Midterm Project Report for Python and NLP

Marina Bennett Wyss
m.wyss@mpp.hertie-school.org

Madeline Brady
m.brady@mpp.hertie-school.org

Abstract

Our project[1] aims to analyze media coverage of the 2020 US Presidential primary election. To accomplish this, we have scraped the politics sections of eight US news outlets representing the ideological spectrum: Breitbart, Fox News, the Washington Times, AP, NBC, the New York Times, Politico, and Buzzfeed News. So far we have scraped nearly 6,000 articles that mention at least one candidate, and over 140,000 sentences. Utilizing this data we analyze 1) the level of coverage each candidate receives, 2) the sentiment of texts relating to each candidate, and 3) which keywords are associated with each candidate. In terms of the level of coverage, we find that Trump is covered at least twice as often as any Democrat candidate, and among the Democrats, Biden slightly outpaces Sanders, with the other candidates far behind. Sentiment analysis using the `TextBlob` library found similar sentiments on average for all candidates, with Sanders having the most positive, and Trump with the least favorable. To study keywords, we plan to use a transfer learning approach combining BERT with CNNs to classify sentences. So far we have created a baseline model using BERT embeddings and logistic regression, finding unimpressive accuracy on distinguishing between sentences referring to either Biden or Sanders.

1. Proposed Method

We have three main parts of our project:

Coverage level: This is the simplest aspect of our analysis. We have simply filtered for articles that contain at least one candidate's name, and looked at how many articles or sentences each candidate has where they were either 1) the only candidate mentioned or 2) mentioned at all, perhaps with other candidates. This was done on aggregate over the entire time frame since early March, and on a day-by-day basis.

Sentiment: Based on the above filtering, we will use the `TextBlob` library to perform a sentiment analysis at the sentence-level where a candidate is mentioned [1]. `TextBlob` is a simple rule-based API for sentiment analysis. It uses a sentiment lexicon of pre-defined words to assign a score to each word, and then provides the weighted average of a full sentence as a sentiment score. The resulting float score is put into five category based levels (1-5) by binning. We analyze the results to see if any patterns emerge regarding how outlets cover specific candidates.

Keywords and language: For this part of the analysis, we focus on only Joe Biden, Bernie Sanders, and perhaps Donald Trump. We classify sentences in which only one candidate is mentioned, and then evaluate the sentences that were predicted to have a high probability of being about a particular candidate.

The process is as follows:

- Break the dataset down into sentences.
- Extract the candidate names, and move them to a separate column. This will give us a `text` column, with the candidate name replaced with [`candidate`], and a `label` column, which includes the candidate's actual name.
- Apply BERT word embeddings to the sentences to create the features for the classifier.
- Pass these embeddings through a CNN classifier.
- Evaluate the sentences that were predicted to have a high probability of being about a certain candidate on the test data set.

As alluded to above, this setup relies on a transfer-learning approach using BERT (Bidirectional Encoder Representations from Transformers)[2] to create the word embeddings, which are then used as a layer for the CNN classifier [3]. BERT is an open-source, pre-trained NLP model that was developed in 2019 by researchers at Google AI Language, with the key innovation being its bidirectional nature: Unlike previous context-free models like word2vec or GloVe, which generate a single embedding for each word (leading to a lack of contextual differentiation among words with multiple meanings), BERT relies on Transformer, which is an attention mechanism that learns contextual relations between words (or sub-words) in a text. The most basic Transformer includes an encoder that reads text input and a decoder for prediction. BERT relies only on the encoder, which reads the entire sequence of words at once (as opposed to left-to-right or right-to-left), thereby learning the context of the word based on all of its surroundings (again, bidirectional). These innovations have led to BERT demon-

[1] https://github.com/madelinebrady/Hertie-NLP-Python-Project

strating consistently superior performance to other similar NLP algorithms.

The word embeddings from BERT can then be used to train a CNN to make our two predictions of interest[4]. Adding BERT as an early layer to our CNN will improve our performance given our small dataset. We chose to work with CNNs because they are better for feature extraction than LSTMs (a type of RNN) due to the fact that they can better "capture short term and long term dependencies between words [5]." For tasks where feature detection in text is more important (i.e. names), CNNs are superior to RNNs, which excel at predicting sequences, something that is not necessary for our project.

