Machine Learning and Media Content: News Article Term Frequency and Political Polarization

Salil Doshi, Sam Goodgame, Susan (Eun) Park, Paul Platzman
Georgetown University: Data Science Certificate Program
Spring 2016

Abstract

America has become increasingly politically polarized while the U.S Congress has become historically unproductive. Media consumption habits might play a role in this emerging ossification. This study explores media content differentiation by examining the word choices of news sources as they relate to the word choices of politicians. We trained a Support Vector Machine model to recognize language used by Republican and Democratic 2016 presidential candidates during primary debates and classified media outlets' articles as using language that is more consistent with one party or the other. Overall, 79% of news articles used language more similar to Republican than Democratic candidates, and differences across media sources did not reflect previously observed differences in each source's audiences' ideological placements. Nevertheless, using term frequency as a tool for understanding political discourse could enable Americans to construct their own ideologically balanced diets and counteract hyperpolarization.

Introduction

Research has shown (Mitchell, Gottfried, Kiley, & Eva Matsa, 2014) that Americans are becoming increasingly polarized in their political ideologies. This atmosphere of discord has serious consequences for governance: the two most recent United States Congresses, which absolved after 2012 and 2014, respectively, passed the fewest and second fewest laws since record keeping began (DeSilver, 2014), and those outcomes were widely attributed to intensified brinksmanship between America's two dominant political parties. Many consider this partisan gridlock a phenomenon worth attempting to counteract.

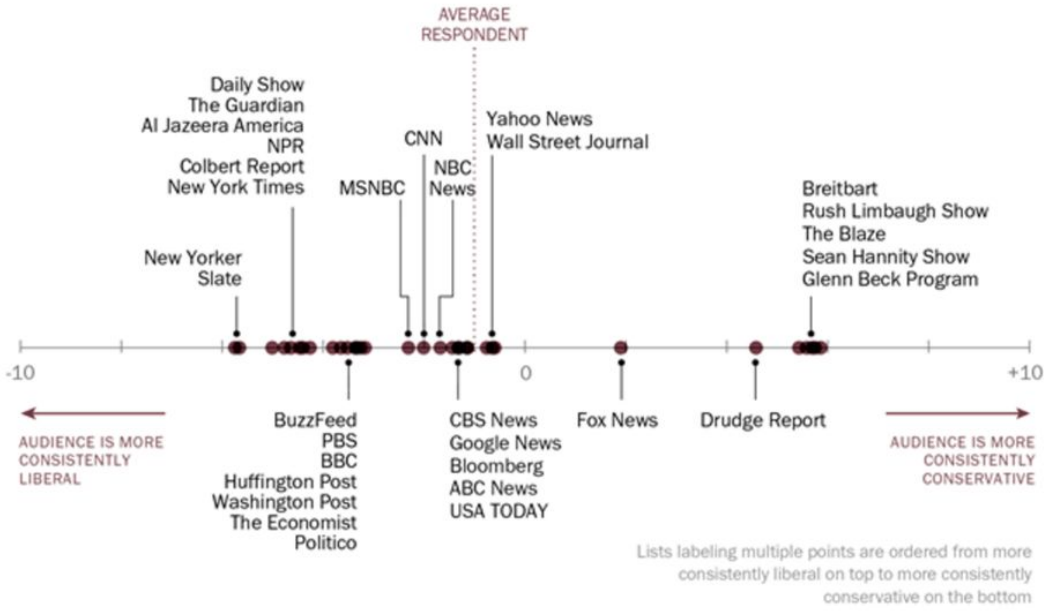
Media exposure is a potential driving force behind this increasingly intractable division. Conservatives and liberals consume different news sources that are believed to espouse and reinforce divergent political philosophies. And social media platforms are breeding grounds for groupthink; political articles on Facebook, for example, are typically consistent with users' pre-existing political orientations.

Because news sources fit nearly every political persuasion (see Figure 1 below), readers are often able to consume content that rarely strays outside of their beliefs. Social scientists have conducted a variety of assessments of journalistic bias (Bernhardt, Krasa, & Polborn, 2008; Dalton, Beck, & Huckfeldt, 1998; Prior, 2013), and they have generally concluded that news sources are ideologically differentiated. Although causation is hard to assess, if it is true that media content differentiation exists, it could presumably influence political divisiveness. Therefore, the polarization cycle of media content is an important area of study.

