CS 5010 Final Project  
Group 10  
Devan Visvalingam (dv5mx)  
Aneesh Sandhir (as6bw)

**News Channel Text Mining: Comment Section Analysis**

***INTRODUCTION***

We are exposed to bias in our media diet on a daily basis. Bias varies quite heavily between different news outlets, and the audiences that tune into these differing outlets also carry varying opinions. This variation of bias and political orientation is indexed by the Media Bias and Fact Checking scale (MBFC), which captures news outlets and media companies that fall on a spectrum from far left to far right bias. The MBFC also categorizes certain news outlets as distributing inaccurate or misleading information.We decided to investigate the variation among comment sections across news media outlets that differ in their political biases and accountability. To investigate the audiences of these differing news outlets, we chose to analyze the Youtube comment sections of four news media outlets with very different positions on the political spectrum; Fox News: right bias, CNN: left bias, BBC: center left bias, and Breitbart News: conspiracy and pseudo science oriented. The political orientation and media bias used to categorize these news outlets was measured by the Media Bias and Fact Checking scale (MBFC).

Before beginning the analysis, we hypothesized that Breitbart News, which was deemed as curating pseudo science content, would result in outlier information in their comments, such as more negative sentiment and lower amounts of unique words. We were also curious to see if right or left bias lead to any type of anomalies found within the comment section analysis.

***DATA & METHODS***

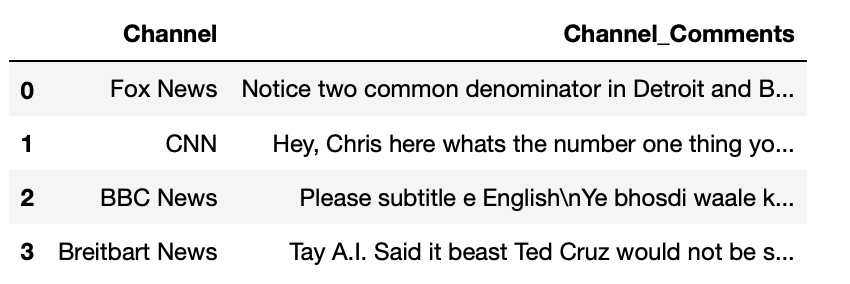
**Youtube Data API Crawler:**

This project applied various text mining strategies onto Youtube comment sections and incorporated basic and intermediate natural language processing (NLP) techniques. We sourced our data from the comment sections underneath the twenty most recent videos uploaded by four youtube channels of popular news outlets encompassing various positions along the political spectrum. In particular, our text corpus, which is the unit of measurement/size of text chunking, was the Youtube channel. This meant that four main corpora (plural for corpus) were analyzed, corresponding to the four Youtube channels, each corpora encompassing all of the comments of the 20 latest uploads for that channel. This data selection strategy allowed us to cast a wide net and analyze a wider array of topics reported by these media outlets as opposed to a list of corpura of individual videos or playlists curated by the administrators of the channel. Because the scraper collected the 20 most recent uploads from a selected Youtube channel, it could be assumed the data in some light captured current events, or at least people’s thoughts and reactions to these events.

Instead of parsing the .html response from each video’s page for comments, we leveraged Youtube’s Data API and parsed the responses from different services maintained within it. By circumventing Youtube’s user interface, we avoided the problems associated with requesting the page which holds a channel’s most recent video and the problems with parsing through ‘dynamic’ html content. Our crawler was coded with respect to flexibility allowing for the user to query the Youtube either by channel or playlist for a given keyword or phrase. The twenty most recent videos of either a channel or a playlist would be returned in a response and parsed for the names and identifiers of each video. These identifiers would be used to create retrieve responses from which the comments would be parsed while the names of the videos were being used to create the names of the files to which the comments would be written. Prior to outputting these comments, each comment would be cleansed of any emojis and foreign characters such that all remaining characters were ascii characters, and each comment was checked against the previous comment to prevent duplicate comments from overrepresenting a word of phrase in our analysis. Finally, all the text from each of output files (each video’s comment section) would then be collected into one text file (the corpus, containing the comments of the channel’s 20 most recent videos), allowing for analysis to be performed conveniently at a more macroscopic scale.

