YouTube Communities: An Analysis of the Political Left and Right

Adil Chhabra & David Green

12/18/2018

Abstract

We set out to discover how YouTube communities engage differently with left- and right-leaning content creators, and how that engagement has changed in the two years since the election of Donald Trump. By accessing the YouTube API, we were able to get data such as the view count, likes and dislikes, and the text of all comments from YouTube videos. Since the other data did not provide substantive insights, we performed NRC sentiment analysis upon the comments to gauge the dominant emotions and overall positivity or negativity conveyed by the words used. We created a Shiny app to illustrate our process and allow a user to view the general sentiments in the comments section of any YouTube channel within a given timespan. In comparing the dominant sentiments of the comments sections, we found that right-leaning YouTubers tended to foster a more angry and negative community than left-leaning YouTubers. We also found that since Trump's election, negative sentiments have increased and positive sentiments have decreased across the board.

Introduction

Over the past decade, comment sections across the internet have earned an increasingly bad rap. Masked by usernames and sitting in the comfort of their own homes, people across the world have felt free to troll and bully others, spewing vitriol left and right. This is called the "Online Disinhibition Effect" (https://en.wikipedia.org/wiki/Online_disinhibition_effect). At the same time, there are still many who go to the internet for content that makes them feel good, and accordingly they react in positive and supportive ways to both the original poster and to other individuals online who share their views and feelings. Internet communities are ways for people with similar interests to connect across vast distances, and are in this way a great source of beauty that the internet has brought about. However, the members of these communities are just as likely to meet each other due to common hatred as they are to unite for more positive reasons. When someone posts hateful content, it may be viewed by many and a community may spring up of like-minded individuals, generating an endless cycle of hateful rhetoric.

In the wake of the presidential election of Donald Trump, the digital atmosphere seems to have darkened. Though Jimmy Fallon's gags and interviews once dominated the late-night scene, his viewership dropped precipitously following the election in favor of such politically polarizing liberal figures as Stephen Colbert and Seth Meyers (https://www.nytimes.com/2017/05/17/arts/television/jimmy-fallon-tonight-show-interview-trump.html). The public has looked on as neo-Nazis have committed hate crimes and the Black Lives Matter movement has brought more visibility to the constant specter of police brutality. With the current US president's tendency to tweet inflammatory statements, Twitter has become perhaps the most visible symbol of extreme internet rhetoric, but YouTube has arisen as another essential battleground for such political issues. As avid YouTube viewers, we wanted to know whether there were differences in the ways that online communities engage with content that has a more right or left political stance. We also wanted to determine the extent to which viewer feelings and reactions to videos have shifted since the 2016 presidential election. Conveniently, the YouTube API is openly accessible to the public and includes all relevant video data. From initial analysis, it became apparent that simple statistics such as view count, subscriber count, and likes/dislikes could not clearly enough characterize the viewing community or portray their feelings. For example, even though we were interested to learn that viewers of left-leaning content liked or commented 8% of the time while viewers of right-leaning content liked or commented 4% of the time, this was such a small fraction it was not particularly generalizable to the population of viewers. Thus, we turned to the comments section, where the heart of YouTube community lies. Using the NRC lexicon of sentiment analysis, which considers certain words as positive or negative and also counts some as reflecting eight fundamental human emotions, we created visuals illustrating what emotions were conveyed by groups of comments.

Ideally, we would have performed our analysis over a great many channels, but the command to get all comments requires a sizable amount of time for each video and was not feasible over too many videos. We addressed this issue with a two-pronged solution. First, we created a Shiny app that could have any channel and timespan inputted to get comments from all videos posted within that time period. A visual would then be created for the gathered comments, showing the user their dominant NRC sentiments. This approach would prevent too much data from being run while giving the user information about a channel that reflected their interests. Second, we chose only four YouTube channels to help address our central questions: Contrapoints, Shaun, Steven Crowder, and Ben Shapiro. The first two lean very left and the latter two lean very much to the right. While Steven Crowder has by far the largest viewer base out of this selection, the other YouTube channels have a similar format and style, and we consider this a more important variable to keep constant than view count.

