GitHub

The following link takes you to our GitHub repository containing the code and data described in the report below: https://github.com/julia-ulziisaikhan/nyt-comments.

Introduction

Are political topics of discussion more conducive to polarization and opposing reactions? We are interested in learning whether the sentiment makeup of comments on news articles can be used to predict if an article's topic is political. The sentiment makeup of comments will be separated into three categories: positive, neutral, and negative. Then we will calculate the proportions of comments which had negative, positive, and neutral sentiment on a given article. We will also engineer a polarization score that gauges how divisive an article is. A polarized article would have a high proportion of negative and positive sentiment and a low proportion of neutral sentiment. We will then create a statistical model to predict whether an article is political or not, based on the comments' sentiment analysis and their polarization score. The data used for this project is New York Times article and comment data.

We identify our audience of the report to be multiple groups of interest. First, political scientists can use this report to see which political topics bring the most negative sentiment, as well as which topics are the most polarizing. The second audience would be the New York Times, as well as other news media companies. It would be valuable for them to understand which articles will receive the most engagement, and price advertising on their pages accordingly. It would also be valuable to journalists who may want to create less polarizing articles and attract less negative sentiment.

The potential for data snooping

Background

The entirety of this project works with New York Times article data, and its subsequent comment data. Only New York Times subscribers are able to submit a comment and some articles prevent users from commenting entirely. According to the New York Time's website, the comment sections are moderated by humans, such that comments must be individually approved for publication. They claim to not tolerate profanity, attacks, and the like, yet the top five most commented and strongest words of negative sentiment are: 'bastard', 'b*tch', 'b*tches', 'cock', and 'n*****'. Approved comments must be in English, and should add "substantive commentary [to] the general readership" (New York Times). In addition, readers and moderators have the ability to express support for a given comment, through recommendations and Editor's selection, respectively. In the comment data, the variables of interest are the body text of the comment, and the article it is commenting on, article_id. In the article data, the variables of interest are article_word_count, headline, and keywords.

A shortcoming of this project is that the readership of the New York Times tends to skew left, older in age, higher in income, more urban in residence, and whiter (Kafka, 2022). As such, whatever findings which may arise cannot be generalized to the U.S. population.

While NYT provides a free API for their comment data, to cut down on time, we used comment and article data already retrieved and processed by a competition host on Kaggle, an online community of data scientists and machine learning engineers (Kesarwani, 2018). The timeline of the data spans 9 months in total, January through May of 2017, and January through April of 2018. There are 7,808 articles which contain comment data. We randomly selected 1568 articles and manually categorized them into news topics.

Data

The first part of this section will describe our data collection process and our data in more detail, and its relation to our research question. The second part of this section will spell out our exploratory analysis of the data, with figures and tables, to get you familiar with what we are working with.

You can download the data for this project from Kaggle here. The timeline of the data spans 9 months in total, January through May of 2017, and January through April of 2018. In total we have 7808 articles that contain comment data. We randomly selected 1568 articles and categorized them using our political lexicon, which we will explain further on in more detail.

Instead of using the New York Times API, we used the Kaggle data because of time constraints and difficulties in retrieving data using the API. We attempted to use AashitaK's Python package, but it appears that the New York Times API was updated since the creation of the package, as it ran into an error on Python. Since we are not familiar with Python, we could not come up with a solution and looked at solutions using R. There are two R packages dedicated to the New York Times API, but we found that we could not easily obtain comments data using them. Additionally, there is a query limit for the New York Times API of 1,000 articles a day. We decided to prioritize improving our algorithm rather than trying to obtain more data. We made our algorithms and lexicon to be reproducible using the most recent data. For example, our political lexicon includes words related to the current presidency and on-going conflicts across the world, such as in Yemen and Ukraine.

Methodology

The methodology of this project has three main parts. First, we performed sentiment analysis, so that we assign a sentiment value to each individual word, then comment, then create variables which comment on the sentiment makeup of an individual article. Then, we created a news topic categorization system to capture the diversity and differences of news media topics.

Lastly, we created several regression models to predict our newly created sentiment-makeup variables, and then we interpreted said models.