In our study, we sought to classify media outlets based on their consistency with language spoken by politicians. Other studies have concentrated on sentiment analysis; we focused on the extent to which media outlets' written articles corresponded with Democratic and Republican politicians' word choices during debates. This analysis positioned us to answer the following question: does media language usage vary according to the same spectrum as the political preferences of their respective audiences? If so, could that suggest a link between language choice and political

polarization? Although word choice may be a less salient dimension of media content than sentiment analysis, analyzing term frequency could identify texts' political orientations -- and perhaps be used as a tool to help address political polarization.

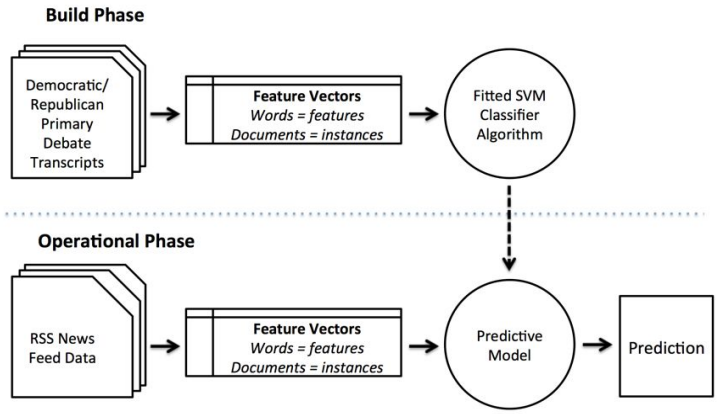
Figure 1: Pew Research Center Ideological Spectrum



Procedure

We trained a Support Vector Machine model using transcripts from the 2016 U.S. Democratic and Republican presidential debates. We parsed and vectorized the data, fit it to our model, and optimized the model. Our operational data consisted of political articles from multiple news outlets, which we parsed and vectorized in a manner consistent with our training data. We then fed our operational data to our fitted and optimized model, which produced one prediction per article: "red" or "blue."

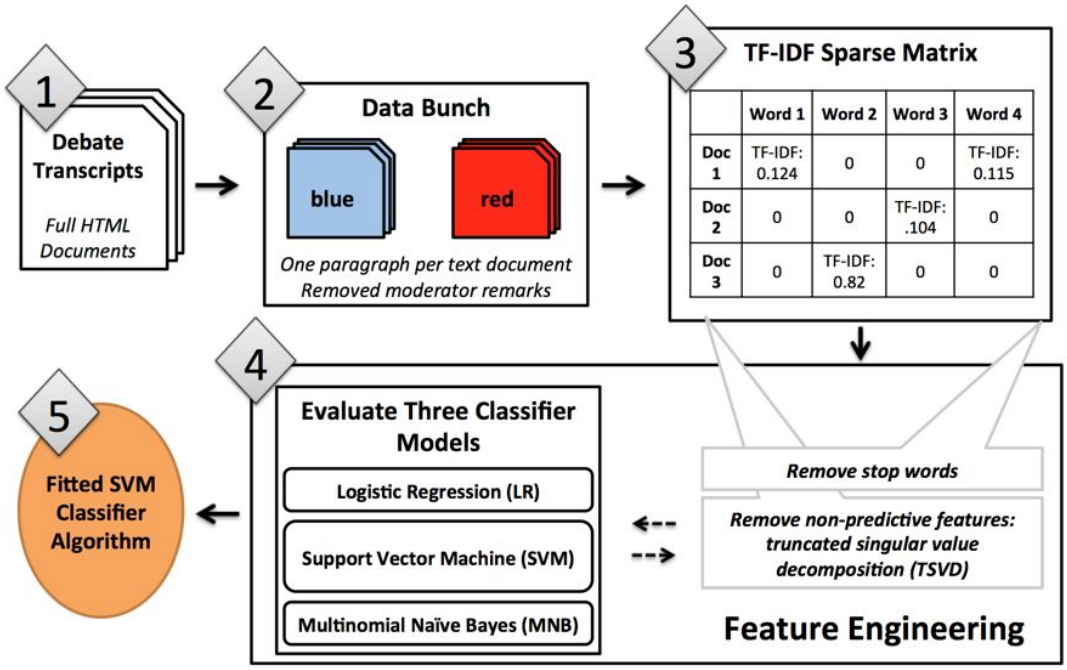
Figure 2: Project Architecture



Methodology: Build Phase

1. Ingest Testing Data.[1] We used transcripts from the Democratic and Republican presidential primary debates (Peters & Woolley, n.d.) as our baseline definitions for "red" and "blue." We vectorized our data, removed "stop words" that did not contribute to our analysis, and then calculated the TF-IDF (term frequency-inverse document frequency) value for each word.