With these four channels, we were able to assess differences in community sentiments between right-leaning and left-leaning content through NRC sentiment visuals based on the past three months of videos. We found that the right-leaning YouTubers had comments sections with a much higher proportion of negative and angry words than left-leaning YouTubers. The four-YouTuber approach also allowed us to assess changes between before the 2016 election and now. Through comparing comments in the three months before the election with the three months of most recent comments, we found that the proportion of negative emotions have increased and proportion of positive emotions have decreased in all videos, with different specific emotions prevalent in left and right communities.

Data

In order to understand how YouTube communities engage with left- and right-leaning content, we decided to pick two YouTubers, Shaun and Contrapoints, who produce left-leaning content, and two others YouTubers, Steven Crowder and Ben Shapiro, who produce right-leaning content. The YouTubers under both categories represent left- and right-leaning content respectively, and the viewers of videos posted by these channels represent the community responding to such content. We obtained data on the YouTubers and the viewing audience through the YouTube Live Data API and accessed this data using the 'tuber' package by Gaurav Sood. Our first step was to get authorized to access the Youtube Live Data API using credentials set up with Google. Once we obtained the authorization, we were able to use functions in the 'tuber' package to get statistics on videos, such as like count, dislike count, view count etc. using video ID's (a unique code that YouTube establishes for all videos posted online). Additionally, we were able to get data on channels, such as subscriber count, number of videos etc. using channel ID's that are uniquely associated to YouTube accounts. Lastly, we were able to get comments and comment threads posted on YouTube videos by parsing video ID's.

To begin our analysis of the four chosen YouTube channels, we start by identifying the unique Channel ID's associated with the accounts. These were found by looking at the URL on the browser upon opening the channels page on YouTube.

```r
crowderID <- "UCIveFvW-ARp_B_RckhweNJw"
shapiroID <- "UCnQC_G5Xsjhp9fEJKuIcrSw"
contraID <- "UCNvsIonJdJ5E4EXMa65VYpA"
shaunID <- "UCJ6o36XL0CpYb6U5dNBiXHQ"
```

After obtaining the channel ID, we get the videos uploaded by these channels as those are representative of what the channels promote and what the viewers engage with. To do so, we create the following get_videos function:

```r
get_videos <- function(channelID){
  videos = yt_search(term="", type="video", channel_id = channelID) #search by channel
  videos = videos %>% #get videos from past 3 months
    mutate(date = as.Date(publishedAt)) %>%
    filter(date > "2018-10-01") %>%
    arrange(date)
  return(videos)
}
```

This function searches for videos posted by the channel passed to it, using the 'yt_search' function from the 'tuber' package. It returns a data frame with each row representing a single video posted by the channel in the past 3 months. The columns include video ID, video title, date published, description and 13 additional variables. Although we would have preferred to run our analysis on all videos posted by the channels, we had to restrict ourselves to the recent 3 months due to computational and temporal limitations. After obtaining the videos, we are able to pass the video ID's associated with them to another function we created that provides us with statistics on the videos. The following is the get_video_stats function:

```r
get_video_stats <- function(videos) {

  videostats = lapply(as.character(videos$video_id), function(x){ #get stats for all
                                                                  #videos
    get_stats(video_id = x)
  })

  videostats = do.call(rbind.data.frame, videostats) #clean data, turn to dataframe
  videostats$title = videos$title
  videostats$date = videos$date
  videostats = select(videostats, date, id, title, viewCount, likeCount, dislikeCount,
                      commentCount) %>%
    as.tibble() %>%
    mutate(viewCount = as.numeric(as.character(viewCount)),
           likeCount = as.numeric(as.character(likeCount)),
           dislikeCount = as.numeric(as.character(dislikeCount)),
           commentCount = as.numeric(as.character(commentCount)))
}
```