To evaluate whether our CNN results in superior predictive power, we will compare the results to a baseline model utilizing logistic regression on the BERT embeddings.

Once the classification is complete, we will evaluate the top 10 or so sentences with the highest probability of belonging to each candidate. We can evaluate these using LIME, activation maps, and/or a systematic, qualitative assessment. Each of these options is described briefly below:

- Ribeiro et. al. invented LIME in 2016 to better understand the reasoning behind black box predictions[6]. LIME works by mimicking complex models with a simple model that can actually provide an explanation. In our case, LIME will create thousands of variations of our text with different words removed, and classify each of these versions as our train set to be tested on our original text. This will tell us which words in the original text were weighted the most in the model's decision-making process. The package allows us to visualize word importance through color at the per-word level, and will provide us with the most important words leading to our per-sentence candidate classification[7] .

- As an alternative to LIME, activation maps may be used to uncover the learned internal features of CNN models. In simple terms, "activation functions help decide whether a neuron should be activated. . . [which] helps determine whether the information that neuron is receiving is relevant for the input[8]". By visualizing activation at each level of the network, we may be able to uncover the most important terms attributed to each candidate.

- Evaluating deep learning models in a qualitative manner may add value to our interpretation as seen in other studies [9].

2. Experiments

2.0.1 Data:

We gathered our data set by scraping the politics sections of eight US media outlets: Breitbart, Fox News, the Washington Times, AP, NBC, the New York Times, Politico, and Buzzfeed News. So far we have scraped nearly 6,000 articles and over 140,000 sentences that mention at least one candidate. The scraping process has been set up to run automatic on an hourly basis via a cron job [10], so this data set continues to grow.

2.0.2 Evaulation method:

For our analyses of coverage levels and sentiment, we will simply rely on a qualitative assessment of the results: do they make sense, are they informative, and how does this fit into the larger picture of the election? For the classification problem, we will rely on accuracy, F1, and binary cross-entropy loss.

2.0.3 Experimental details:

Coverage levels: Overall, Trump received by far the most coverage of any candidate for President, which is not surprising given his status as the incumbent **(Figure 1)**. This trend holds true whether evaluating at the article or sentence level, and regardless of whether looking at text where Trump was the only candidate mentioned, or mentioned perhaps along with others.

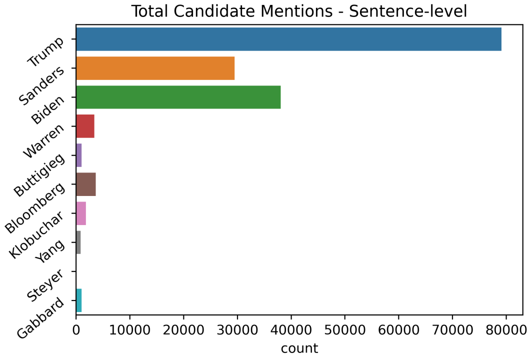


Figure 1. Total Candidate Mentions

For example, when evaluating on the sentence-level, nearly twice as many sentences mention Trump in comparison to the next-highest, Biden.

An over-time comparison shows the same trend, with coverage of Trump dwarfing that of other candidates, particularly in late March as the coronavirus crisis took a more prominent media position than the primary election (**Figure 2**):

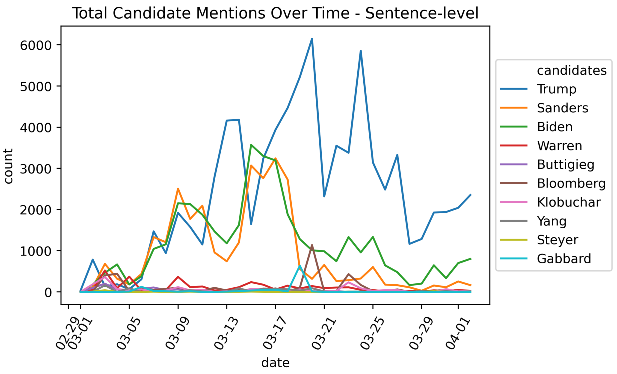


Figure 2. Candidate Mentions Over Time

When evaluating only Democrats, Biden and Sanders receive the most coverage on the article and sentence levels, regardless of whether they are the only candidate mentioned or one of many **(Figure 3)**. Biden leads Sanders by some margin.

