Project Proposal for Python and NLP

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

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

1. Abstract

A common concern voters in the United States express is the lack of (fair) media coverage of their preferred candidate. This has been particularly pronounced during the 2020 Democrat Presidential Primary election, with supporters of more progressive candidates (such as Bernie Sanders and Elizabeth Warren) alleging that their candidates' successes are consistently downplayed, or their campaigns downright ignored. Given the clear importance of media framing in influencing elections, this could potentially play a key role in influencing the outcome of the Democrat Primary.

To study this, our project scrapes at least five news websites representing the ideological spectrum (beginning with Breitbart, Fox News, the Wall Street Journal, the New York Times and Buzzfeed News), and processes the relevant political articles to determine 1) the level of coverage each candidate receives, 2) the sentiment of texts relating each candidate, and 3) which keywords are associated with each candidate. We plan to rely on a transfer learning approach combining BERT with CNNs to conduct sentence-level analyses of sentiment and keyword associations, and then to evaluate these relationships visually over time using a Flask dashboard. The web-scraping, modeling, and visualization process will be automated such that the dashboard can continue to play an observational and informational role throughout the remainder of the primary election, and perhaps even general election.

2. Introduction

The media plays a critical role in influencing political outcomes. In particular, the level of media coverage can have a direct impact on a candidate's name recognition, which is in turn related to their chances of electoral success [1]. Additionally, the way in which candidates are discussed in the news frames the electoral process, impacting perceptions of candidates' likability, electability, and even their political agendas.[2]

Both what the media decides to cover (agenda-setting) and how (framing) are inherently ideologically-driven decisions, which can tarnish the ideal of the news as neutral and unbiased. Indeed, news organizations - much the same as other businesses - must make profit-driven decisions, and cover events that align with their readership and the personal goals of those involved.

Because of the important role of the media and the inherent differences in coverage between outlets, there are often criticisms about a lack of fair or ethical coverage of political events. The two most recent Democratic Presidential Primary elections are examples of this: In 2016, many outlets were accused of not taking the Sanders campaign seriously at an early enough stage to allow him the necessary name recognition to compete with Clinton on an even playing field.[3] [4]

This allegation persists in the current 2020 primary race with the candidates receiving very different levels of coverage, independent of their current standings in the polls. For example, In These Times analyzed the coverage from MSNBC - a major liberal U.S. news outlet - in August and September 2019, finding that among Biden, Warren, and Sanders, Biden was covered twice as often as Warren and Sanders. Beyond total mentions, Biden also was the sole candidate mentioned in approximately 25% of articles, versus 5% and 1% that mentioned only Warren or Sanders, respectively, and Biden's coverage increased during the month of September, despite steadily decreasing in the polls at this time [5]. A November 2019 analysis from Politico found similar results, with Biden receiving three times the cable media coverage of Sanders and Warren, despite polling at a similar level [6].

The In These Times study also looked into the sentiment of the coverage, finding that Warren had the most positive, followed by Biden, and Sanders at the rear[5]. Sentiment analysis was also conducted by Storybench - a project of Northeastern University's School of Journalism - showing that between June 1 and September 28, 2019, media coverage of Buttigieg, Klobuchar, Sanders, and Warren was the most positive, while Gabbard, Booker, and Biden received the most negative coverage [7]. Earlier on in the campaign they noted a trend of female candidates receiving more negative coverage than the male contestants, though this trend did not persist throughout the primary [8].

For our project, we plan to expand on and improve these

findings. Our intention is to analyze the following over time:

1. The level of coverage for each candidate, measured by the number of articles where they are either 1) the sole candidate mentioned or 2) mentioned along with another candidate.
2. The sentiment of sentences in which they are the only candidate mentioned.
3. Which keywords are the most predictive of a particular candidate.

The intention is to see if we can isolate trends regarding the level of coverage, sentiment, or keywords, and see if they are associated with:

- The candidate's standing in the polls,
- More radical versus moderate candidates,
- Candidate gender,
- Or any other interesting patterns that may emerge.

3. Motivation

While this is a topic that has been extensively researched, most previous analyses rely primarily on simpler text analysis techniques (for example, the research from Storybench relies on sentiment analysis of unigrams using the Bing library), which may not adequately reflect the contextual nuance of each article. Additionally, prior analyses reflect only a discrete period of time, and do not allow for a time-series approach to uncovering trends.

From an academic development perspective, we also find this project to be useful by incorporating some more technical aspects. For example, we plan to automate the web-scraping and modeling process on an ongoing basis using a cron job, and to develop a Flask dashboard that reflects new findings in real time. Our idea is to create a tool that can be useful not only for retrospectively looking at how the media has behaved so far, but also as a way to monitor developments throughout the primary, and perhaps also the general election.

4. Evaluation

We would define success if, at the end of the project we:

- Scraped a representative amount of data.
- Automated the web-scraping process.
- Created an accurate (on the basis of minimizing Categorical Cross-Entropy Loss) and informative sentence-level classifier to uncover which words are the most predictive of a particular candidate.
- Created a dashboard to clearly display results.