To this function, we are able to simply pass the data frame returned by the get_videos function. It computes video statistics for all videos in the data frame passed to it using the lapply function. The various video ID's are passed to the get_stats function in the 'tuber' package, which like many functions in the 'tuber' package, returns a list. A key step here was to convert this list into a suitable data frame so that it would be easy to manipulate and visualize the data using the 'dplyr' and 'ggplot2' packages. This conversion, along with other such conversions form a common necessity across our data wrangling process. On several occasions, we were required to convert obscure/ messy data into tidy, easy to understand and manipulate data frames in order to conduct our analysis. Even more frequently, we had to convert numerics to characters and vice versa, as well as vectors to data frames and the other way around. To enable us to achieve these tasks, we learned and made use of the 'tibble', 'stringi' and 'tidytext' packages. This manipulation and conversion of data type was the most challenging aspect of our project. Finally, we were able to get videos uploaded by our four chosen YouTubers in the recent three months.

```r
crowderVideos <- get_videos(crowderID)
crowderVideoStats <- get_video_stats(crowderVideos)

shapiroVideos <- get_videos(shapiroID)
shapiroVideoStats <- get_video_stats(shapiroVideos)

contraVideos <- get_videos(contraID)
contraVideoStats <- get_video_stats(contraVideos)

shaunVideos <- get_videos(shaunID)
```

```r
shaunVideoStats <- get_video_stats(shaunVideos)
```

After obtaining statistics on videos posted by the four chosen channels in the past three months, our initial thought was to look at how viewers are engaging with these videos by looking at what percentage of the viewers like or dislike the videos. This would allow us to get a sense of how the community is responding to left- and right-leaning content - a like would mean a favorable response and a dislike would mean the opposite. However, our exploration into this led us to believe this may not be the best indicator since only 8% and 4% viewers reacted to left- and right-leaning channels respectively, and of those reacts only 6% and 3% were dislike reacts respectively indicating that viewers were more likely to leave a like than a dislike, if anything at all. Thus, we decided to adopt a different approach to how the viewers respond to the content. Our new approach was to conduct a sentiment analysis on all the comments published on the all the videos posted within the chosen time frame of three months by our four YouTubers. In order to run the sentiment analysis, we made use of the 'syuzhet' package to conduct 'NRC' lexicon analysis on the text from the comments. The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The first step in conducting this analysis was to obtain all the comment text from all the videos published by a channel within a given time frame and then, to break down the sentences into words to pair them up with their associated emotion and sentiment. To this end, we came up with a model that performs such a collection, break down and analysis. We provide a demo of the process with our shiny application that can be found here: https://adilchhabra.shinyapps.io/stat_231_-_fp/. You simply input the channel ID corresponding to the YouTube channel you wish to look at, along with a time frame. The application gets all the videos uploaded by the channel within that time frame. It does so using the get_videos and get_video_stats functions as described above. Then, it fetches the comments posted on all the videos as follows:

```r
  allComments <- reactive({
    comments = lapply(as.character(selectedData()$ID), function(x){
      get_comment_threads(c(video_id = x), max_results = 1000)
    })
    comments_text = lapply(comments,function(x){
      as.character(x$textOriginal)
    })
    comments_text = tibble(text = Reduce(c, comments_text)) %>%
      mutate(text = stri_trans_general(tolower(text), "Latin-ASCII"))
  })
```

In our wrangling process, we loop through the video ID column of the data frame returned by the get_videos function for all four chosen YouTubers. These video ID's are succesively passed to the get_comment_threads function in the 'tuber' package which returns a list corresponding to each video. Each list contains text of all the comments for that video in the textOriginal element of the list. This text is stored as a factor type. We convert the type from factor to character and then create a data frame in which each row corresponds to a single comment. The final data frame consists of all the comments posted on all the videos within the chosen time frame by a chosen channel, with each row containing the text of a single comment. As a last step, we combine the data frames containing the comment texts for the two chosen YouTubers in each of the categories - left and right. Thus, we obtain two data frames, one which contains all comments published on videos with left-leaning content, and another which contains all comments published on videos with right leaning content. For example, below are the comments on videos uploaded by our chosen right-leaning YouTubers, Steven Crowder and Ben Shapiro, in the past 3 months in a data frame representing comments on right-leaning content.

```r
str(rightComments)
```

```
## 'data.frame':    104183 obs. of  1 variable:
##  $ text: chr  "perhaps our most controversial change my mind ever! what do you think--is rape cultur
```

It is worth noting here that the computational time required to fetch comments is substantial, specially for popular YouTubers who post frequently, such as Steven Crowder. This was a major motivation behind why we restricted ourselves to only looking at videos published within the recent three months.