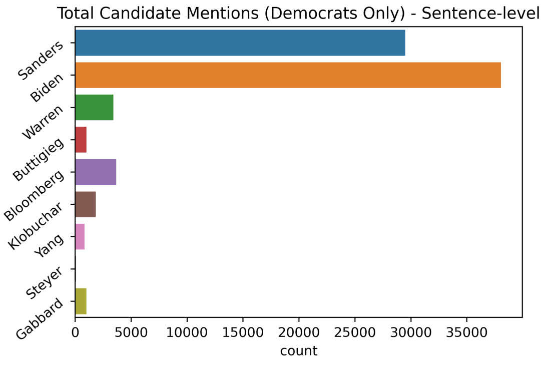


Figure 3. Democratic Candidate Mentions

We can see that early in the month, before Super Tuesday, the candidates were much more equitably covered, but all at a lower level **(Figure 4)**. After Biden and Sanders' victories and the subsequent suspension of other campaigns, they emerged as the front-runners, and media attention grew. Biden's lead over Sanders is relatively recent as well, and likely as a result of his recent primary election successes. The other candidates show spikes when dropping out of the race (Warren and Gabbard), or when they made a high-profile move (such as Bloomberg's massive donation to the DNC on March 17th). In general there has been a decline in coverage recently, probably because of the national attention on the coronavirus, though Biden has had an uptick in recent days, perhaps due to new sexual assault allegations.

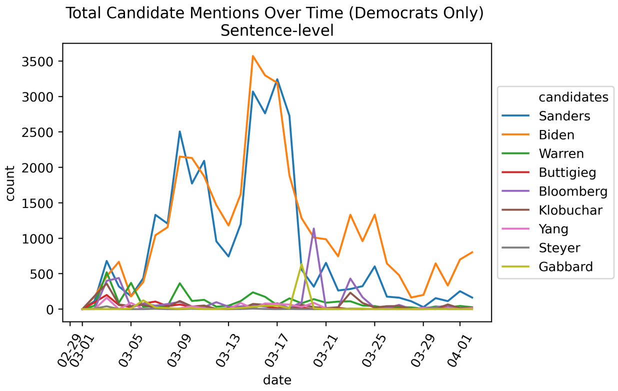


Figure 4. Democratic Candidate Mentions Over Time

Sentiment: We have run a sentiment analysis on 103,466 sentences using `TextBlob`, and analyzed the results quantitatively and qualitatively.

Regardless of publisher, most sentences were neutral, which makes sense given the media's role in relaying events and stories **(see appendix A)**. There are no clear patterns regarding extremely positive or negative language.

The average sentiment across all outlets was 3.12 and individual candidate's scores varied only slightly from this average **(Figure 5)**. Sentences mentioning Trump were on average 0.38 points lower than the overall candidate average. Perhaps this has to do with the fact that Trump is mentioned in the most news stories, which are related to world news events broadly. The general descriptives show that the number of news articles about Trump shot up drastically with the US corona-virus outbreak, which are probably more negative than articles mentioning Buttigieg's Iowa primary win.

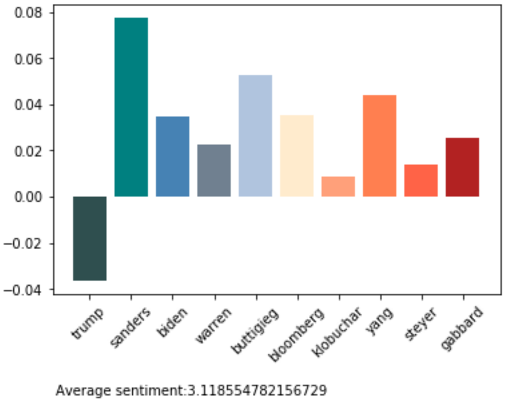


Figure 5. Candidate Sentiment Avg. compared to Overall Avg.