Figure 3: Training Data and Model Fitting Pipeline



2. Parsing and Wrangling Debate Data. We parsed the HTML documents for each debate using Python's `BeautifulSoup` module, and we transformed the HTML files to text documents. Parsing each debate into one document yielded a low sample size for our model, so we re-parsed our debate transcripts to yield one document per paragraph. Doing so increased our sample size to 4,597 Democratic and 11,507 Republican documents, significantly reducing variance for our model's F-1 and accuracy scores. Next, we removed instances[2] that contained debate moderators' remarks, which reduced our Democratic training data files by 26% to 3,425 files and Republican training data by 22% to 8,997 files. We organized our text files into a data "bunch" format, which is a particular type of data organization compatible with machine learning algorithms in scikit-learn (Documentation of scikit-learn 0.17., n.d.).

3. Vectorize the Data and Generate TF-IDF Values. Our next step was to vectorize our text corpus and calculate TF-IDF weighted frequency values, which accounted for relative document lengths when determining a given word's weight. This produced a sparse matrix consisting of one

[1] See `dem_parse.py` and `rep_parse.py`.
[2] See `instance_engineer.py`.

instance per paragraph and one feature per word. Because each instance included few words relative to the entire corpus, most spaces in our feature vector were empty.

This process also included the first step in feature optimization. In addition to scikit-learn's standard set of 318 common English words, we removed candidate and moderator names from our data. This reduced the impact of context-specific dialogue in order to make our model more generalizable.

4. Evaluate Machine Learning Models and Conduct Further Feature Engineering.[3] We evaluated the performance of three models: Logistic Regression (LR), Multinomial Naive Bayes (MNB), and Support Vector Machine (SVM). We chose LR, MNB, and SVM model forms because they were each appropriate for the binary classification nature of our analysis. The LR algorithm classifies data by obtaining a best-fit logistic function. The MNB algorithm is a probabilistic classifier that applies Bayes' theorem; it assumes (naively) that the features in the model are independent. SVM separates categories in data by drawing a separating hyperplane between instances of different classes.

To select the most efficient model for our data, we optimized each using a technique called Truncated Singular Value Decomposition (TSVD). We fit our TF-IDF vector to a TSVD model across multiple iterations, and evaluated each of our three models after each iteration.

In the first iteration, the TSVD contained 11,000 features (228 fewer than the full, original set). Removing the features did not affect model performances. We reduced the number of features in an iterative fashion until we reached 2,000 features, at which point we began to observe gradual reductions in model performances. For this reason, we selected 2,000 as the optimal number of features to include in our models.

The SVM model consistently had the highest accuracy and F-1 scores, regardless of feature set size (the LR model consistently performed second best), so we selected it for deployment.

5. Parameter Tuning. After selecting SVM as our model, we performed a grid search in order to optimize the model's hyperparameters. Our objective was to further increase performance without compromising the generalizability of the model. Normally in an SVM model, both the 'C' and 'gamma' parameters need to be optimized, but because we used a type of SVM model called linear Support Vector Classification (SVC), we only needed to optimize 'C.'

Scikit-learn sets the default 'C' parameter at 1.0, so we began the first iteration of a grid search by seeking out our initial order of magnitude. We evaluated 'C' values of 1, 10, 100, and 1000, and analyzed the resulting F-1 and accuracy scores for each. We determined that 1.0 was still the optimal parameter value for 'C.' Next, We narrowed our search by one order of magnitude and examined the model's performance between 'C' = 0.5 and 'C' = 1.5 in increments of 0.1. We

[3] See `classify_svm.py`.

conducted successively more focused iterations of this search before reaching an optimal value of 'C' = 0.45.

Because the 'C' value represented the penalty or misclassification parameter, our smaller value gave us larger margins in the hyperplane of our SVC, meaning that our model wasn't overly optimized on its ability to correctly fit the training data. This afforded us greater generalizability when we introduced article text content to our classifier.

Finally, we used Python's `pickle` module to save our optimized SVM model for later use.

Methodology: Operational Phase

I. Wrangle RSS News Feed Data. We instantiated an OPML document and a MongoDB database for each of our news sources. Then, we ingested RSS feeds using a service called Baleen (District Data Labs, n.d.).[4] We exported the RSS feeds into a directory and converted them into text documents.[5]

II. Classify RSS Text as "Red" or "Blue".[6] We called the pickled SVM model and used it to classify our articles as red or blue, corresponding to language usage more consistent with Republican or Democratic candidates, respectively. Our prediction script generated an array of predictions for each news outlet, which we converted to percentages.