We used R to create all models, figures, and analyses mentioned henceforth.

We will now describe the iterations we have performed, algorithms we have created or implemented, and features we have engineered.

We engineered two features for this project, the sentiment analysis and the polarization score. We utilized the sentiment analysis feature to create the polarization score. The polarization score informs us on how divisive an article is. A highly polarizing article would have a similar share of both negative and positive comments.

The equation for the polarization score is as follows:

$$polarization = (\frac{negative}{positive} - 1) + neutral$$

Where *negative* equals the proportion of comments with negative sentiment in a given article. Similarly, *positive* equals the proportion of comments with positive sentiment and *neutral* is the proportion of comments with neutral sentiment.

Results

Evaluation of features engineered:

Evaluation of prediction models:

Conclusion

We found that

Critique

The final project requires a written critique about another group's final project. Here, we will use Abby Rooney's project about analyzing Shakespeare text data.

What was the initial motivation for tackling the project? What datasets were used? What aspect of the project is considered a data-mining and what is discovered? Is there anything you would have done differently? For example, Used different datasets, Used different models to explore the data, Generated a different feature, Arrived at a different conclusion from the given data. (These are simply suggestions, you team only has to come up with one thing)

Ailene's motivation for her project was to determine if student demographics and Regents test scores were important factors in a school receiving a passing rating for student achievement. She used two datasets from NYC Open Data and included data from 2014 up to 2018. The data mining aspect of this project was 'mining' a passing or not passing rating for each high school. Something that we would have done differently would be to include more data from other school levels besides high schools. This would provide more variety into the data. Since there are a limited amount of high schools in New York, including elementary and middle schools could provide more insight on what impacts a school's student achievement rating -- especially since these schools don't partake in Regents exams.

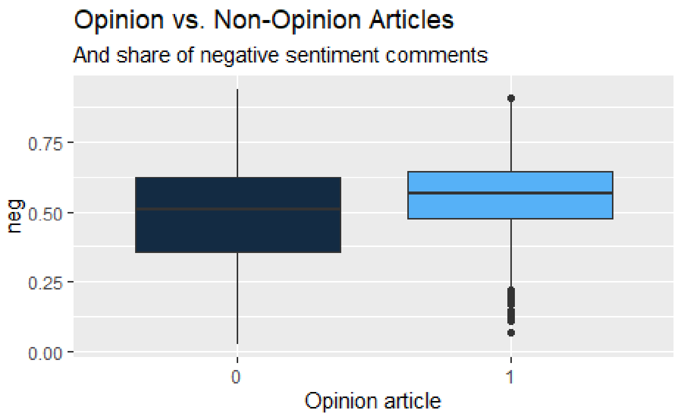

Figure 1. Histogram of *neg* values. A wide normal, almost uniform distribution.

