Final Capstone Report

Laura Friese

Understanding the linguistic and word choice differences between political parties can provide a lot of information of interest. One clear example of presidential speech is executive orders, which will be examined here. The tone of executive orders and the words chosen likely vary according to factors such as time period and the political party of the sitting president. This analysis will examine the relationship between these factors and the word choice of the executive orders. It would also be easy to examine the topics addressed in these executive orders, and see how they correlate to each political party. This could provide insights about the values of each party, as well as the typical persuasive strategies and rhetoric each party uses. We will also examine the use of these strategies over time, or for each president individually. This would inform the public of strategies used by each president or party and the use of those strategies over time. Knowledge of common rhetoric is a vital tool in the dismantlement of misinformation, which is so important in our current political climate of "fake news." Thinking critically about the tactics used to convey information, such as creating a sense of fear or duty, helps us better understand our political system.

To investigate the relationship between various linguistic elements of executive orders and the year, president, and political party in control, I will conduct some natural language processing on a dataset of executive orders given since 1994. This dataset is from the national register, via the Office of the Federal Register, and thus should be quite reliable. The data is available as a simple CSV, downloaded from their website. This would be a look into the

composite of all executive orders, and could hopefully provide some insight as to how executive orders are created, and whether orders given by one president or party differ significantly from the other.

I would like to use machine learning to see whether or not I can predict the executive order giver's party from the text of the executive order itself. Additionally, I will create a model that will try and predict whether the given executive order was written in the president's first or second term. If I can create a strong model, then there are likely some key predictive factors that differentiate the two.

To investigate the relationship between various linguistic elements of executive orders and the year, president, and political party in control, I will conduct some natural language processing on a dataset of executive orders given since 1994. This dataset is from the national register, via the Office of the Federal Register, and thus is quite reliable. Before analysis with NLP(Natural Language Processing) can begin, however, the data must be cleaned and wrangled, and some initial probing using statistical inference must be completed. The details of this process are outlined below.

The Executive Order dataset contained 928 Executive Orders, represented as rows, and 13 columns: the citation, the document number, the end page, the html url, the pdf url, the type (all were Presidential Documents), the subtype (Executive Order), the publication date, the signed date, the start page, the title, the disposition notes, and the executive order number. Most of these columns won't be used, but we will use the url and the publication date. The other numbers are largely referential.

This dataset didn’t contain the text of each order directly, only links to the PDF and HTML versions of the text. Thus, pulling the text from these URLs and attaching it to the DataFrame was the bulk of the data cleaning process. Before extracting the text from the URLs, I wanted to create another column called "President" and one called "Party" so that we have those pre-labeled before we make any transformations. I can see online that orders 13765-13851 were signed by Trump, 13490-13764 were signed by Obama, 13199-13487 were signed by Bush and 12890-13197 were signed by Clinton. This will help us assign each order to the president that gave it. This will be of use to differentiate the orders for later analysis. To create these columns, I made custom functions filling the new columns based on values in preexisting columns.

The next step was to extract the HTML text of the executive orders from the HTML links. This was done using Beautiful Soup and urllib to access the link information and then clean up the HTML results to make them more readable. After the text was added to the dataset, it was still in messy HTML format. To extract just the text of the order, I identified phrases at the end and beginning of the orders and used them to slice the text. First, I looked through the documents to try and find a phrase that all the executive orders started with. I used this to split the document so we only get the text of the order, not all the HTML information. I chose "Use the PDF linked in the document sidebar for the official electronic format". It's long, but I had issues with short phrases appearing earlier in the text. Below, I do the same with the end of the order and the phrase "[FR Doc.". This slicing, while not perfect, pared down the data so that only a few words from the HTML formatting remained. After creating the code to extract a single order, I then wrapped that code in a for loop so it would go down the entire dataset pulling the text for each order.

There was an issue with the splitting, however: many of the executive orders were actually not in HTML format. I discovered this when the for loop didn't work for every row in the dataset. I examined the links for one of the data points that didn't work, and found that if you click on the link in the actual dataset, it takes you to a page that says that the text of the executive order is not available in this format, and links you to another flat text page. To fix this, I removed all pages that did not have an HTML copy of the executive order, removing 223 of our 928 observances. This is a large portion and they were disproportionately taken from Clinton's executive orders, as it was mostly the older executive orders they weren't in HTML. This could be a source of error later, although we will later see that there ended up being a fairly even amount of Republican and Democratic executive orders after these were removed. The end result of this cleaning was 705 total executive orders to work with. To further clean the textual data, all of the "\n"s, denoting a new line, were removed so they wouldn't interfere with our results.