Now that we have our data frames containing text of comments made on left-leaning content and right-leaning content, we are ready to perform sentiment analysis on the text. The first step is to break down sentences in the comments into words.

```
# Breaking down sentences from comments into tokens (words)

tidy_left_comments <- leftComments %>%
  tidytext::unnest_tokens(word, text) %>%
  anti_join(custom_stop_words, by = "word")

tidy_right_comments <- rightComments %>%
  tidytext::unnest_tokens(word, text) %>%
  anti_join(custom_stop_words, by = "word")
```

Next, we select only those words that the 'NRC' lexicon analysis includes and assign the individual words with their associated emotion(s) and/or sentiment. We group-by sentiment and obtain data frames which each row represents a word and the variables are word, sentiment and n (number of times that word features in the comments). For example, see leftTokenScores as below:

```
leftTokenScores <- tidy_left_comments %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  #assign sentiment based on NRC lexicon
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup()

rightTokensScores <- tidy_right_comments %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup()

head(leftTokenScores)
```

```
## # A tibble: 6 x 3
##   word     sentiment     n
##   <chr>    <chr>     <int>
## 1 argument anger      1514
## 2 argument negative   1514
## 3 love     joy        1108
## 4 love     positive   1108
## 5 hate     anger       841
## 6 hate     disgust     841
```

As we will see in the Results header, using these data frames, we construct two seperate plots showing the number of words associated with each sentiment/emotion in the comments posted on left-leaning content and right-leaning content.

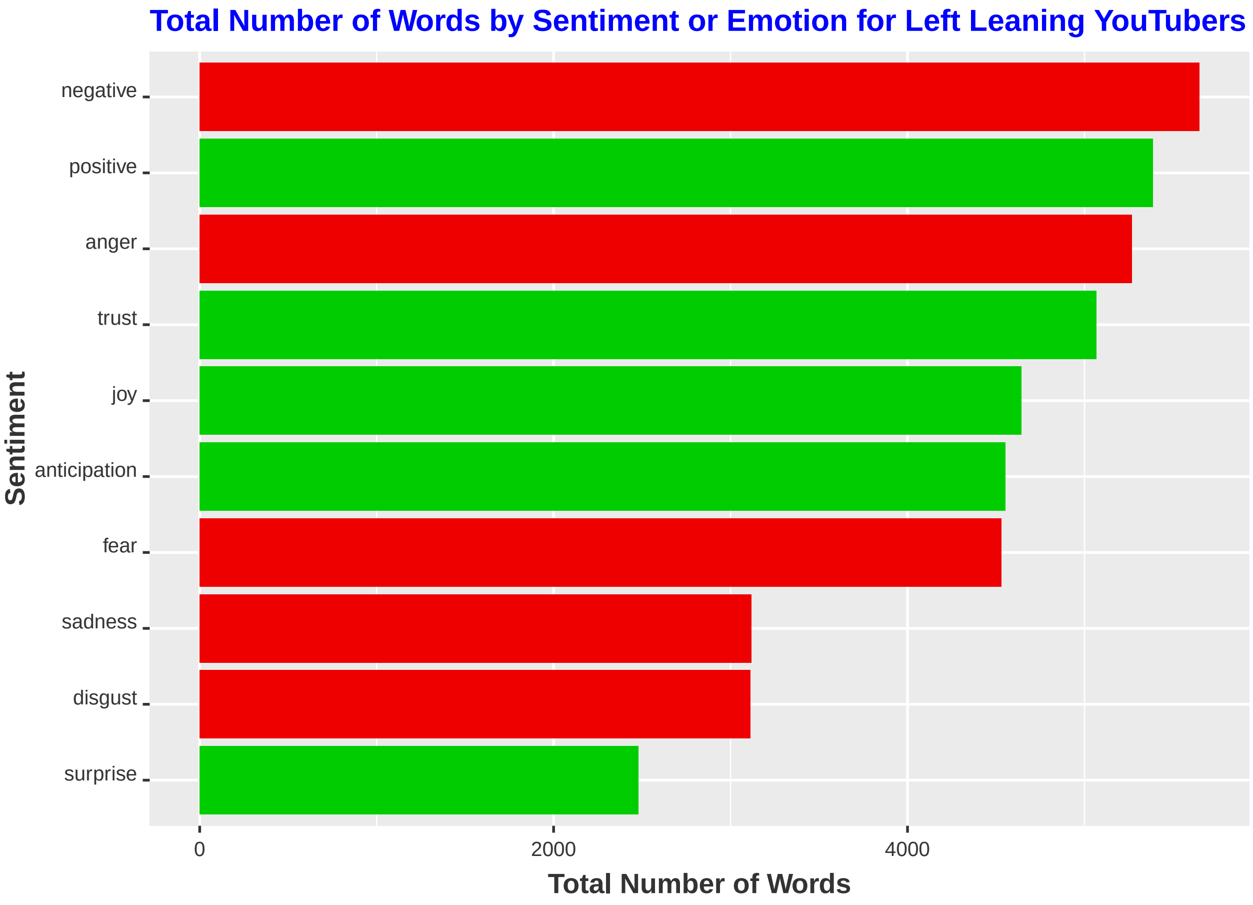
Now, we turn our attention to looking at how the community of viewers changed in the way they engage with left-leaning and right-leaning content prior to Trump's election to now. For this analysis, we consider all the videos uploaded by our four chosen channels in the three months prior to Trump's election in November of 2015. We then proceed with the same process as described above to get comments for all the videos uploaded by the channels during those three months. We run the 'NRC' Lexicon analysis on the comments and obtain two new data frames, similar to the ones described as above.