**Data Preprocessing:**

Our web crawler was ran 4 times on each of the four Youtube news channels (Fox News, CNN, BBC, & Breitbart News). The resulting text files, one per channel, needed to be compiled into a Pandas dataframe. To do this, two lists were first created: a list containing the Youtube channel names, and a second list that contained the comments in the form of large string bodies of the four .txt files. The second list retrieved the four .txt files using the ‘with open’ function in Python, and they were subsequently read, and later appended to the second list, containing all of the comments per Youtube channel. After these two lists were created, they were both appended to a dictionary as the value pairs, while their keys were ‘Channel’ (for list 1) and ‘Channel Comments’ (list 2). In this dictionary form, the columns were transferred to a Pandas data frame:



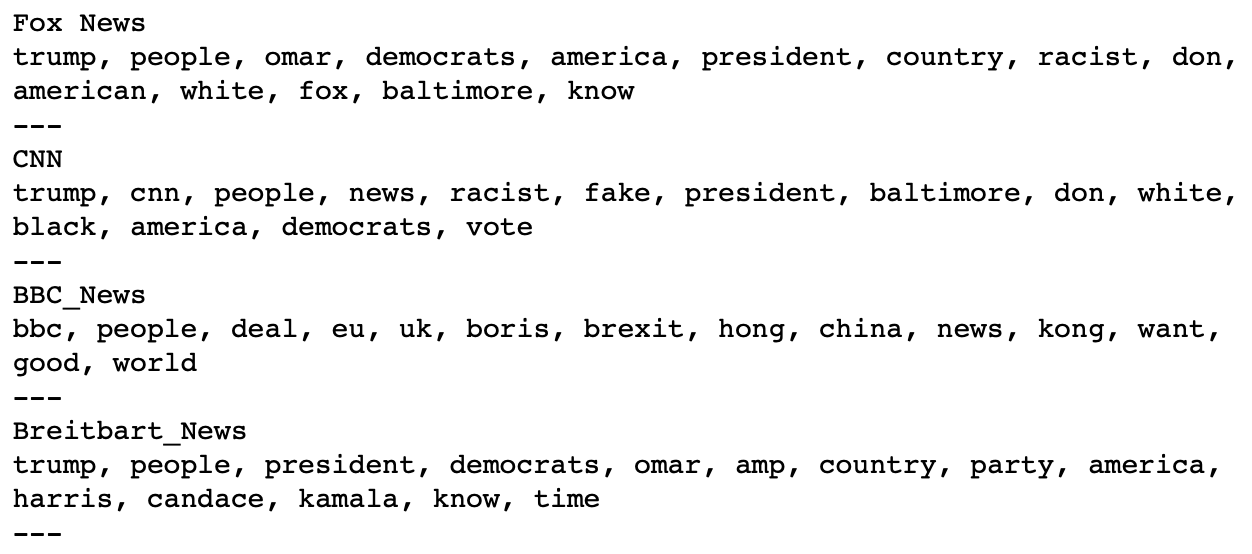
Now that the data was in an easily manipulatable format, we could look at the strings of comments themselves. For the purpose of the majority of the exploratory analysis of the text in this project, punctuation and other characters needed to be stripped from all of the Channel\_Comments strings. To do this, we created a function that ran each of the Channel\_Comments rows through various regular expressions that parsed and replaced punctuation and characters with blank strings ‘ ’. This form of text data was useful for performing word counts and other techniques that disregarded sentence structure. Later in the results, an algorithm was used that required sentence structure, so the original data frame was fed to the function to weigh sentence structure.

After getting rid of punctuation and character, a data-term matrix was created from the cleaned data to take the counts of all of the word vectors across the four channel comment sources. This was done through sklearn’s CountVectorizer function, which takes the count of individual word vectors across the string sources. The CountVectorizer also automatically got rid of ‘stop words’ inside the strings, which are common words that have little to no meaning when doing text analysis based on counts. Because of this, these common ‘stop words’, such as ‘and’, were removed in the data-term matrix.

Next, the top words across the four corpora were selected for. When doing this, we noticed that there were frequent words that needed to be removed that had little to no meaning that were not included in the original ‘stop words’ list. These words were a mix of meaningless words and formatted words, such as ‘br’, ‘quot’, ‘com’, and ‘href’.

These additional words were added to the ‘stop word’ list and we reiterated over the process of creating the document-term matrix, or simply, the count of words.