As a result, it may be most interesting to compare the Democratic candidates. Interestingly, previous front-runner Sander's score appears to be almost a full point above the average in comparison to the current front-runner, Biden. Perhaps Buttigieg's high score is explained by his Iowa Caucus win. Sentiment of candidate sentences varies minimally by media outlet **(see appendix B)**. Sentences mentioning Donald Trump in NBC, Buzzfeed, Breitbart and Politico were higher than his average. For Sanders, the AP, Buzzfeed, Fox and NBC were above his average while the remaining were below. For Biden, the AP and NBC were above his average while the remaining were below.

An analysis of sentiment over time leads to random results with some clear links to news coverage **(Figure 6)**. On March 5th, we would expect to see a positive sentiment spike for Biden after he won the South Carolina primary and went from under-dog to front-runner, but this is not the case. Biden's sentiment score peaked around March 20th which may be explained by his at-home corona related address. Trump's sentiment score drops severely in late March, which may be explained by the corona virus outbreak.

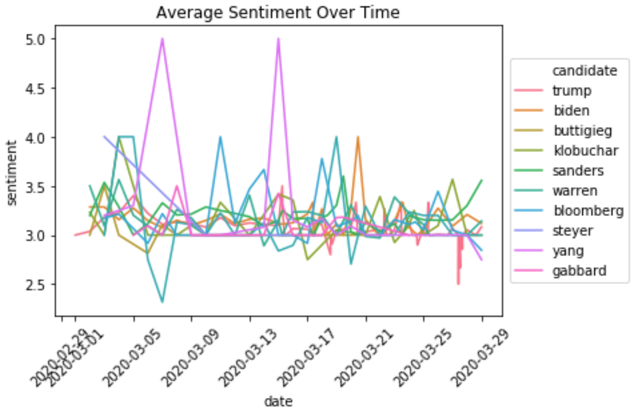


Figure 6. Average Sentiment Over Time

The above quantitative analysis shows minimal differences between candidates and outlets and the results do not appear to be conclusive. One would expect more clear patterns between partisan right and left-leaning outlets, though this is not the case.

A qualitative analysis of classified sentences was also conducted by checking random samples from each category. Sentiment analysis may not be the ideal method to capture media bias for a number of reasons. First, many sentences include a quote by a given candidate, so the sentiment analysis captures the sentiment in what Trump or Biden is saying rather than what the media actually says. On the other hand, perhaps the media selects more negative or positive quotes per candidate which may be a form of bias in and of itself. This tautological problem may pose challenges to isolating the true media bias effect. Second, sometimes the media reports on objectively negative events (i.e. financial crisis, etc.), and this sentiment analysis cannot distinguish between actual negative coverage toward a candidate and random events that occur in the news. Perhaps outlets that report in the least biased way actually show the most extreme sentiment scores because they are simply reporting on a candidate winning or losing. A more fine-tuned sentiment tool would be needed to pick up the nuance in media-bias language.

Keywords and language: So far, we have prepared our data with text and labels, and applied the BERT encodings. We have created a baseline model using logistic regression on a subset of the data (for computational reasons) - 500 observations referring to Biden, and 500 referring to Sanders. This baseline achieved an accuracy and F1 of 0.72.

Taking a look at the top sentences that were identified as highly likely to be referring to each candidate, some interesting trends emerge. For Sanders, his notation as being an Independent from Vermont (I-Vt.) seems to be highly predictive, with six of the top ten sentences just listing his name and title. One sentence stands out as a logical description for Sanders: *’[candidate] stands with them,” Casca said, ”and his agenda would bring the most powerful change to their lives.’*

The sentences discussing Biden are far more diverse. Many talk about his son, Hunter Biden: ”During the House impeachment proceedings, a career State Department employee testified that he had flagged Hunter [candidate]'s apparent conflict of interest, but was told essentially not to bother the vice president's office,” and others describe his supporters, *”Women are propelling [candidate]. Women make up a considerable majority of the Democratic electorate, and they're proving to be a key ingredient to [candidate]'s success.”* Still others describe notable personality traits and behaviors: *”At times, though, [candidate]'s penchant for snapping back has been less clearly directed at someone attacking him with false information”*

In future steps, it will be interesting to see if removing the candidates' titles may help give a more nuanced picture at media descriptions of Sanders, in particular, and if the superior classification capabilities of the CNN will highlight further interesting findings once we can analyze the results with LIME to see why these sentences were chosen in particular.