In order to measure a change in sentiment from before Trump's election to now, we create a new data frame which combines our four sentiment score data frames - "now" scores and "pre-Trump elections" scores for left-leaning content and for right-leaning content - into 1 data frame. In this data frame, each row represents a sentiment, and the variables are sentiment, content, and percent_change. The variable "content" here refers to the kind of content, that is, left-leaning or right-leaning. The variable percent_change is a percentage change in the share of sentiment. Share of sentiment, as we define it, is the number of words with a given sentiment/ number of words with an associated sentiment. Since the total number of comments posted on the videos, and thus the number of words posted might be different "now" as compared to "pre-Trump election", looking simply at a change in the number of words with a given emotion/sentiment is not sufficient. Thus, instead we look at a percentage change in the number of words with a given emotion/sentiment over the total number of words that we have associated an emotion/sentiment to. This tells us how the share of, say, words with the emotion "fear", has increased or decreased with respect to the total number of associated words.

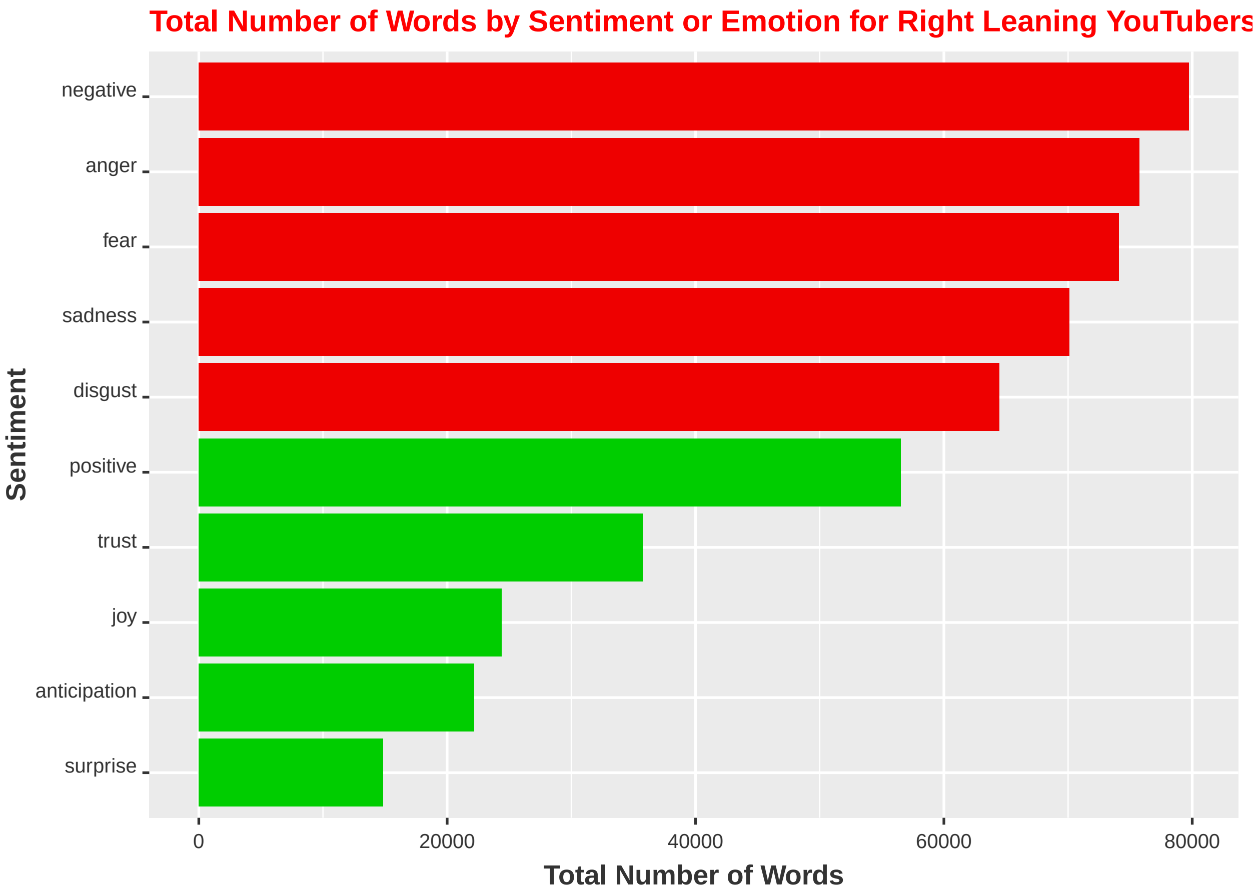
```
masterData
```

```
## # A tibble: 20 x 3
##    sentiment    content percent_change
##    <fct>        <chr>            <dbl>
##  1 anger        Left             3.35
##  2 anticipation Left             0.138
##  3 disgust      Left            -1.36
##  4 fear         Left             2.29
##  5 joy          Left            -2.44
##  6 negative     Left             0.798
##  7 positive     Left            -4.22
##  8 sadness      Left            -1.08
##  9 surprise     Left             0.731
## 10 trust        Left             1.80
## 11 anger        Right            0.935
## 12 anticipation Right           -3.09
## 13 disgust      Right            5.56
## 14 fear         Right           -0.743
## 15 joy          Right           -2.91
## 16 negative     Right           -1.78
## 17 positive     Right           -0.135
## 18 sadness      Right            5.37
## 19 surprise     Right           -0.975
## 20 trust        Right           -2.23
```

