Nick Videtti

nvidetti@syr.edu

Assignment 8

Latent Dirichlet Allocation

Scikit Learn’s LatentDirichletAllocation, CountVectorizer, and TfidfVectorizer, and Matplotlib’s PyPlot

6/7/2023

| An Attempt at Making it Easier to be an Informed Voter | |
| --- | --- |
| **Introduction** | Government affects many aspects of life. Participating in elections is often referred to as a “civic duty”, meaning that many believe the opportunity of participating in elections is so great that it becomes a responsibility to take advantage of it. Elections can be for anything from Town Treasury to President of the United States, but many would consider every election equally as important. While the federal government affects everybody and may trickle down to local government, local government has a much larger and immediate effect on people and has an effect on the future of federal government, as it is often the pipeline that officials take to eventually get to the federal government. Regardless of how it is broken down, there are reasons to value any election.      That said, a vote must be an educated vote. Election results are intended to reflect the overall opinion of the population, and in order to do that, each individual vote must reflect the opinion of the voter. This makes it extremely important for everybody to be informed on their choices and on the government in general. However, it is essentially impossible to be completely informed on all officials at levels of government, but a good place to start is with Congress. Congress is a part of the federal government, but is the largest branch and is the closest branch to that of local government. Every state in the United States is represented in Congress and most of the local officials that transition to the federal government will find themselves in Congress. Being informed on Congress should give a good base knowledge on government everywhere and at every level.    For these reasons, this study will look to inform the public by educating on what is said in debates by the members of Congress. Specifically, this will look at the floor debates of the 110th Congress (House of Representatives Only) and find the most prevalent topics that were discussed throughout the debates while showing some of the words that were used most while discussing those topics. This should help to give an idea of what was discussed rather than how it was said and ensure the discussion topics as a whole are considered instead of the people discussing them. Given that, there will be additional breakdowns based on sex (male/female), political party (republican/democrat), and a combination of both. |
| **Analysis** | This analysis will use Latent Dirichlet Allocation to perform topic modeling on corpora from the floor debates of the 110th Congress. Corpora have been supplied in four different files: one for male democrats, one for female democrats, one for male republicans, and one for female republicans. While more than this are supplied, only the first 25 corpora found in each folder will be used for these analyses. There is on exception since there are only 18 corpora for female republicans, so there will only be 18 used for that group rather than 25. While only 93 may seem like a small sample size, it is actually still over 20% of all the corpora, which should still give a decent amount of data without running into high latency during model training. This will allow for more models to be trained in a reasonable amount of time, and will at the very least give a good trial run for using all available data in the future.  First, the user will need to have the main folder that contains the four sub-folders with the corpora downloaded and saved somewhere on their machine. Then, the user will be prompted to select the main folder in a file dialog. From there, the data will be read into the Python session and stored into a list. Each element in the list will be a list itself containing the text, the corresponding sex (MALE or FEMALE), and the corresponding political party (DEMOCRAT or REPUBLICAN).  This list will be used to create a Pandas DataFrame, but will first need to be cleaned, as the text from the file is in a semi-structured format similar to HTML format and is not yet in the optimal form for performing Latent Dirichlet Allocation. The text has the document name and the actual text of the document, which will need to be split from each. Once they are split up, the document text will need to be cleaned as well. This will include splitting the text into words, cleaning the words, then putting the cleaned words back together into a column called “CLEANED\_DOCUMENT”, which is what will be used for the Latent Dirichlet Allocation. The process of cleaning the words will include converting words to lowercase, removing NLTK’s English stopwords, removing phrases with numbers, removing phrases without letters, and stemming all words so that words like “listens”, and “listening” will be converted to words like “listen”. This will ensure that words are analyzed for their meaning rather than their context.  From there, a function to perform the Latent Dirichlet Allocation will be created in order to replicate the process across multiple datasets without redundancy in code. This function will have many parameters that allow a wide variety of options for which vectorizer is used, what the title of the resulting visualization is, the number of topics that will be found, and the number of words in each topic. Based on function parameters, the function will take a dataframe and a field in that dataframe containing text (in this case, the “CLEANED\_DOCUMENT” column of the cleaned DataFrame that was created) and create a vectorized DataFrame using the vectorizer specified in the function parameters, or the default count vectorizer. After that, a Latent Dirichlet Allocation model will be trained using the vectorized DataFrame to model the number of topics specified in the function parameters, or a default of 5. After that, a visualization created using Matplotlib’s PyPlot will be displayed showing the top words in each of the topics. The number of words in each topic is also a function parameter, but is defaulted to 15.  The function name, parameters, and defaults will be:  VectorizeLDA(dataframe, text\_column, n\_grams = 1, tfidf = False, boolean = False, vec\_in\_title = False, title = 'LDA Model Topics', LDA\_topics = 5, top\_words = 15)  Finally, this function will be applied 18 times, and the resulting visualizations will be utilized and interpreted from there. Nine of these 18 times will be using SciKit Learn’s CountVectorizer and the other 9 will use SciKit Learn’s TfidfVectorizer. The nine times for each of these vectorizers will be all data, all males, all females, all republicans, all democrats, all female democrats, all male democrats, all female republicans, and all male republicans. |
| **results** | The resulting visualizations of the analyses described in the previous section were as follows.  **Cleaned DataFrame - First Few Columns of First 5 Rows**    **Cleaned DataFrame - Last Couple Columns of First 5 Rows**                                          There are absolutely many differences and even a few similarities between the overall words and the topics of each of these visuals. It seems that the type of vectorizer used does indeed make a difference, so the best vectorizer would need to be determined to find the best possible results. An obvious next step would be to work with the entire dataset rather than the roughly 20% that was worked with in these analyses. Another interesting next step would be to name the topics that are modeled, whether that be by using the most common word in the topic or some other method. Parameter tuning to find different results would be the final part of creating different results to compare. Actual insights would likely require a deeper dive into these results using analytics rather than interpretation of these visuals alone. All of these suggestions should be considered in a future run-through of these analyses. |
| **conclusions** | After an initial look into the floor debates, it seems that there is a way to determine what was being talked about in terms of both topics and words in general, but not in a way that is very informative. There are certainly some similarities between the different sexes and political parties due to the fact that the same discussion is being had across sexes and political parties. However, and more importantly, there are many differences in the debate topics between the sexes and political parties.  Pertaining to the goals of this study, this can be a way to be somewhat informed of what is going on in Congress and government in general, but there is undoubtedly more research that needs to be done in order to be ready for making a truly educated selection in the next election. This exercise was intended to remove any bias and unnecessary context through simplification, but it was concluded that complexities and context are needed, almost by definition, to paint the full picture and give a much better idea of exactly how the Congress floor debates ensued.  Overall, it appears that a higher level of effort is still required to keep up with government news, which unfortunately brings another issue of finding a reliable and unbiased news source. One potential solution is to take in many different sources to avoid one single news source forcing their opinions on you, but that brings back the fundamental reason this study was done which is that it takes a lot of time and effort to do due diligence when it comes to being informed on local and federal government. Perhaps another solution is to focus more on the local government. It seems that most people do the opposite of this, but many politicians will state that local government is just as important, if not more. |