CHANGE IN SPEECH PATTERN OVER TIME:
PRESIDENTS GEORGE W. BUSH AND BARACK OBAMA

Winnie Yan
CUNY Graduate Center

Change in Speech Pattern Over Time: Presidents George W. Bush and Barack Obama

Winnie Yan
CUNY Graduate Center

ABSTRACT

As an extension of a previous project, this project aims to compute and compare changes in speech pattern over time of Presidents George W. Bush and Barack Obama, using their speeches from the American Rhetoric website. This was done using a 5-step procedure to scrape the speech transcripts, tokenize them, build language models, compute perplexities on the test data, and then compare the perplexities to determine changes in speech pattern between speeches, as time progressed. I used 50 older speeches from each president to train the language models, and 40 more recent speeches from each president, with each newer speech being evaluated with the language model. Resulting perplexities were mapped on a line graph to display changes over time. Analysis of the data revealed that it is inconclusive as to whether President Bush or President Obama exhibits greater changes in speech pattern over time.

Index Terms – Speech patterns, George W. Bush, Barack Obama

1. INTRODUCTION

This project is a continuation of my project from Methods in Computational Linguistics I, where I compared word frequencies of Presidents George W. Bush and Barack Obama's top 50 speeches to a third control corpus [5]. The speeches were gathered from American Rhetoric, an online speech bank of famous speeches, categorized by speaker or by category, including transcripts, audio, and video. Among the archive are the speech transcripts of Presidents George W. Bush and Barack Obama. President Bush's list consists of 100 speeches ranging from December 2000 to May 2020, and President Obama's list consists of over 450 speeches ranging from October 2002 to January 2017 [2, 1]. For my previous project, I developed a three-step procedure where I scraped the speech transcripts, tokenized and word-counted each president's set of speeches, and then compared the word frequencies.

For the current project, I decided to build language models with the speech corpora for Presidents Bush and Obama. Initially, my idea was to train a model on one president's speeches and then test on the other president's speeches. However, after consulting with Professor Michael Mandel and practicum leader Arundhati, I decided to evaluate each president's changes in speech pattern over time. I was unable to find any previous published studies comparing Presidents Bush and Obama's changes in speech pattern so my goal for this project is to compare the changes in speech pattern to see which president's speech patterns changed more.

To find out which president's speech patterns changed more over time, I developed a five-step procedure that was an extension from my first project. I can reuse the first two sections with a few adjustments to suit the new project. Instead of using the tokenized corpora to compare word frequencies, I will use it to train separate language models on each president's older speeches. With the language models, I can compute the perplexity on the test data of the more recent speeches. The final step for this project is to present and interpret the data gathered from the language model. Ultimately, analysis of the data will determine whether President Bush or President Obama exhibit a greater change in their speech patterns over time.

2. PROPOSED PROCEDURE

When I initially conceived of the project, I proposed the following 5-step procedure. As I progressed with the project, I expected that adjustments would need to be made within each step, but the overall steps remained. Obstacles, feedback, and subsequent adjustments are discussed in section 3.

2.1. Step 1: scrape speech transcripts

The first step is to scrape the speech transcripts from the website. In my previous project, I found GitHub user yimihua2013's code for scraping the American Rhetoric website and made changes to it for my project's needs [6]. For this project, I will make further changes to my edited code from the previous project, to scrape 100 speeches from each president [5]. At this step, I will decide which speech transcripts will be used for training data and which speech transcripts will be used for testing data.

2.2. Step 2: tokenize transcripts

After scraping the speech transcripts into individual .txt documents, I need to tokenize them to prepare for building the language model. Part 2 of my previous project was to word-tokenize and then word-count all the speeches [5]. For this project, I will remove the word-counter and update the tokenizer script to tokenize and casefold all the transcribed speeches. Part of the tokenizer script was provided by Arundhati during Practicum 5 [4].

2.3. Step 3: train trigram language model

With the transcriptions fully tokenized, I can use them to build my trigram language models. From lecture, we learned that trigrams have the lowest perplexity, and thus are the most efficient for building n-gram language models [5]. Using the designated older speeches, I will build one language model for each president. For building the language models, I have the option to either build in the terminal, as learned in Practicum 5, or to build in Python, as indicated in Analytics Vidhya's comprehensive guide to building a language model in Python [4, 3].

2.4. Step 4: compute perplexity on test data

The next step for my project is to use the language models built on the train data, to compute perplexity on the test data. The older speech transcripts will be allocated to the train data and the more recent speech transcripts will be used as the test data. The resulting perplexity indicates the change between speech pattern in the old speeches versus speech patter in the new speeches.