Doing so proved to be quite useful for the sake of counting common words in across the four Youtube channels and within each Youtube channel. The following table shows the top 15 words that occurred in the comments of each Youtube news channel, respectfully:

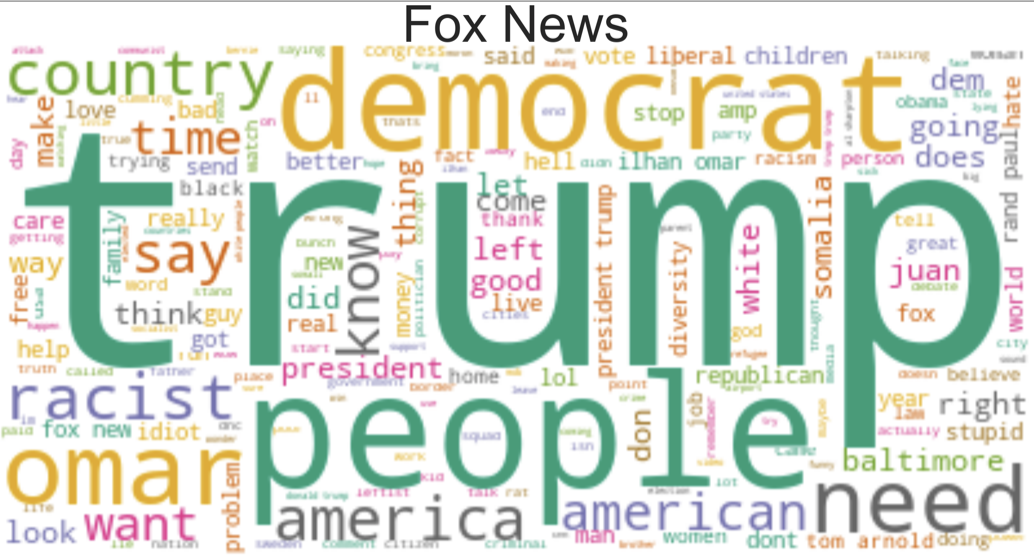


Interestingly, Trump was most common for the two major news channels, Fox and CNN, and Breitbart News, but not for BBC. When looking at BBC, it can be seen that the top 15 words had to do with international terms, which is not surprising. BBC is known for international news, while the other three news channels primarily cover United States affairs.

***RESULTS & ANALYSIS***

**Word Counts and Exploratory Data Analysis:**

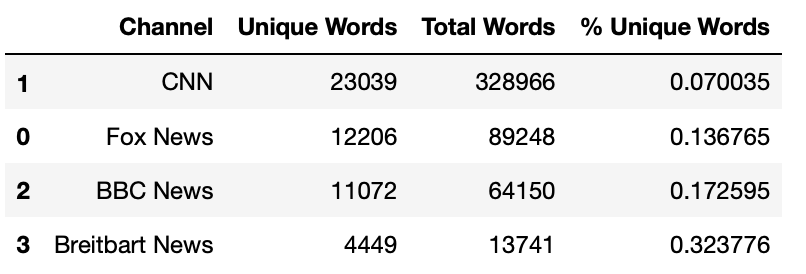
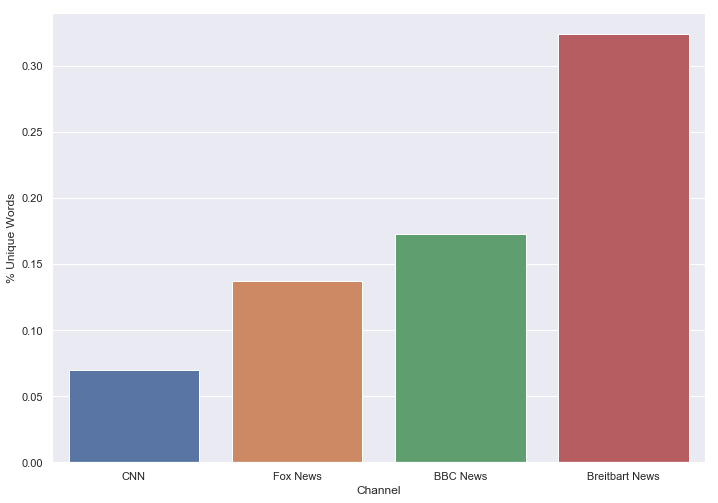
The pre-processing for this project was done iteratively as anomalies were caught during the analysis, leading to the need to backtrack. Going off of the previous table of top 15 word counts for each individual channel, we decided to visualize these findings using a word cloud. To do so, we used the Python package ‘wordcloud’.