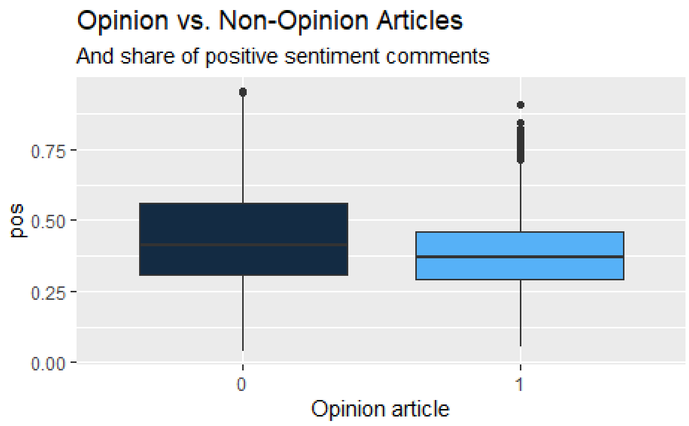

Figure 2. Histogram of *pos* values. Right-tailed distribution, median ~

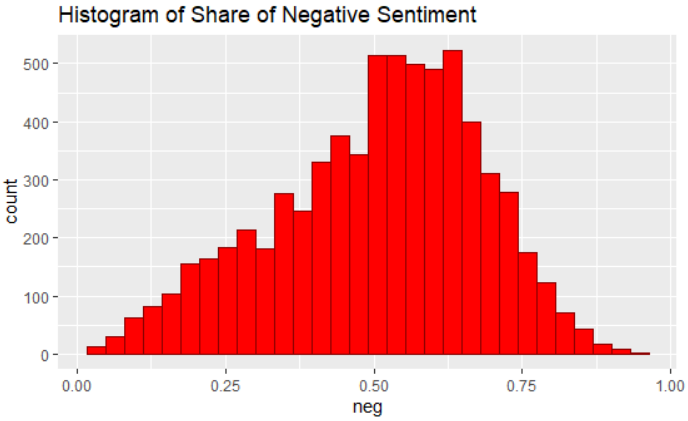

Figure 3. Histogram of *neg* values. A wide normal, almost uniform distribution.

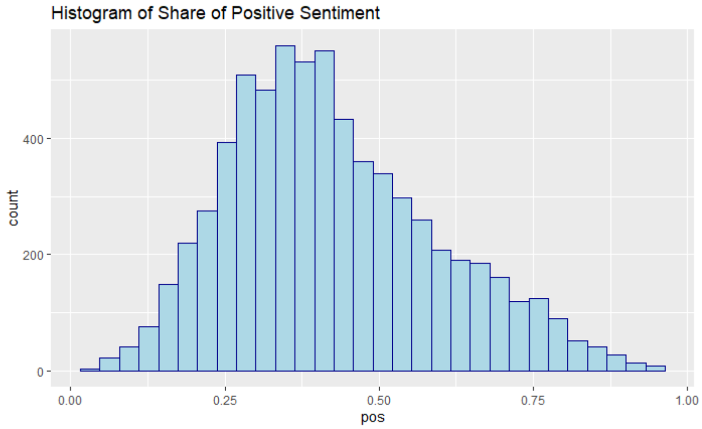

Figure 4. Histogram of *pos* values. Right-tailed distribution, median ~

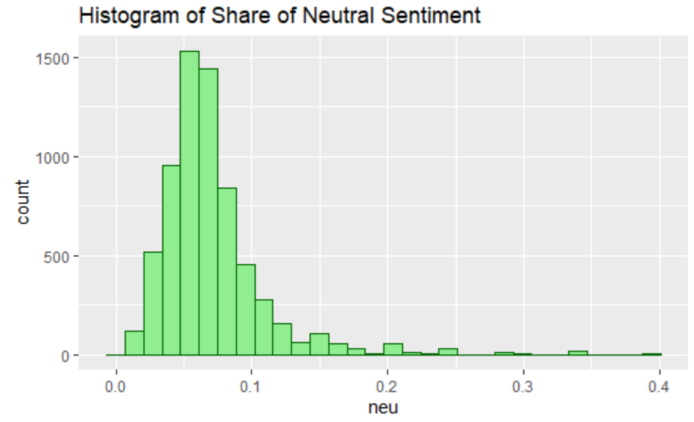

Figure 5. Histogram of *neu* values. Right-tailed distribution.

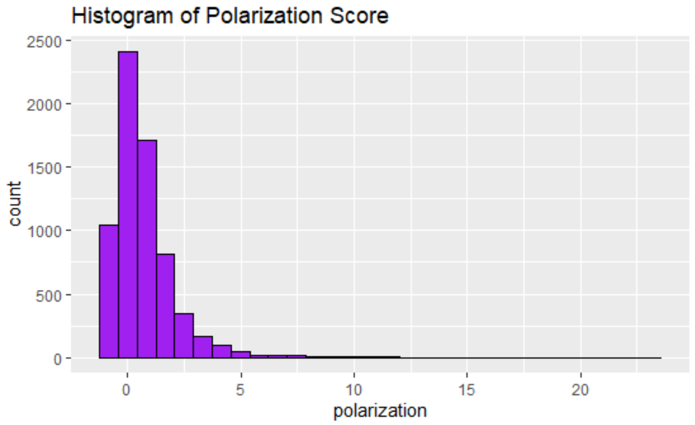

Figure 6. Histogram of *polarization* values. Right-tailed distribution, median ~ 1.