2.5. Step 5: manual comparison of perplexity

The final step of the project is to evaluate the results from the previous steps. The best way to present and interpret the results is visually, using a table or a graph. By interpreting the results, I will be able to determine which president exhibited greater changes in their speech pattern over time.

3. OBSTACLES IN EXPERIMENT

3.1. Updating previous code to fit current project

The first obstacle I encountered during this project was having to update code from my previous project to fit the current needs. This included expanding the scraping code to grab the designated speeches for train and test data from each president, tokenizing the train data as a single file, and tokenizing the test data as separate files. While this was not a large obstacle, it was time consuming to figure out which parts of the previous code were necessary for this project, which parts needed to be removed, and which parts need to be adjusted. Some of the difficulty came with figuring out how certain functions worked, and how they would be affected if any adjustments were made. Furthermore, the American Rhetoric website has updated since I last accessed it and I had to include headers to gain access for scraping the speeches.

While scraping the transcripts, I decided to incorporate the casefolding and tokenizing into the scraping process. This eliminated the need to have Step 2 as a separate step from Step 1.

3.2. Deciding how to allocate train/test data

Originally, I intended to train with one president's speeches and test with the other and vice versa, to compare overall speech patterns to each other. During a consultation with Arundhati, she recommended I split the individual speaker's speeches into train and test data to see how much the speaker changed over time, since it is a given that two speakers would likely have vastly differing speech patterns. Thus, I considered doing an 80 train/20 test split for each speaker.

However, I realized that President Bush's archive has 100 speeches spanning 20 years while President Obama's archive has over 450 speeches only spanning 15 years. With such an uneven distribution of speeches between the two speakers, I was not sure how to balance the data for my language model. Given that I only had 100 Bush speeches to work with, I decided that for each speaker, I would designate the older 80 speeches in one .txt file for train data and the 20 most recent speeches in another .txt file for test data. That way, I would have 100 speeches from each speaker, while also gathering speeches from a closer date range.

After presenting my model, Professor Mandel offered feedback regarding a change in the assignment of train/test data. By allocating more speeches as test data, I can map the differences between the older and newer speeches as time passes. Considering the feedback, I changed the 80/20 designation to a 50/50, with 50 older speeches in one .txt file as train data and 50 more recent speeches as separate .txt files for test data. Each test speech will be evaluated by the language model and plotted onto a graph to show change over time with every speech.

Unfortunately, not all 100 of President Bush's speeches were available for scraping, as some were pulled from a third-party website. Thus, I had to compromise with a 50/40 model of 50 older speeches for train data and 40 newer speeches for test data. With this 50/40 method, there could be a visual representation of change over time from the 40 speeches each evaluated for perplexity, whereas the previous 80/20 method would only produce a single perplexity result for each speaker. For President Bush's speeches, the train data ranged from December 2000 to June 2004 and the test data ranged from June 2004 to May 2020. For President Obama's speeches, the train data ranged from October 2002 to January 2009 and the test data ranged from April 2016 to January 2017.

3.3. Using Virtual Machine to build language model

A major technical error I encountered during this project was with using my Virtual Machine. I had to use a Virtual Machine running Linux Ubuntu on my home computer to use OpenGRM and Pynini for building the language model. Unfortunately, my Virtual Machine has limited device memory and the system automatically "kills" any commands I try to run in terminal when there is not enough memory.

To resolve this, I had to find alternative ways to build trigram language models. I tried increasing the memory allotted to my Virtual Machine and building the language model in Python instead of via the terminal. With the increased memory, my Virtual Machine was less likely to "kill" commands due to low memory.

While I was able to build the language model in terminal when I increased the VM's memory, I had difficulty accessing the language model to evaluate perplexity in the following step. Thus, I went back and built the language model in Python using NLTK so I could use the NLTK's perplexity function. With the NLTK trigram model, I was able to evaluate perplexity of the 40 newer speeches for each president. With the perplexities, I created a line graph and tables to show changes over time.

3.4. Implemented procedure

After considering all obstacles and feedback, I made the necessary adjustments to my five-step procedure.

Step 1 required further edits to the scraping code, to account for the new 50/40 designation of train/test data, with the 50 older speeches as train data and the more recent 40 speeches for test data. I created new directories for each President and scraped the 50 train speeches as a single file and the 40 test speeches as separate files.

Step 2 was incorporated into Step 1. I added a line to the text writing code where the speech transcripts were tokenized and casefolded while being scraped.

Step 3 required some trial-and-error with first building the language models in terminal, and then in Python. After building in Python, I was able to evaluate perplexities of my designated test speeches.

Step 4 required writing a for-loop to evaluate all the speeches in the directory. Then, I used the lm.perplexity function from the NLTK library to compute the perplexities of the speeches using the previously trained language model.