‘Trump’ appears to dominate conversation in the comment sections of both of the major american news outlets with ‘people’ a solidly in second place. But while CNN viewers appear to express more concern about racism and fake news, Fox viewers appear to be more concerned with the president's opposition party and his critics. 

While ‘people’ appears to be the most prevalent theme discussed among viewers of both BBC and Breitbart news that is where the similarities end. BBC viewers appear to discuss more international news with the EU and Hong Kong being discussed in the comment section, Breitbart viewers appear to be more concerned with specific figures in american politics.

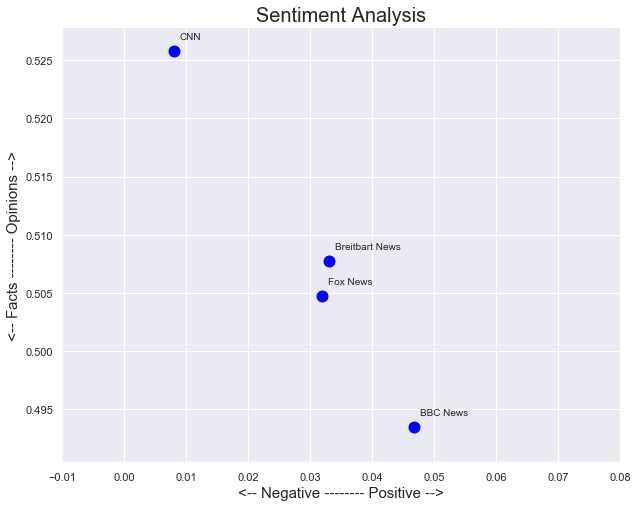
After looking at the top words for each channel, we decided to evaluate how many unique words were found in each channel. In other words, we tested the ‘vocabulary’ of each channel’s comment section. However, we had to keep in mind that both CNN and Fox had much higher views on their last 20 videos, and in turn had a much higher frequency of comments and words. Because of this, it is hard to cross examine each of the other channels to each other when evaluating their total vocabulary, but we still tried to do so by taking the % of unique words, which was the number of unique words divided by the total words of the channel’s comments. See below:



The table above shows that, apparently, Breitbart has the highest number of unique words per total words. However, this is most likely heavily skewed by the lesser amount of comments, since it can be hypothesized that there is a natural limit cap of unique words once the comment sections become large, and that repetitions start to increase dramatically. We hypothesized this was the case for CNN, which had % unique word index of .07, while there total words was very high.

**Sentiment Analysis:**

Our last point of analysis was examining the ‘sentiment’ of the four youtube channel comment sections. For this section, we used the package ‘textblob’, which performs a mix of a ‘bag of words’ and ‘sentence structure’ approach to evaluating the sentiment of the comment sections. In particular, the algorithm measures two different dimensions: ‘polarity’ of positivity, which ranges from negative to positive sentiment, and ‘subjectiveness’, which ranges from factual objectiveness (low subjectiveness) to highly subjective (opinions). The algorithm proved to be very powerful, but we had to make sure we input the original uncleaned data that included all punctuation and ‘stop words’ to account for sentence structure. The algorithm yielded very interesting results:



When looking at the y-axis, measuring subjectiveness, we see that CNN’s comment section seemed to contain highly subjective comments that were made up of mostly opinions, not facts. On the other end, BBC was evaluated to score very low on subjectiveness, containing comments that were mostly factual. Both Fox News and Breitbart seemed to be somewhere in the middle, which was quite surprising, especially for Breitbart News (which we mentioned in the introduction was rated by the Media Bias/Fact Check as ‘conspiracy/pseudo-science source’).

