Analyzing Political Candidate Speech Patterns

Christine Cook, Kyle Magida

Machine Learning for Public Policy

**Abstract:**

In this project, we used transcripts from the last forty years of presidential and vice-presidential debates to generate a measure of similarity in candidates’ speech patterns. We pulled the transcripts from the University of California Santa Barbara’s Presidency Project site and used regular expressions to build a catalogue of speech by candidate. With this information, we created features to describe language patterns, and used cosine distance to determine the similarity between each candidate. The various weights input into the distance formulas were calculated through iterative clustering with a goal of creating clusters that incorporated as many election years as possible. The result is a program that can accept one candidate’s name and return the ten most similar candidates, or accept two names and compare them based on the features we built.

**Introduction and Significance:**

Political rhetoric can be powerful and, at the same time, very subtle. With this project, we hope to reveal something about the nature of a candidate by comparing them without consideration of year, party, or gender. There are many similar projects out there. In fact, during the quarter, FiveThirtyEight[[1]](#footnote-1) came out with something extremely similar where they returned the top grams sorted by TFIDF for the 2016 Republican candidates. Our project differs from most because we’re using data from all candidates that participate in a debate back to the 1970’s. We also use more features than just a bag of words similarity. Our project has a wider scope and a more ambitious approach to determine speech similarities.

There is a reasonably large body of work on political speech surrounding partisanship and how politicians in different parties talk about varied issues. These analyses tend to focus on granular focus on words that can be features that indicate positions and different issues.[[2]](#footnote-2) Our project diverged from these approaches in that it was unsupervised and we wanted to dilute the influence of partisan language. However, it was helpful guidance to understand how TF-IDF analysis might highlight common issues being discussed and issue similarity over similar speaking patterns.[[3]](#footnote-3)

We’ve had fun with the project, and have been able to learn a little more about our current political candidates during our testing process. It is especially interesting to see how current politicians compare to historical figures and how politicians talk in different races. We’re hoping people will find this work as revealing as we have.

**Our Solution:**

First, we found all the debate transcripts online at the UCSB’s Presidency Project site. We used UrlLib and BeautifulSoup to pull the links to the debates, follow the links, and add the text to a dictionary in python. We also stored the name of the debate and the date/time for later reference. Next we extracted the names of the participants in each debate, including moderators, and added this information to the same debate dictionary.

The next step was to attribute text from each debate to the participants in each election cycle. This meant looping through the text of every debate and attributing each sentence to a candidate. With this information, we generated features meant to describe each candidate’s speech. These include lists of words and bigrams, average word length, and lexicographical diversity, all with and without “stop words”. These features can be thought of as a candidate’s “profile”.

With these profiles, we sought to determine a distance between each candidate. This was a challenge due to the diversity in data types. To determine distance between the bags of words, we used TFIDF and normalization. The TFIDF vectors were created excluding stopwords. The literature suggests that For the numeric features, we normalized the raw distance between all candidates and squared it. We thought of this process as a sort of weighted K-nearest-neighbor, with atypical and heterogeneous data types.

We created clusters using this data across time and party affiliation to make observations about the importance of time vs. party affiliation and found that time has a stronger effect than party in our model. We also created the interface described above where a user can input one or two names and receive information on similar candidates. The individual candidate profiles included the bigrams that candidate was most likely to use, speech characteristics such as word length and lexigraphic diversity and the other candidates that a speaker was the most and least like. Our comparison interface shows the speaker profile for each candidate along with the distance between the two and how similar each candidate is to the other as compared to the entire set of candidates.

**Tuning Model**

Once we had our pipeline set up to output distances, we needed to determine various constants that would go into the model. These parameters were the weights that were given to the different features and the number of ngrams that we included in the TF-IDF analysis. Since we were doing unsupervised learning we needed to come up with an evaluation metric to targeting with our weight adjustments to create final pairings that were both interesting and relevant.

In the first few passes we found that candidates from the same year tended to match most closely with candidates from the same year. While this wasn’t surprising in and of itself we didn't want to have as much of a bias to the year as we initially got. Through a cursory examination of the feature we surmised that the primary driver of the small intra-year distances was due to the different topics identified by TF-IDF. This bias was likely due to similar topics being discussed at different rates in different elections.