The next step in the analysis is to separate the data out by President and political party in order to make comparisons. After the data cleaning, the dataset contained 87 executive orders written by Trump, 272 written by Obama, 291 written by Bush, and 55 written by Clinton. This amounts to 327 executive orders written by Democrats and 378 by Republicans. Trump's orders were all in his first term. Bush and Obama had a relatively even number of orders in each term, with Obama having 144 orders in his first term and 128 in his second and Bush having 171 in his first term and 120 in his second (slightly less). The orders from Clinton in this dataset were all from his second term, as that's when the database began recording executive orders digitally.

After the data cleaning and wrangling were complete, I performed some basic statistical analysis on the data. As much of the data is text, not a lot could be done without using NLP. I created another column that contained the length of each document in characters, and compared this number across president and party. I found that on average Republican executive orders were 6545.6 characters long, and Democrat's executive orders were 7998.9 characters long. I then broke it down by President and found that Trump's executive orders were 8724.5 characters long on average, Obama's were 8187.8, Bush's were 5894.1 and Clinton's were 7064.6. The high number of Trump's was likely balanced out by Bush's low order length, especially as Bush has many more executive orders in this dataset.

After recording these basic statistics, some graphs were constructed to compare the executive orders. The box plots showed that executive orders written by Republicans were significantly shorter, however another plot also showed that Trump, a Republican, had the longest executive orders on average. This observation is explained by the low length of Bush's executive orders. As they make up the bulk of the Republican executive orders, these short observations lowered the Republican executive order length average. If Trump continues this pattern, the difference will likely become negligible. Our initial statistical observations also indicated an increase of executive order length over time, but this increase was so small that it is likely clinically insignificant, even if it is statistically significant. To test this, I did a linear regression of the relationship between time and the length of executive orders. The linear regression produced a model with an R-squared value of 0.447, which is a moderate association. However, the coefficient of the publication date (our time variable) was only 0.0098, which is quite small. This further reinforces the idea that, although this trend might be

statistically significant, the actual difference is not clinically significant, meaning that is doesn't matter in reality. This small difference indicates a change of only about 3 characters per year. We also created a model using only the non-extreme values of this dataset, which we chose based on the initial visualization to be below about 21000. This model produced an R-squared value of 0.667, which is a moderate association and larger than our previous value, indicating a stronger relationship. However, the coefficient of of the publication date (our time variable) was still only 0.0084, which is even smaller than it was previously. We can conclude that this relationship is likely not a useful one for our investigative purposes.

The next steps in this process will use NLP to analyze the linguistic associations with each party and president. This will hopefully reveal some information about the two parties or the terms of interest. In order to do this, we will use Regular Expressions from the re package to remove any extra punctuation or words that include numbers. This will clean our text into individual words so we can take out words that aren't significant and any punctuation, which doesn't carry meaning. In this case this largely refers to words that have numbers in them, as these are used for clerical reasons and don't carry any linguistic meaning in them. We also want to take the lower-case versions of each word, so that words with different capitalizations are not counted separately. After the data is cleaned in this way, we are ready to build predictive models based on the variables we created earlier.

To create the models, we use train test split, from the sklearn package, to split the data into a training set, with which to train the model, and a test set, which we will use to test the accuracy of the trained model. We use about a third of the data (33%) to test our model, as is standard. The model we are using will be TfidfVectorizer, as this model works well with very

sparse datasets. As the words in executive orders are so various, there will likely be many words that are only used a few times, which will create a very sparse dataset to work with. After this model is trained, the next step will be to fit a classifier to this training data, and use the classifier on the test data.

The classifier chosen for this project is the MultinomialNB (Naive Bayes) classifier, which is suitable for discrete variables (in this case, a word count). We instantiate the classifier and fit it according to the training data. We then use the classifier to make predictions based on the test data (aka the executive orders not used to train the model). These predictions are then compared to the actual values, which will either be the political party of the president that gave the order or the term that the order was given in, depending on what we're examining. This creates the initial model for each variable (party and term).

Now that the model is created, it's best practice to test several Alpha values for each model. The alpha values for the MultinomialNB (Naive Bayes) classifier acts as a smoothing parameter, and different values of this parameter will give us different accuracy scores. After testing several alpha values, we choose the alpha that gives us the highest accuracy score, and if there are multiple alpha values that give us the same score we choose the smallest alpha to reduce variance. This is the last step in model creation, and all that is left is to print the feature names and weights. This reports what features (or words) are indicative of the different levels of each variable. In other words, it tells us which words indicate whether an executive order was written by a democrat or a republican or whether they were written in the president's first or second term. The above processes were completed for our two variables of interest, and the results for each model are explained below.