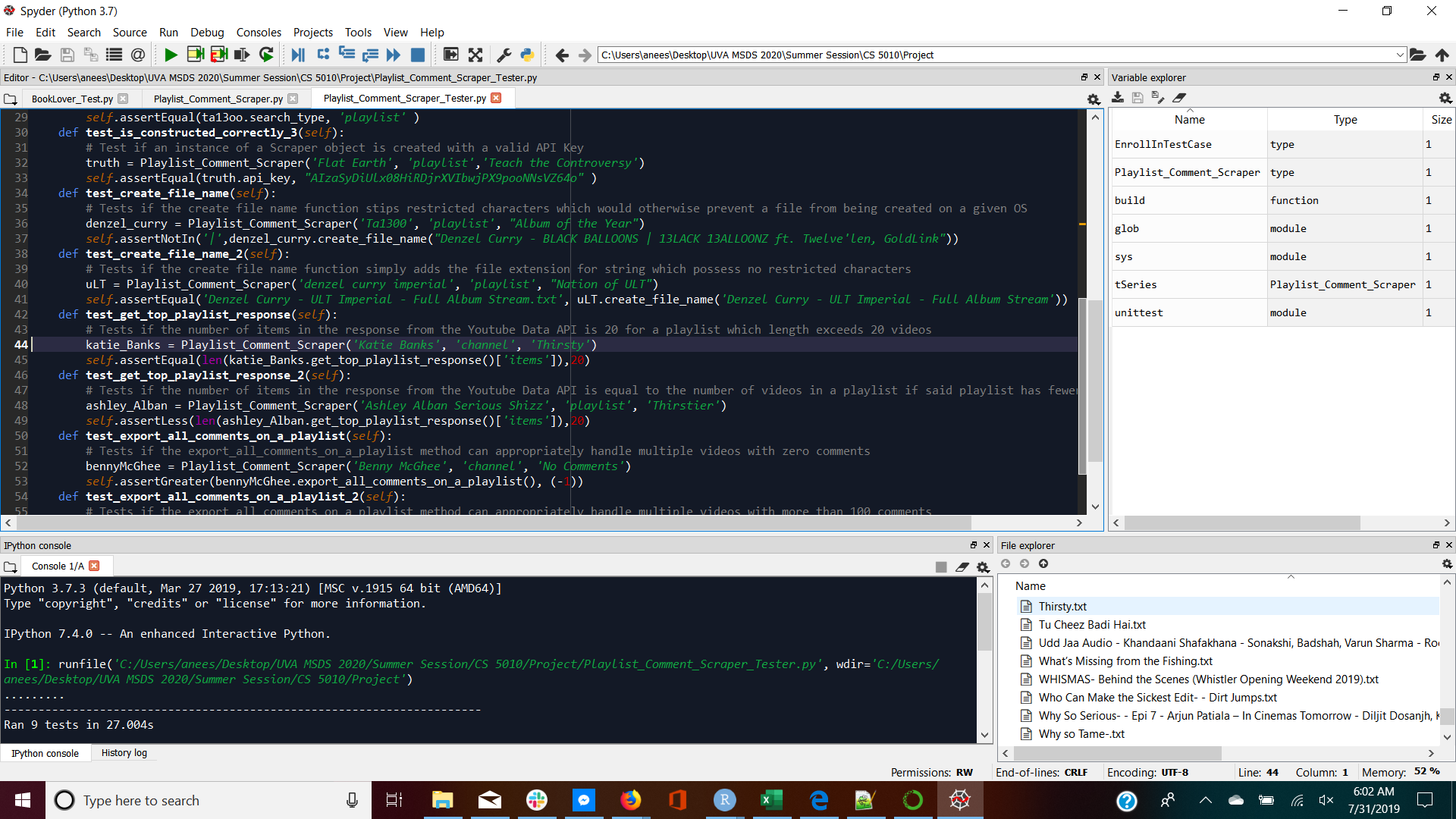
When looking at the x axis of ‘polarity’ of positiveness, more surprising results occurred. CNN was rated as containing the most negative skewed comments, while both Breitbart and Fox contained relatively more positive polarity in their comment sections. BBC had the highest positive polarity, which intuitively makes sense with their factual baseness to their comments. The surprising polarity and subjectiveness of CNN, Fox, and Breitbart is mentioned under the conclusion section, ‘The Context of Timing’.

***BEYOND THE ORIGINAL SPECIFICATIONS***

In addition to the main requirements of this project, we performed a couple of requirements that went beyond the original specifications. First, we created a ‘web crawler’ that scraped Youtube comments by accessing the Youtube API. This took considerable time to troubleshoot, and it took even more time to develop a succinct way to output the data in a way that would be digestible.

