than prior. From my personally liberal perspective, this reads as Fox being irresponsible in their messaging in denying the severity of the pandemic by outputting a false sense of security. Of course, this deserves more investigation before definitive conclusions can be drawn. To Fox's credit, the apparent drop in science coverage was surprisingly also exhibited by the component labeled "US Politics in Media", described in more detail in the report. The conclusion I took away from this was that prior to the pandemic Fox seemed to cover party politics much more strongly, but COVID has changed the messaging climate in such a way that they felt it best to curb divisive language against the left.

Peter Jachim

In addition to coming up with the idea for this project and collecting the data, I contributed to the project by doing the correspondence analysis and the cluster analysis. I also did all of the visualizations for each of those portions. Additionally, as we worked, and teammates made observations of things in the data that I had missed, I made updates to the python data cleaning scripts to make adjustments to the data.

I thought it was a lot of fun to work with my team, we all knew we were tackling a challenging dataset, and as a result, I found that my team-members were both pretty

motivated to try to find awesome insights, which was especially interesting because this is a dataset which I had personally found interesting enough to start collecting.

I had very little familiarity with the techniques being taught in the course, so I underestimated the amount of work that would go into just extracting variables that we could actually work with. In many of my other projects, I have been able to abstract away much of the cleaning of the data, but many of the tools that I was more familiar with were not very effective on this dataset. While I have worked with text data before in python, the data is normally stored as a numpy sparse dataset after the text is vectorized. As a result, the python is normally a lot faster, and I don't have to give it as much thought, versus the sparse CSVs we created to analyze in R.

As a result of this project, as well as this course more generally, I am a lot more comfortable working in R, and I switched a codebase I had been working with in python over to R.

Chris Figy

I mainly added the word clouds and beginning exploratory analysis. I then ran the PCA analysis on the dataset. I produced a K-Means model but the one in this article is Peters. I also produced a manual calculation of the TF and TF-IDF values.

I ran a tree model on the TF and compared its accuracy to the TF-IDF running the same cross fold validation model. The results showed only slightly better accuracy for the TF-IDF verse TF. I was hoping this would show better and would like to study this area further.

I ran some additional models but time did not allow for completion. I wanted to run a n-gram analysis but was not able to complete it in this document. An n-gram analysis is the same as a single word analysis we did above but you also included two words in their order. This basically will explode the size of the matrix.

I enjoyed working with the team and also learned a little python from studying Peter's data processing scripty in jupyter notebooks.

Citations

6.2. Feature extraction—Scikit-learn 0.23.2 documentation. (n.d.). Scikit-Learn. Retrieved October 5, 2020, from https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

Hutto, C. J., & Gilbert, E. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

Sweigart, A. (2019). Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners (Illustrated Edition). No Starch Press.

Appendix A: Detailed Description of Data Cleaning

This has been an iterative process, and as my teammates have explored the data, there have been a variety of things to fix, and make adjustments to the data, and the logic in the script. Additionally, I want to ensure we are starting on a strong foundation for our remaining milestones by focusing on getting the basics right now.

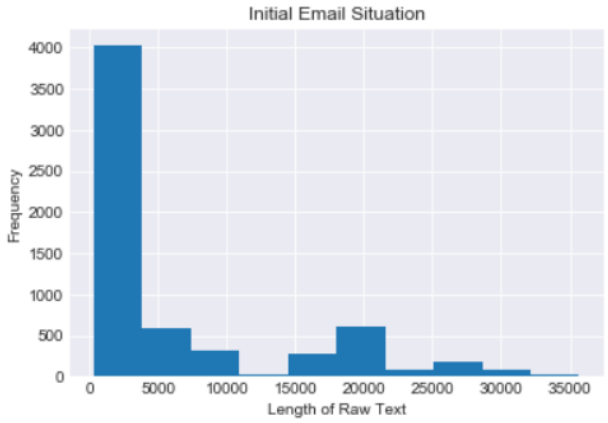
Data Collection

I started collecting these emails a couple of years ago with the intention of analyzing them as I learn more techniques for analyzing data, so I signed up for Fox's newsletters with a designated email and let them accumulate over the course of two years, and I thought that these might make an interesting dataset for this class.