In creating the model based on party using the above methodology, we found the model to be able to predict the party of the test executive orders with 65% accuracy. While this is not an extremely high statistic, it's still much higher than chance, indicating that there are some features or words that can be used as predictors for each party. We also looked at several potential alpha values (the constant for this model) and found that the best model resulted from an alpha value of 0.3, and created a slightly better model with an accuracy score of about 66.1%. We then looked at the individual features (words) to see which ones were the best predictors for each party. The topic of these words should give us some idea about what Democrats and Republicans are most likely to write about in their executive orders. The 10 most predictive words for each party are outlined in the table below:

Democrats	Republicans
aapi	secretary
aapis	act
abandoning	amended
abate	national
abduction	federal
abetted	agency
aboriginal	security
abortion	inserting
abortions	council
absentee	property

The first thing that jumps out from these results is that all the Democrat words start with A. This is because many words had the same weight, and thus the model sorts them alphabetically. This leads us to believe that Democrats had a large variability of words, and of

that corpus (group of words) no words were used disproportionately. Republicans, on the other hand had a lot more variability in the weights of the words. This means that those words in the Republican column were used disproportionately more than other commonly used words, which justifies the higher weight. For instance, Republicans tended to use “security” in their executive orders more than Democrats, so perhaps that’s a larger issue for their party, or at least it’s a more frequently referenced topic. Democrats, on the other hand, don’t particularly have any words or topics they use especially frequently, which is why the sorted list fell back on alphabetical order.

In creating the model based on term using the previously outlined methodology, we found the model to be able to predict the party of the test executive orders with 56% accuracy. This is not an extremely high statistic, as it’s much not higher than chance, indicating that there are few features or words that can be used as predictors for the two terms. We also looked at several potential alpha values (the constant for this model) and found that the best model resulted from an alpha value of 0, and created a slightly better model with an accuracy score of about 60.1%. This is an improvement, but it appears that the model cannot very accurately predict from what term the various executive orders come from. We then looked at the individual features (words) to see which ones were the best predictors for the terms. The topic of these words should give us some idea about what presidents are most likely to write about in their executive orders for each term. The 10 most predictive words for each term are outlined in the table below:

First Term	Second Term
____	secretary
_____	national
_____	act
_____	federal
aapi	property
aapis	agency
abbreviations	amended
abedinico	security
abetted	government
abiding	director

The blank lines in the first term were clerical pieces that were not properly removed from the corpus. The fact that they're the first terms in this list, as well as all the words starting with a, indicates that like with the previous words, this list was done in alphabetical order as no words were used more frequently. This is confirmed by examining the weights of these words, all of which are equal. It does not appear that there are any strong features that indicate whether or not an executive order was written in the first term. When we look at the words in the second term, we again see that there's more variability in the weights for each word. However, a lot of these words are very similar to those in the Republican column, which occupies the same position (aka an indicator of 1 instead of 0). This may mean that these words were just used ubiquitously in many orders, and simply carry a higher weight. This hypothesis is confirmed by the low predictive power of the model.

From our initial findings, it does not appear that we are able to create a good predictive model to determine the party or the term of the president based on the text of the executive order. We found which words were more common, and we were able to garner some

predictive power, but the highest we could achieve came from the party model with an accuracy of 66%. This is higher than chance, but is still not an incredibly reliable model. Additionally, the repeated high weight words in the term model indicate that our findings may have more to do with frequency used words and the distribution of the executive orders. There were slightly more orders written by Republicans (about 54% vs 46% by Democrats), so assigning more popular words to the Republican orders made for a more robust model even though it had little to do with the actual topics of the executive orders. The same can be said for the first term vs second term calculation, as 57% of executive orders were given in the first term compared to 43% in the second term.

It appears that the party of the President giving the executive order doesn't have a lot to do with the words chosen in the order itself, which it this surprising, and neither does the term. Overall, while our models have a slight predictive power we are unable to obtain any clinically significant features that would tell us something about the composition of the executive orders. Perhaps our political parties are not as polemic as we believe them to be, or at the very least some of that notion is created by the media and not by the actual legislation, such as executive orders, that they pass. The most variation found in this examination was the length of the executive orders, which seemed to vary by the verbosity of the president rather than align to any one political party. This is exemplified by the fact that Trump had on average the longest executive orders, while Bush by far had the shortest.

This analysis could be further improved by using a more robust dataset. We had to remove a large number of the executive orders that hadn't been digitized in the same format as the others, and that further limited the scope and applicability of this dataset. With more

historical data perhaps the model would have been much more accurate. Additionally, the examination over time would have been much more fruitful and the variation between individual presidents would have mattered less.