3. Future work

Up to this point we mostly focused on the Python programming aspect of our project because we needed to gather enough data via our web scraping tool to implement our deep learning models, and have only briefly worked on NLP methods through the baseline BERT embeddings logistic regression model. Next, we will move further into the predictive deep-learning portion of our project. Moving forward we also plan to limit our analysis to the two Democratic front-runners - Biden and Sanders - and perhaps also Trump. Given that the other candidates have dropped out at this point, we should be able to see more granular differences if we zoom in on just this smaller subset.

Upon building the logistic regression baseline using the

BERT encodings, we ran into computational constraints, so a major part of the next phase will be setting up AWS. Then, we will extend the baseline BERT encodings to use a CNN classifier.

We will then evaluate the results of the classifier using LIME, activation maps, and/or a systematic qualitative analysis of the sentences which were classified as extremely likely to be about a particular candidate, to see what we can uncover about how the media portrays candidates differently.

We will also re-run our analyses of coverage levels and sentiment, to see if we can see any interesting patterns emerging over time. Additionally, we will finalize our code, and write tests to monitor its performance on an on-going basis. If time permits, we lastly intend to create a Flask dashboard to present our findings on an on-going basis.

References

[1] Pete Keen et. al. Simplified text processing. *TextBlob*.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882 [cs]*, August 2014. arXiv: 1408.5882.

[4] Manu Suryavansh. Using transfer learning for nlp with small data. *Medium*, May 2019.

[5] Henry Weller and Jeffery Woo. Identifying russian trolls on reddit with deep learning and bert word embeddings. *Stanford*, Apr 2019.

[6] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

[7] Adam Geitgey. Natural language processing is fun part 3: Explaining model predictions. *Medium*, Jan 2019.

[8] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, Feb 2018.

[9] Han S. Lee, Alex A. Agarwal, and Kim. Junmo. Why do deep neural networks still not recognize these images?: A qualitative analysis on failure cases of imagenet classification. *School of Electrical Engineering, KAIST*, Sept 2017.

[10] cron-job.org. 2002-2020.

A. Appendix A

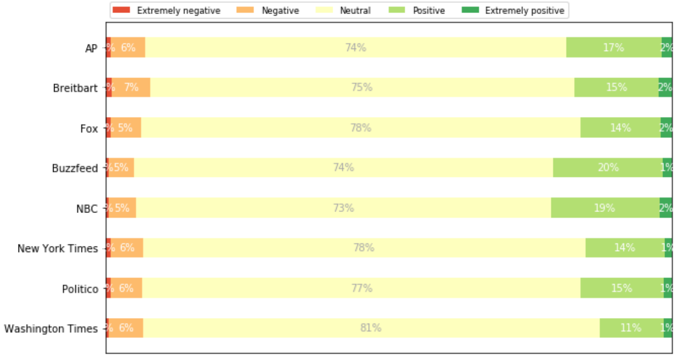
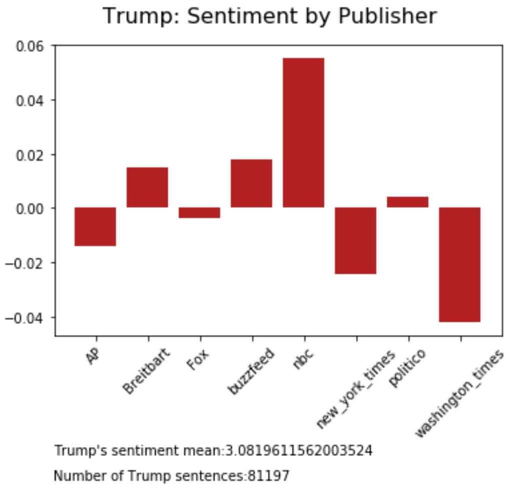