Results

Table 1: Model performance scores for SVM estimator

	Precision	Recall	F-1 Score
Democratic	0.76	0.58	0.66
Republican	0.86	0.93	0.89
Average/Total	0.83	0.84	0.83

Table 2: Classification matrix of SVM model predictions on labeled test data

Correct Democratic	Incorrect Democratic
392	279

Correct Republican	Incorrect Republican
1693	121

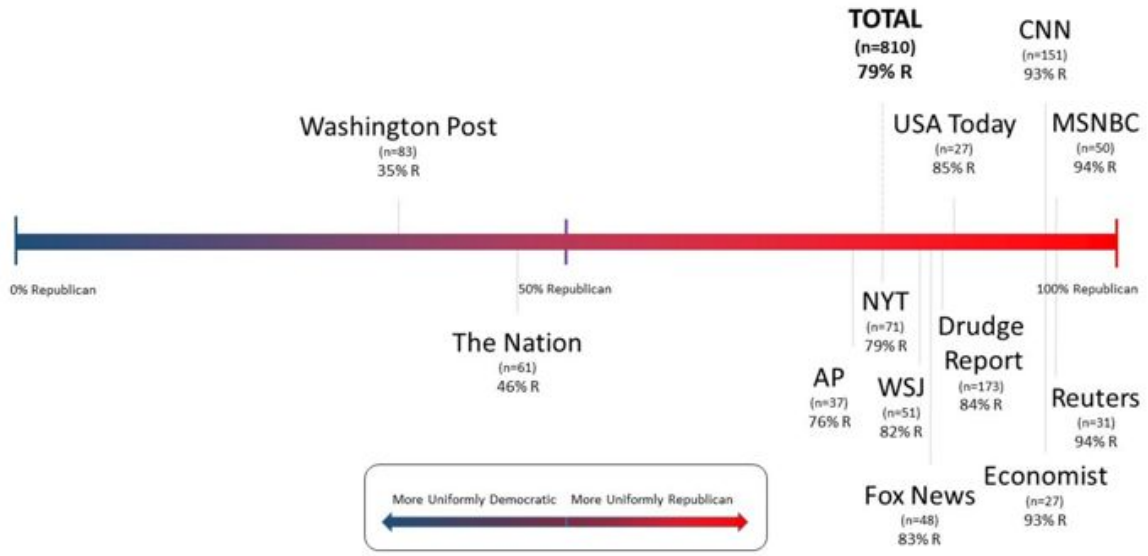
[4] Baleen is an automated ingestion service for blogs to ingest a corpus for NLP research.
[5] See `transform.py`.
[6] See `predict.py`.

The SVM model correctly classified 84% of test data. Although this represented high overall accuracy, performance was better when classifying Republican than Democratic speech. This is likely due to the large disparity in sample sizes of the two categories in the training set.

Using this model, we predicted party classifications of political article text from 12 media sources: The Associated Press, CNN, The Drudge Report, The Economist, Fox News, MSNBC, The Nation, New York Times, Reuters, USA Today, The Washington Post, and The Wall Street Journal.

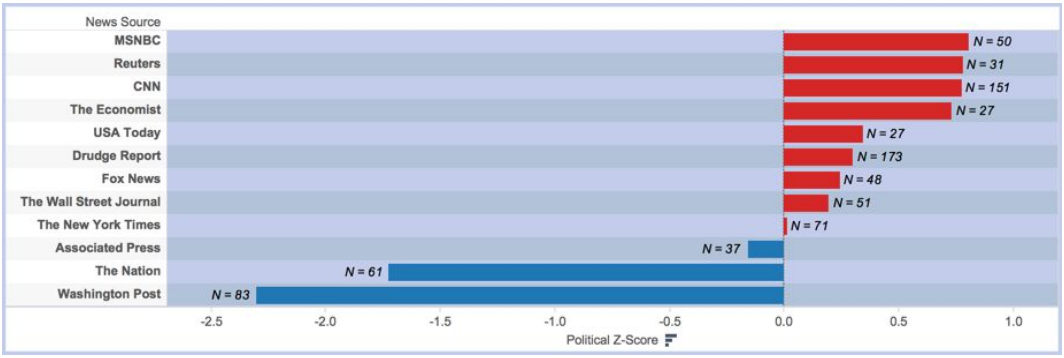
The majority of all articles were classified as more consistent with Republican candidate language than Democratic candidate language. Overall, our model classified 79% of articles in this manner ("red"); 21% were classified as "blue."