Step 5, I wrote the resulting numbers into an Excel file, creating a separate table for each President's speech perplexities. Then I created a line graph with both tables' data represented, for a visual representation of the speech pattern differences between the speeches.

4. RESULTS

For President George W. Bush's speech data, the perplexities ranged from 4.034 to 69.239, with a mean perplexity of 41.930 and a median perplexity of 47.750.

For President Barack Obama's speech data, the perplexities ranged from 11.812 to 68.992, with a mean perplexity of 44.091 and a median perplexity of 43.978.

	Min	Median	Max	Mean
President Bush	4.034	47.750	69.239	41.930
President Obama	11.812	43.978	68.992	44.091

Figure 1: table showing significant values from data

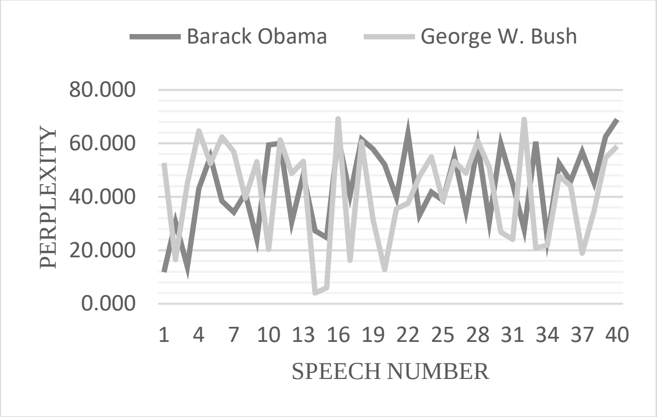

Figure 2: line graph showing perplexities of test speeches for each speaker

5. CONCLUSION

From the data gathered, it is difficult to concretely determine which president exhibited greater change in speech pattern over time. While President Obama's speeches showed a higher mean perplexity than President Bush's, the difference was not by much. The minimum, median, and maximum values are also fairly close between both data sets. Depending on evaluation, there are different ways to interpret which president's speech patterns changed more over time.

For the purposes of this project, a lower perplexity indicates greater change in speech pattern over time. If evaluating by overall lowest perplexity or by mean perplexity as the greatest change, President Bush has the lower values for both. However, if evaluating by overall highest perplexity or by median perplexity, President Obama has the lower values for both.

Since the perplexities of the two sets of speech data are fairly similar, I consider my results to be inconclusive. While I was able to successfully create language models and compute perplexities with the data available to me, there were still many limitations to my study, such as data availability, and technical difficulties. I believe my study would have yielded more conclusive results if some of these problems were addressed in advance.

Some considerations for further study include using a different speech database, comparing speech patterns differently, or implementing the project on a larger scale. The American Rhetoric speech bank has a limited archive for the purpose of my project. There are only 100 speeches for President Bush and while there are a lot of speeches for President Obama, they are within a limited time range. Many of the speeches available were short, with not much data to work with, and addressed very specific events at the time, thus almost guaranteeing low perplexity. Using a larger corpus or a different speech bank may yield different results, as the speeches would cover more topics, with more speech pattern examples from different speakers. With limited speech data like in my project, one could consider revisiting my original project idea of training a language model on one speaker and evaluating on speech data of the other speaker. There is an expectation of low perplexity because of comparing different speakers, but if comparing more than two speakers, one could find significance in evaluating the degree of difference between the multiple speakers. Another possibility would be to build the language model based on parts of speech, rather than tokens. For further study, I hope to implement this project on a larger scale using more speakers and larger speech corpora, spanning different speech genres.

REFERENCES

[1] American Rhetoric Online Speech Bank, "Barack Obama Speeches," https://www.americanrhetoric.com/barackobamaspeeches.htm

[2] American Rhetoric Online Speech Bank, "George W. Bush Speeches," https://www.americanrhetoric.com/gwbushspeeches.htm

[3] Analytics Vidhya, "A Comprehensive Guide to Build your own Language Model in Python!," https://medium.com/analytics-vidhya/a-comprehensive-guide-to-build-your-own-language-model-in-python-5141b3917d6d

[4] M. I. Mandel, Methods II Lecture Slides and Practicum Material, http://mr-pc.org/t/ling83800/

[5] W. Yan, "Word Frequency Comparisons between George W. Bush and Barack Obama's Top 50 Speeches," https://github.com/wyan3683/LING78100_Term_Project

[6] Y. Zhao, "Mining-the-Web_Python / ObamaSpeech.py," https://github.com/yimihua2013/Mining-the-Web_Python/blob/ master/ObamaSpeech.py