5. Resources

Data will be acquired by scraping the Politics sections of at least five news outlets representing the ideological spectrum in the US: We are beginning with Breitbart, Fox News, the Wall Street Journal, the New York Times and Buzzfeed News, and have already set up web-scraping tools for all five sources [1] using the BeautifulSoup library.[2]

BeautifulSoup allows us to request the HTML content from a given link and work with an HTML parser to get specific items or text from a given page. Using this method, we can locate and extract all article links from the politics homepages. With each article link, we extract the article date, title, and full text. We can run the code several times per day, saving the article data into a .csv file in our GitHub data folder.[3] Currently this process requires us to manually run all lines of code in our script, but during this project we will set up the script to run automatically using a cron job.

In terms of methods, the first part of our work regarding the level of coverage of each candidate can be done using simple text analysis: Does this candidate's name appear in the text? If so, do other candidates also appear?

For the NLP tasks, we will use BERT (Bidirectional Encoder Representations from Transformers)[9] for word embeddings, to augment sentiment analysis and classification tasks with a transfer learning approach. 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 represents words based on both the previous and next context (thus, bidirectional).

BERT relies on Transformer, which is an attention mechanism that learns contextual relations between words (or subwords) 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 demonstrating consistently superior performance to other similar NLP algorithms.

We intend to implement BERT with a CNN to create a

[1] https://github.com/madelinebrady/Hertie-NLP-Python-Project
[2] Beautiful Soup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[3] https://github.com/madelinebrady/Hertie-NLP-Python-Project/tree/master/data

transfer learning model. We will conduct the analysis on sentence-level data where a candidate is mentioned, with the candidate name removed, and then attempt to predict (a) the sentence sentiment, and (b) which candidate the sentence is referring to.[4]

In our transfer learning model, we will use BERT as the first layer to learn the contextual relations within our sentences. As mentioned, BERT is pre-trained based on large amounts of data and has proven successful for smaller datasets in comparison to FastText or LSTM and Word Embeddings.[10]

The word embeddings from BERT can then be used to train a CNN to make our two predictions of interest. 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."[11] For tasks where feature detection in text is more important (i.e. angry terms, sadness), CNNs are superior, which is what we will be doing in our project with sentiment analysis and classification. RNNs are better at predicting sequences, which is not necessary in our project.

In terms of making sentence-level predictions using CNNs, we will use Yoon Kim's method of convolutional neural networks for sentence classification (2014).[12] Kim's paper showed the success of "unsupervised pre-training of word vectors" and a "simple CNN with one layer of convolution" when working at the sentence level. This process is described well by Roman Orac[13]:

1. Take a word embedding [n x m] (n = words in sentence, m = embedding length) from BERT.
2. Apply convolution operations of various n values (n = 2 words, 3 words, etc).
3. Apply Rectified Linear Unit (ReLU) to add the ability to model nonlinear problems.
4. Apply 1-max pooling to down-sample the input representation and to help prevent overfitting.
5. Concatenate vectors from previous operations to a single vector.
6. Add a dropout layer to deal with overfitting.
7. Apply a softmax function to distribute the probability between classes (i.e. sigmoid function).

Then we will use LIME (Local Interpretable Model-Agnostic Explanations) to uncover the keywords that were found to be most predictive of a particular candidate, and visualize our results in the dashboard. Ribeiro et. al. invented LIME in 2016 to better understand the reasoning behind black box predictions.[14] 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.[15]

As an alternative to the above, if it turns out that we do not have sufficient data for a competent CNN/transfer learning model, our backup plan is to use Professor Simon Munzert's news dataset to train a model that can predict outlet ideology based on newspaper articles, and test this model on our collected data. Professor Munzert's dataset includes over 1 million articles from 21 American news outlets throughout 2018. Although this prediction may not be as relevant to our overall question of interest, this alternative will allow us to test a deep learning method and still provide interesting results.

[4]The primary incentive for using sentence-level data is simply to ensure that we have adequate data quantity to complete this project.

6. Contributions

We intend to contribute evenly throughout the process, and both work on every step.

References

[1] Cindy D. Kam and Elizabeth J. Zechmeister. Name recognition and candidate support. *American Journal of Political Science*, 57(4):971–986, 2013.

[2] DHAVAN V. SHAH, DAVID DOMKE, and DANIEL B. WACKMAN. "to thine own self be true": Values, framing, and voter decision-making strategies. *Communication Research*, 23(5):509–560, 1996.

[3] Has the times dismissed bernie sanders? *The New York Times*, Sep 2015.

[4] Pre-primary news coverage of the 2016 presidential race: Trump's rise, sanders' emergence, clinton's struggle. *Shorenstein Center on Media, Politics and Public Policy*, Jun 2016.

[5] Branco Marcetic. Msnbc is the most influential network among liberals - and it's ignoring bernie sanders1. *In These Times*, Nov 2019.

[6] Ryan Heath and Beatrice Jin. Where 2020 democrats shine and stumblewhere 2020 democrats shine and stumble. *Politico*, Nov 2019.

[7] Alex Bajak. Gabbard, booker and biden get most negative media coverage over last four months. *Storybench*, Sep 2019.

[8] Alexander Frandsen and Aleszu Bajak. Women on the 2020 campaign trail are being treated more negatively by the media. *Storybench*, Mar 2019.

[9] 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.

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

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

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

[13] Roman Orac. Identifying hate speech with bert and cnn. *Medium*, Dec 2019.

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

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