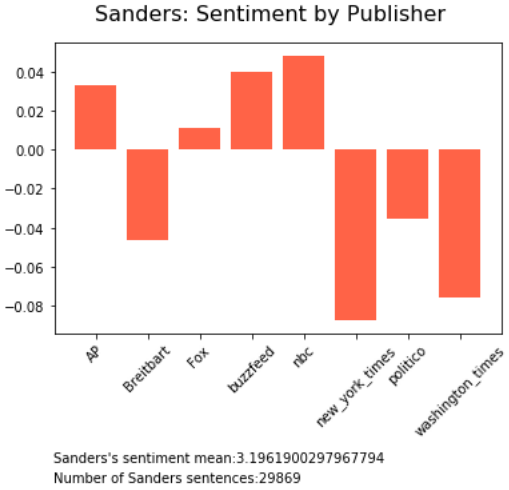
Figure 7. Sentiment by Publisher

B. Appendix B



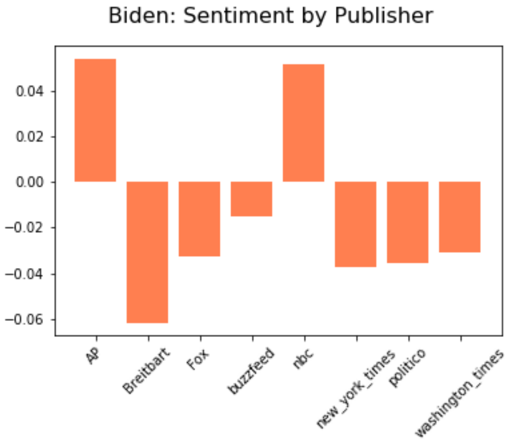
Trump's sentiment mean:3.0819611562003524
Number of Trump sentences:81197

Figure 8. Trump’s Sentiment Average by Publisher



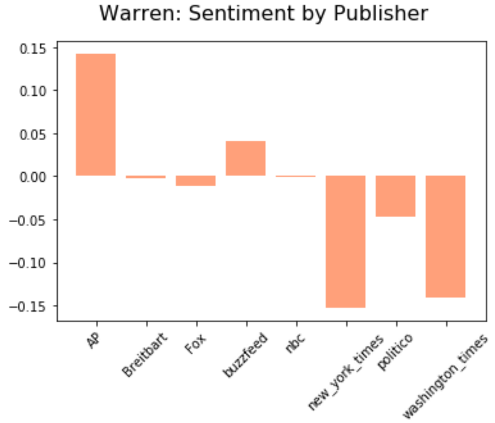
Sanders's sentiment mean:3.1961900297967794
Number of Sanders sentences:29869

Figure 9. Sanders’ Sentiment Average by Publisher



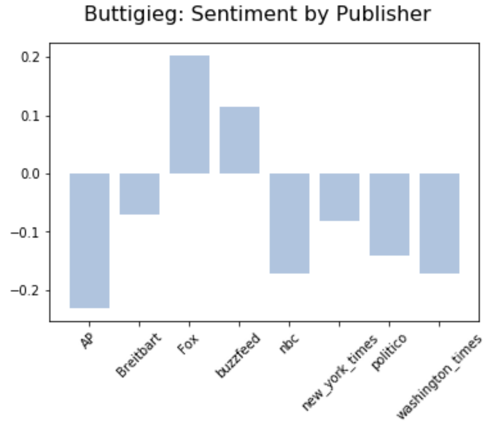
Biden's sentiment mean:3.153363827928252
Number of Biden sentences:38914

Figure 10. Biden’s Sentiment Average by Publisher



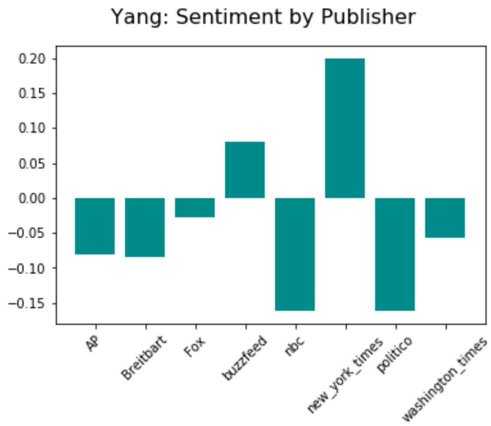
Warren's sentiment mean:3.141335636154289
Number of Warren sentences:3474