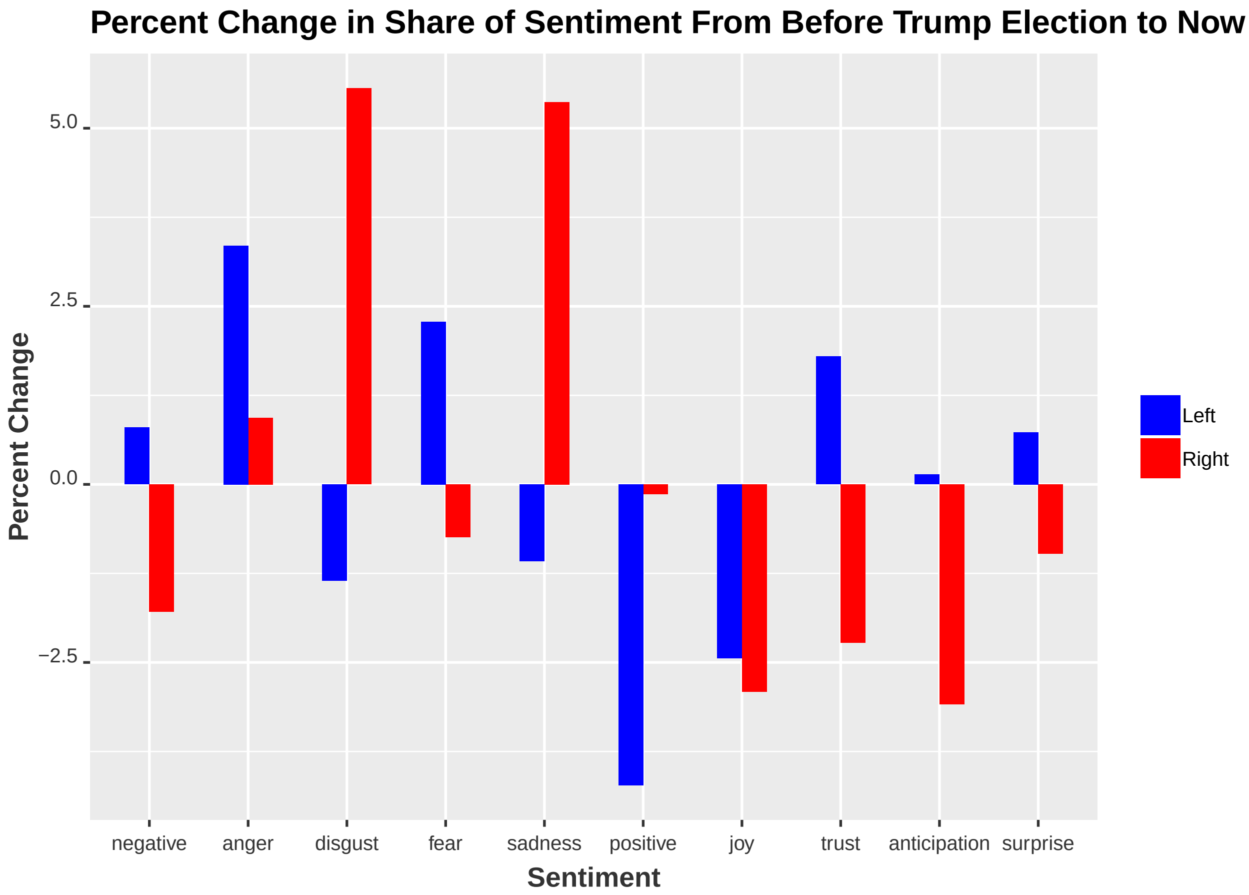
Results



This bar chart focuses on the two left-leaning YouTubers, Contrapoints and Shaun, and displays the number of emotionally relevant words in the comments of their recent videos. The red bars represent negative emotions and the green bars represent positive emotions. The number of negative-categorized words is the highest, nearing on 6,000, and negative sentiments seem somewhat dominant overall, but there is certainly a balance between positive and negative emotions in the comments section.



This bar charts stands in stark contrast to the previous one. It references the right-leaning YouTubers Steven Crowder and Ben Shapiro, showing the emotions conveyed by words in their comments sections. Unlike the left-leaning chart, this has a clear message. The sentiments expressed are overwhelmingly negative, with all negative emotions ranking more prominently than positive ones. There seems to be a relationship here: the negativity of YouTube comments can be explained by the political bent of the YouTube channel they are posted on. It should be noted that the scale here is far greater than that of the previous graph due to Steven Crowder's popularity, with almost 80,000 negative-categorized words (this, again, was the most common category).