Figure 4: Political News Article Predictions by Media Sources



The range of scores (displayed in Figure 4; normalized in Figure 5) among media sources was 35% Republican classification (The Washington Post) to 94% Republican classification (Reuters and MSNBC). The spectrum according to which media sources differed on this measure did not mirror the continuum of ideological placement of each source's audience (Figure 1).

Figure 5: Z-Scores of Media Source Classification Distributions



Discussion

Our results did not match the Pew distribution of news sources' audiences' ideological placement. 79% of the news articles that we analyzed used language that was more consistent with Republican presidential candidates' speech than Democratic candidates' speech. These are interesting findings with many possible interpretations. One possible explanation is that media outlets across the political spectrum may use the Republican candidates' lexicon more often than the Democratic candidates' lexicon. Alternatively, perhaps the conservative agenda has simply been dominating the news cycle; the particular election cycle in which this project was conducted featured more Republican presidential candidates, more Republican debates, and arguably more compelling Republican content than what came from the Democratic primaries. Another interpretation is that Republican politicians might have mirrored the linguistic choices of the press more regularly than Democratic politicians. Future studies should attempt to control for such possibilities by incorporating time components into their models (to address causal directionality) and sampling a wider range of election cycles.

The rank-order of media outlets' usage of Republican language had no apparent relationship with the rank-order of the sources' audience's ideologies, which may indicate that word choice does not influence partisanship. Our results may also suggest that mapping oral transcripts to written publications requires a more sophisticated model. Perhaps it would be more appropriate to apply our SVM model to oral forms of media, such as television or radio news segments, interviews, or speeches.

Additionally, we experienced some methodological challenges. In hindsight, we could have engineered our features more thoroughly. After transforming the HTML documents pulled from RSS feeds, we discovered text documents with jQuery script tags in addition to journalistic content. Other transformed text documents contained placeholders for advertisements. Further, different news sources produced different kinds of RSS feeds. Some were long-form with in-depth analysis and reflection, whereas others simply included article titles and subtitles. However, we believe that our methodological errors were distributed randomly across our data, which likely mitigated their effects.

Future research could improve and expand upon our implementation in many ways. First, future studies could include more news sources and many more articles per news source. Second, they could even out the distribution of Republican and Democratic speech in the training set. Third, they could improve feature engineering, specifically regarding transforming data from its organic form into text documents and vectors. Our parsed data included, for example, pieces of jQuery code that may have adversely affected our analysis.

Future studies could also alter our methodology. Instead of relying solely on debate transcripts for the training data corpus, a future study could use debate transcripts to fit an initial model, then use that model to make predictions about a cross-section of article data, then feed the labeled article data back into the fitted model to strengthen and generalize it.

Conclusion

Americans are consuming news in an increasingly polarized and segmented fashion, and the rise of social media outlets as vehicles for news transmission is contributing to this entrenchment. As this problem grows, it will become even more important to identify partisan rhetoric. The ability to accurately label journalism by its place on the ideological spectrum could counteract polarization; readers could more easily consume news sources that represent diverse viewpoints. The ability to deliberately craft an ideologically balanced diet of news could contribute to broader civil discourse and open-minded engagement, which could have profound impacts for society as a whole.

References

Bernhardt, D., Krasa, S., & Polborn, M. (2008). Political polarization and the electoral effects of media bias. Journal of Public Economics, 92(5), 1092-1104.

Dalton, R. J., Beck, P. A., & Huckfeldt, R. (1998). Partisan cues and the media: Information flows in the 1992 presidential election. American Political Science Review, 111-126.

District Data Labs. (n.d.). Welcome to Baleen. Retrieved May 18, 2016, from http://baleen-ingest.readthedocs.io/en/v0.2/

DeSilver, D. (2014, December 29). In late spurt of activity, Congress avoids 'least productive' title. *Pew Research Center for the People and the Press.* Retrieved May 18, 2016, from http://www.pewresearch.org/fact-tank/2014/12/29/in-late-spurt-of-activity-congress-avoids-least-productive-title/

Documentation of scikit-learn 0.17. (n.d.). *Scikit learn.* Retrieved May 18, 2016, from http://scikit-learn.org/stable/documentation.html

Mitchell, A., Gottfried, J., Kiley, J., & Eva Matsa, K. (2014, October 21). Political Polarization & Media Habits. *Pew Research Center for the People and the Press.* Retrieved May 18, 2016, from http://www.journalism.org/2014/10/21/political-polarization-media-habits/

Peters, G., & Woolley, J. T. (n.d.). The American Presidency Project. Retrieved May 18, 2016, from http://www.presidency.ucsb.edu/debates.php

Prior, M. (2013). Media and political polarization. Annual Review of Political Science, 16, 101-127.