We decided to focus on creating weights that would provide the greatest dispersion of years within clusters. That is, we wanted to create clusters that would have the highest number of distinct years. We wrote a function that would calculate this number for each set of clusters. The same function was applied to parties as well but most clusters had both parties so that was less of an issue. We used a range of weights and TF-IDF minimums that attempted to skew the influence of different factors. The results ended up indicating that equal weights and a high TF-IDF floor provided the highest average of distinct years across multiple cluster numbers. We ended up giving all the variables equal weights and setting TF-IDF to include 3,000 features. It is likely that by increasing the features we swamped some of the signal from each individual year. The full results from the parameter sweep can be seen in the included iPython notebook.

**Results**

With the tuned weights we were then able to make our final clusters and build the distance matrix that can be used to do a full comparison. The analysis is more qualitative that quantitative as we are doing unsupervised learning. Based on our weights we did check to make sure that years weren’t the only defining characteristic of the distances. Through spot-checking we did see that while people tending to be close to others in their year, candidates from nearby years also showed up as fairly similar. This general pattern can be seen in the charts in the appendix. They show the clusters for our final model and are divided by time and party but do cross both party and time boundaries.

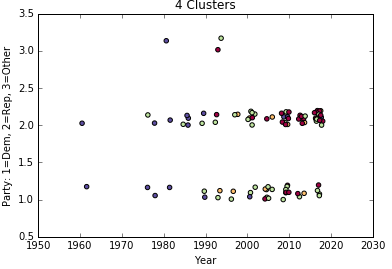
Visual inspection of the clusters and the final distance matrix (included as final\_distances.csv) shows clusters of Vice-Presidential debaters, modern GOP candidates, historical candidates in addition to more amorphous clusters. Additional, individual candidates that we have checked with our interface seem to have logical similarities based on the cluster characteristics described above. It is a challenge to properly analyze the type of results that we have produced but given a relative diversity of years in similar candidates and reasonable similar candidates we feel that our model has produced meaningful and informative findings.

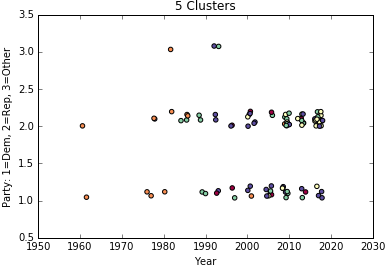
**Additional Analysis**

For this analysis we focused primarily on unsupervised learning on debate performances. A worthwhile extension of this work would be to formalize some of the clusters that we have found in this work and run a supervised model on them. A project like this could be useful in identifying historical analogues to rising politicians through the areas of policy they focus on and their speaking style. There would need to be more initial work done on making stable clusters but the ones we currently have appear to be relatively stable and descriptive. Clusters could also be built from candidates from a single year, eliminating year effects. This could make the predictive model more accurate and relevant to current politics but would eliminate interesting and informative historical comparisons.

The analysis that we did and the procedure proposed above could also be expanded through additional data. Our data source[[4]](#footnote-4) provides additional speeches including weekly addresses, inauguration speeches and State of the Unions. We made the intentional choice to not use this data as it limited the number of candidates and would introduce speech from different contexts. Further work could address these issues and provide interesting insights into how different politicians speak over time.

**Appendix:**

****

****

1. Beckman M. These Are The Phrases Each GOP Candidate Repeats Most. *FiveThirtyEight*. 2016. Available at: http://fivethirtyeight.com/features/these-are-the-phrases-each-gop-candidate-repeats-most/. Accessed March 18, 2016. [↑](#footnote-ref-1)
2. Grimmer, Justin, and Brandon M. Stewart. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political Analysis* 21.3 (2013): 267-297. [↑](#footnote-ref-2)
3. Monroe B, Colaresi M, Quinn K. Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict. *Political Analysis*. 2008;16(4):372-403. doi:10.1093/pan/mpn018. [↑](#footnote-ref-3)
4. *Presidencyucsbedu*. 2016. Available at: http://www.presidency.ucsb.edu/ws/index.php. Accessed March 18, 2016. [↑](#footnote-ref-4)