There is a lot to unpack with this graphic. The y axis represents how much the share of a particular sentiment has changed. Anger has increased in comment sections of right-leaning YouTubers by 0.01, which means that now, 1% more of the emotion-laden words used in comments on these videos are angry than before Trump was elected. The left side of the chart is the negative emotions, and the right side is the positive. The overall trend here is that the share of words that are negative have increased and the share of positive words have decreased in these intervening two years. But within this trend lies more specific insights into the emotional proclivities of the right and the left. As far as negative emotions, the left saw an increase in anger and fear, while the right felt much more disgust and sadness. These were mutually exclusive changes, such that the left actually became slightly less disgusted and sad in the words they used. In the realm of positive emotions, the right experienced less of all positive emotions and the left strangely saw most of a decrease in general positivity. In fact, the only point of agreement across the aisle in these numbers is a decrease in share of joyous words used. A notable exception towards the overall trend of more negativity and less positivity was a spike in trust in comments on left-leaning content.

Conclusion

We first set out to find out how YouTube communities engage with both right- and left-leaning YouTubers and witness any changes that might have occurred in this engagement since before the 2016 presidential election. By all accounts, the apparent zeitgeist of gloom and anger following the election of Donald Trump found a home in the comments section of political YouTube content creators. Creating visuals based not on vague perceptions about the political climate but on raw word data inputted into a sentiment analysis provides real evidence for the change that has taken place. Moreover, the first two main visuals presented a clear picture of how right-leaning content has a tendency to stir negative rhetoric among viewers.

All that being said, these particular results lack much generalizability and cannot be stated as truth. It is essential to remember that due to computational limitations, the analysis was only run on four pre-selected YouTubers, and can hardly be said to extend to the entire YouTube community. On the bright side, the code we wrote would certainly work for any number of creators and videos, so with enough time to run code, this exact analysis could be carried out with far greater generalizability. Likewise, we did not include years of videos that would have strengthened our analysis, but with enough computing power our code could accomplish this. Furthermore, the prominence of Steven Crowder relative to the other content creators gave him an outsized impact on the results, such that the "right-leaning" results were more or less a reflection of his comment section in particular. This was especially clear from the sentiment analysis charts of right and left, in which the largest category of left was almost 6,000 words while the largest category of right was almost 80,000. While our project results presumably benefitted from choosing YouTubers with a similar style, adding more YouTubers with a wider range of viewer-bases, and possibly scaling by number of viewers or number of comments, could have standardized our findings and made them more reliable. Nevertheless, the point stands that, proportionally, Steven Crowder had an enormous amount of negative rhetoric in the his comment section. This could have been due to one of three factors, the most obvious of which is that conservative folks use nasty, sad, disgusting language. Another possibility is that liberals saw the controversial topics Crowder covers and rushed to comment angrily and troll the conservatives in the thread. Finally, the specific videos put forth by the right- and left-leaning YouTubers within the past three months likely had an impact on the rhetoric employed in the comment section. If Crowder or Shapiro put forth something especially controversial, it may have garnered hate.

In a different arena, the type of sentiment analysis we used played a major role in our results. While we chose NRC for its clear portrayal of basic human emotions, there are many others, such as "AFINN" and "bing" that have their own merits. Testing out different lexicons would have increased this project's validity, since the emotion assigned to a word is necessarily arbitrary. In a broader sense, our reliance on visuals and raw data over statistical testing never allowed us to make any claims about significance. However, this can hardly be seen as a major flaw in the project, since the structural issues stated about would have negated any statistical significance uncovered. In any statistical endeavor, the first priority must be to retrieve a sample generalizable to the desired population, and with our computational and time limitations we were unable to really make this happen. The results, however, remain genuinely of interest to everyone we discuss them with, and we now have code and access to an API that could in the future lead us to broader-sweeping insights and new discoveries about the political landscape.

The exact emotional turbulence of the post-election years remains to be seen, but our data hints at negative rhetoric amongst viewers of right-leaning content and a wave of increasing negativity in YouTube communities over the past two years.