Figure 11. Warren’s Sentiment Average by Publisher



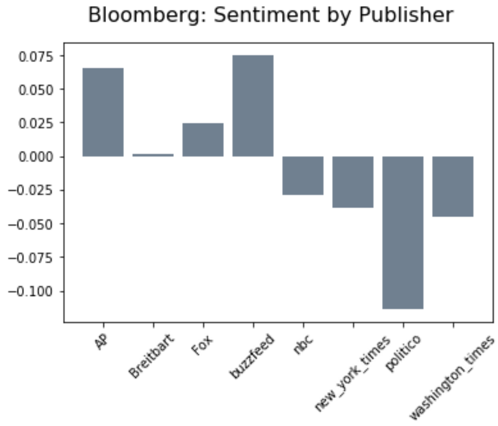
Buttigieg's sentiment mean:3.171401515151515
Number of Buttigieg sentences:1056

Figure 12. Buttigieg’s Sentiment Average by Publisher



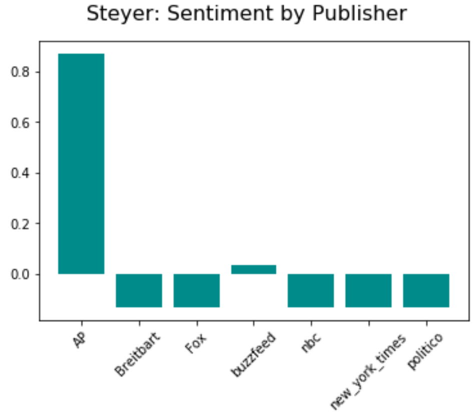
Yang's sentiment mean:3.162291169451074
Number of Yang sentences:838

Figure 13. Yang’s Sentiment Average by Publisher



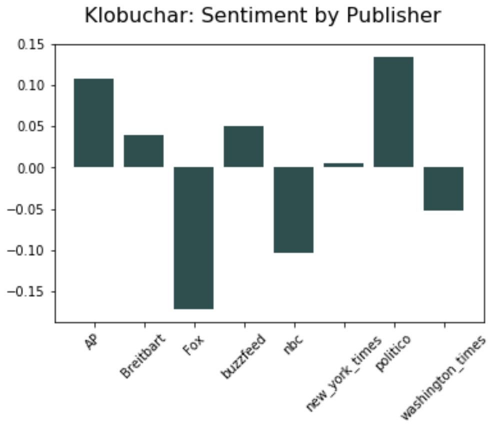
Bloomberg's sentiment mean:3.1540697674418605
Number of Bloomberg sentences:3784

Figure 14. Bloomberg’s Sentiment Average by Publisher



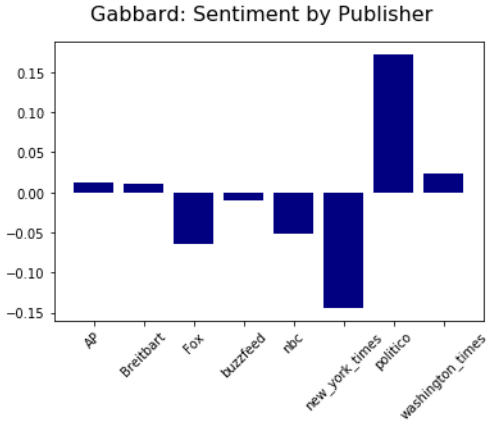
Steyer's sentiment mean:3.13265306122449
Number of Steyer sentences:98

Figure 15. Steyer’s Sentiment Average by Publisher



```
Klobuchar's sentiment mean:3.1273408239700373
Number of Klobuchar sentences:1869
```

Figure 16. Klobuchar’s Sentiment Average by Publisher



```
Gabbard's sentiment mean:3.143700787401575
Number of Gabbard sentences:1016
```

Figure 17. Gabbard’s Sentiment Average by Publisher