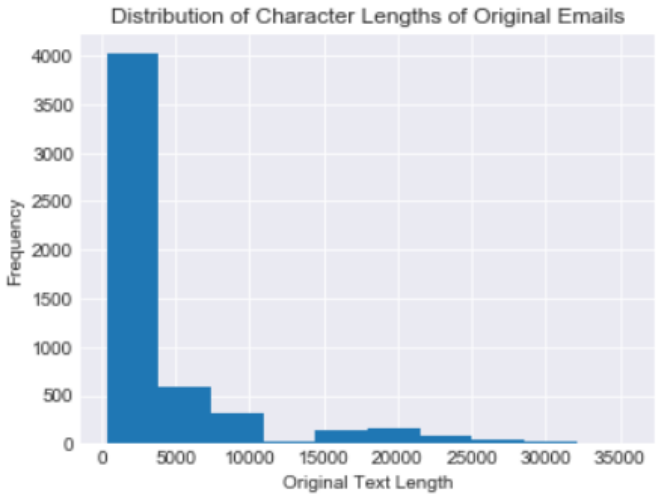
I wrote a Python script to download the emails, and to parse them using a script I wrote a couple years ago, mostly relying on tricks from the book Automate the Boring Stuff with Python, (Sweigart, 2019) and it created a pandas dataframe from the emails that I could output as a CSV, which my teammates and I can load into R.

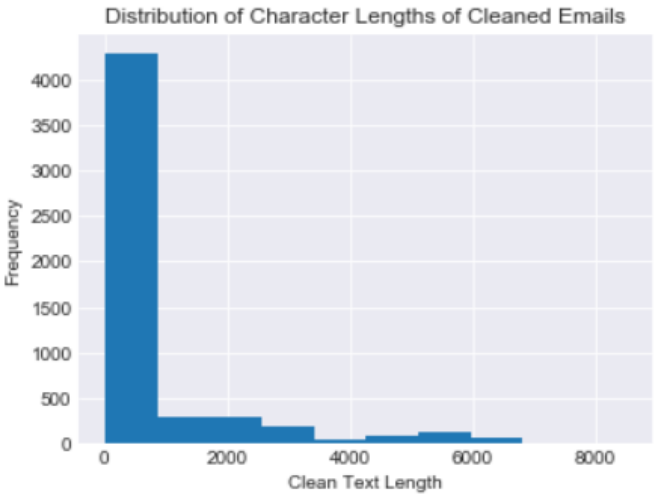
Text Cleaning

To pre-process the text, I created a column with "Clean Text", where I tried to remove the boilerplate headers and footers that don't provide much analytical value, like "Breaking News". Many of the emails also contained embedded JavaScript, which also needed to be removed. One of the ways I identified which emails are more likely to contain additional html attributes that need to be removed is I looked at a histogram of the length of all of the emails:



What's interesting in this distribution is that so many of the emails are significantly less than 5,000 characters, with sort of a lognormal distribution until the emails are about 15,000 characters in each email, before there are a bunch of super long emails. When I dug into what was in those emails, the





emails were mostly JavaScript and HTML, without any text. I identified patterns in those emails, and used regular expressions to strip out the code. The emails that remained that were more than 10,000 characters after stripping the code out were generally all code, so I dropped them from the data. The histograms below show the cleaned and raw text with the long all-code emails dropped.

Notice the difference in the scale, and the extent to which there was irrelevant text in the emails.

To help reduce the total amount of HTML text in our analysis, we used BeautifulSoup, then manually removed text using a series of strings to eliminate, along with a series of regular expressions intended to remove common phrases, along with HTML and JavaScript in the text of the emails.

To help reduce the total number of features, we performed lemmatization of the data using the WordNet lemmatizer, which we combined with a part of speech tagger. The lemmatizer helped to combine words that were different forms of the same word, so "running", "ran" would both become the word "run".

To increase our effectiveness in removing terms, we used the first five principal components of a messier version of the data to help us. Using the top terms from the first five principal components, we identified a large number of additional html related terms that we could drop from the text.

Feature Extraction

We created a few columns with VADER sentiment analysis features. These are a numeric estimation of the positivity, negativity, neutrality, and the compound score of text. I decided to use VADER sentiment analysis features because it's designed for short snippets of text, and it is intended for social media, which generally has a similar tone to the short text snippets found in these emails. VADER has been demonstrated to produce high-quality attributes for machine learning purposes (Hutto & Gilbert, 2015).

I created numeric vectors of the text using TF-IDF, which helps to weigh the number of times a term occurs in an email against the number of emails that contain the word to help weigh the effects of each point in the dataset (scikit-learn 2020). I used Term Frequency-Input Document Frequency, or TF-IDF, to create numeric features from the text. TF-IDF is commonly used in classification to help weight how interesting terms are, along with other information retrieval applications. Additionally, I removed stop words, and normalized the output. This will help to scale these variables for use in PCA so common words aren't valued as highly in the principal components. I dropped all terms that weren't in at least five emails weren't in more than 97% of emails.

At a group member's suggestion, I added a Boolean column indicating whether the email was sent before or after 1/20/2020, the date of the first confirmed case of COVID in the US. I am excited about the Boolean variable because it will allow us to compare discussions that happened pre and post COVID. I think that we will have to be careful as we are use it because not all of the changes in the emails from before 1/20/2020 are due to COVID, for example since COVID has started, Biden was nominated, so the words used to describe Biden have changed, and this is not due to COVID.