Second, because we did not scrape from a premade table, but instead from raw unstructured text, our data source was not structured in anyway. The comments had to be ordered in a way that kept them sourced to their channel and had to be engineered several times for different function applications. Because of this, the analysis and pre-processing were intertwined throughout the analysis.

***TESTING***

In addition to the main method testing we performed while developing the Playlist Comment Scraper class, we developed a suite of unit tests using the unittest library. Our unit tests were designed with positive and negative testing as well as validating the programs behavior at the boundaries in mind. We designed three tests of the constructor method aimed at verifying the API Key, search type, and keyword or phrase searches are performed with are unchanged through the initialization of Playlist Comment Scraper instance. While the full functionality of an instance of this class is only possible if the search type assumes one of two values, we did not include any functionality to handle contingencies where search type has a different value. This is because the system was designed for a user who was aware that there were no mechanisms in place to prevent or correct an invalid input for this value. We tested the create file name method with one positive and one negative unit test. The positive test was designed to see if the method would strip restricted characters from a string before creating a file with that name, while the negative test verified the string would remain the same if it were devoid of those restricted characters. The get top playlist response method was tested for each valid search type, channel and playlist. The export all comments on a playlist method was tested by adding an index which would count every comment posted at least once. Due to the Youtube Data API returning a maximum of 100 comments at a time, we tested this method using two playlists. The first was a playlist which consisted wholly of videos with more than 100 comments, this verified the code retrieved the next page token and cycled through multiple comment bearing responses. The second was a playlist composed of videos all of which had less than 100 comments many of which had none at all. Both tests yielded counts of the comments indicating the code appropriately handled both the presence and absence of a next page token in the comments response. The results of the unit tests can be seen in the figure below. 

***CONCLUSIONS***

**The Context of Timing**:

This analysis is inherently influenced by the timing of when we decided to scrape the data from Youtube. We did so during the week of the 2nd democratic primary debate, which has multiple potential implications on the nature of the comments. We conclude that the reason that CNN possibly had the lowest score on sentiment polarity was that the comment sections were flooded with debates, leading to lower positive sentiment allocation. CNN spotlighted the debates, which are inherently heated. Because of this, we also think this caused the CNN comment section to rate fairly high on its ‘subjectiveness’, which rates how far from factual statements the comments are.

CNN also had the most comments and video views out of the corpora, which we also believe could’ve inflated the lower sentiment. This is because the combination of the popularity of the democratic debates plus the larger amount of comments lead to more room to debate. We also think that CNN, when compared to right wing news outlets like Fox and Breitbart, have less political homogeneity in their comment section user base. While those that use Breitbart could be said to hold a ‘political niche’, and thus would cause less toxic comments, CNN invites more comment section users that have differing views. Of course, we did not prove this, but simply brought to light multiple interesting questions.

BBC had a lot of qualities that were expected. It is known as a more center and less-biased news outlet, and usually contains highly factual information. Its comment section both reflected the high factuality of the video content and the relatively higher positive sentiment compared to the lower sentiment polarity shown by the other three channels. CNN, Fox, and Breitbart all have a higher focus on American politics, which is more tense at the current moment.

**Future Considerations:**

Should we be given the opportunity to continue looking into this topic, we would like to retrieve more data on the comments left underneath videos via the API. In its current form the crawler currently only retrieves the the top level comments on a video while ignoring the replies and the metadata associated with the parent and its children. Retrieving the replies to comments might allow us to get a better picture of the sentiment and level of discourse amongst the communities which frequent these media outlets. While retrieving the metadata associated with theses comments and replies might allow us to analyze which members of a community are most active and when as well as how the community receives a particular comment. These features along with increasing the number of videos scrapped from a given channel comes with additional computational expense and would likely push us past the data and request limits associated with our API Key.

Another area of exploration would be seeing how the sentiment of and themes discussed by a community which frequents a given media outlet changes over time. This would be done by ordering the comments on an entire playlist by the upload time of the video under which said comments reside. The themes and sentiment could then be analyzed for each video and the change in each investigated from the beginning to the end of the playlist.

**References**

TextBlob:

<https://planspace.org/20150607-textblob_sentiment/>

Media Bias/Fact Check:

<https://mediabiasfactcheck.com>