Homework 8

Kyle Walter

11/30/2021

# Introduction

Today’s world, text data, like all data is growing quite rapidly. Society communicates in text messages, sends emails, leaves reviews, and tweets about issues that are happening to them in real time to get corporate level relief. As Text data continues to grow, the challenge of searching through it and organizing it for various tasks becomes almost impossible. There has to be a better way to organize this information for the ability to do many tasks.

While twitter has continues to promote heavily the use of Hashtags for categorization, most text does not have this feature readily available to it. An example would be congressional data. During a session of congress thousands of speeches, bills, and other activities are carried out. Political Observers estimate the number of major topics passing through congress in a session to be about between 40 and 50. The fortunate thing is all of it captured in writing.

Can the major themes of a congressional session be caputured in a small subset of their were from a session?

# The Data

This review will focus on 429 speeches made during the 110th Congress in the House of Representatives. Keeping in mind that this is only a small portion.

The speeches were provided split by whether a male or female member provided the speech and the political party to which they belong. However; since the goal is topic modeling and not looking for features that indicate the party to which a person belongs, that information will not be utilized elsewhere in this analysis.

## The Laten Dirichlet Model

Laten Dirichlet Allocation model is an unsupervised topic modeler. Similar to k-means clustering, it requires a number for the number of groups which is the experiment part of the model. Like K-means it does select groups of documents to test and having enough iteration helps achieve better results.

The model then produces topic clusters and the major words from the topic. At that point, the process becomes a bit more are than science to determin what each topic represents.

## Analysis

As mentioned earlier the text data from 429 speech in congress were provided for this analysis. The speeches were read recursively into a python list using the os.walkpath function which built the corpora that was used in the model.

The next step was to define Tokenization and Vectorization. Sklean’s vectorization method will apply Tokenization directly. However; for the purpose reducing the number of features, a function used in prior assignments was used again. This applied the Natural Language Toolkit’s workpucnt tokenizer which splits of punctuation from words so that it can easily be eliminated and also then applies the wordnet lemmatizer to return words like buys and buying to the root form.

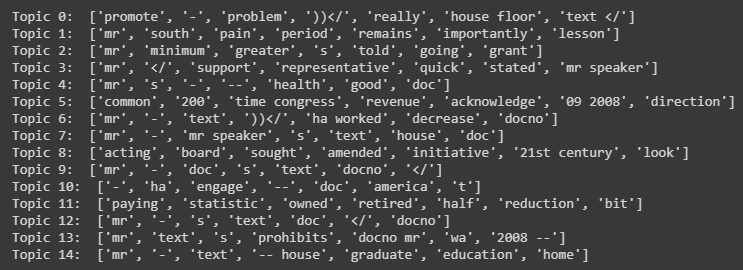
As for the vectorizer, since Topics or words of importance was keep search point, the Term Frequency Inverse Document Frequency (TFIDF) was used. The vectorizer looks for words that are less common among document and assigned them weights for the Corpora. This is helpful in weeding out words that are often used but hold little function. In addition the stopword remover was applied using the vectorizers built in list.

Lastly in order to reduce features sets even further words with low TFIDF ratings below 45% were removed from the Matrix all together. The other feature utilized was the ngram range which in the tokenization process determines how many words form a unit. For the first test both unigrams and bigrams were used. The vectorizer was adjusted many times through the process and will be noted below in the model runs.

## Model Run 1

For the first run of the model, the number of topics was defined at 15, using initially 10 iterations defined by the base form the model and online learning and an offset of 50. This returned almost no usable information as most the topics were the same set of words repeated over and over again.

This was adjusted to 100 iteration and the topics returned were actually different. Although some of the issues that presented were the tokens the model had picked up were often punctuation chains and document section markets like text, docnum, etc. In addition formal speech information used in the chamber were noticed such as mr, madam, and speaker. These were are not very useful forgetting to what the topic is.

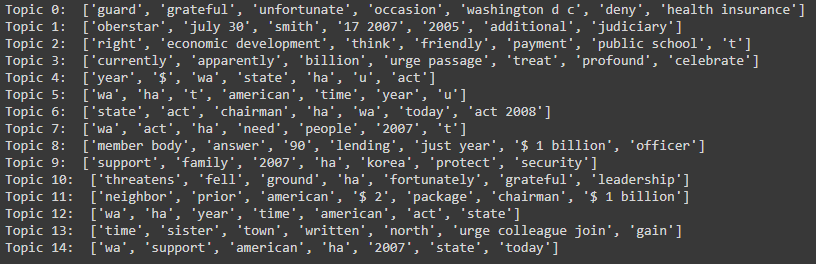


Sample Output from first model run

## Model Run 2

For the second run of the model, the stopwords list from sklearn’s vectorizer was updated to include many of the issues noted in model 1. In addition on later runs the ngrams range was expaned from bigrams up to quadragram (four word expressions). The thought behind this was legislation often has longer names. More recent examples such “Tax Cuts Jobs Act” and “US Mexico Canada Agreement” would likely be lost if similar names were in the speeches.

In addition to expanding ngrams the min document frequency was lowered from 45% to 33% as the initial results were not useful. This presented many topics that actually appeared to be defined, though some of them looked to be rather specific or themed similar to on another.



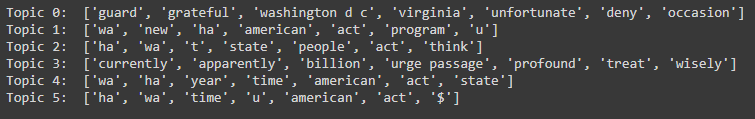
Sample output from the second model run

Only Trigrams were seen in the model’s 7 most effective words, so the expansion to quadragrams was useful.

## Model Run 3

Due to the smaller document set, a third model was run with a focus on topic reduction. As some of the topics at a glance appeared similar in nature in Model Run 2, the attempt was to see if reducing the number of topics for the model to interpret would yield more generalizable results.

The number of topics was reduced to 6. The goal was to see if topic variation around the Iraq War and Military would more or less consolidate. The results however did not support meaninful topic convergence as similar to the first run, many of the topics contain the same words as other topics.

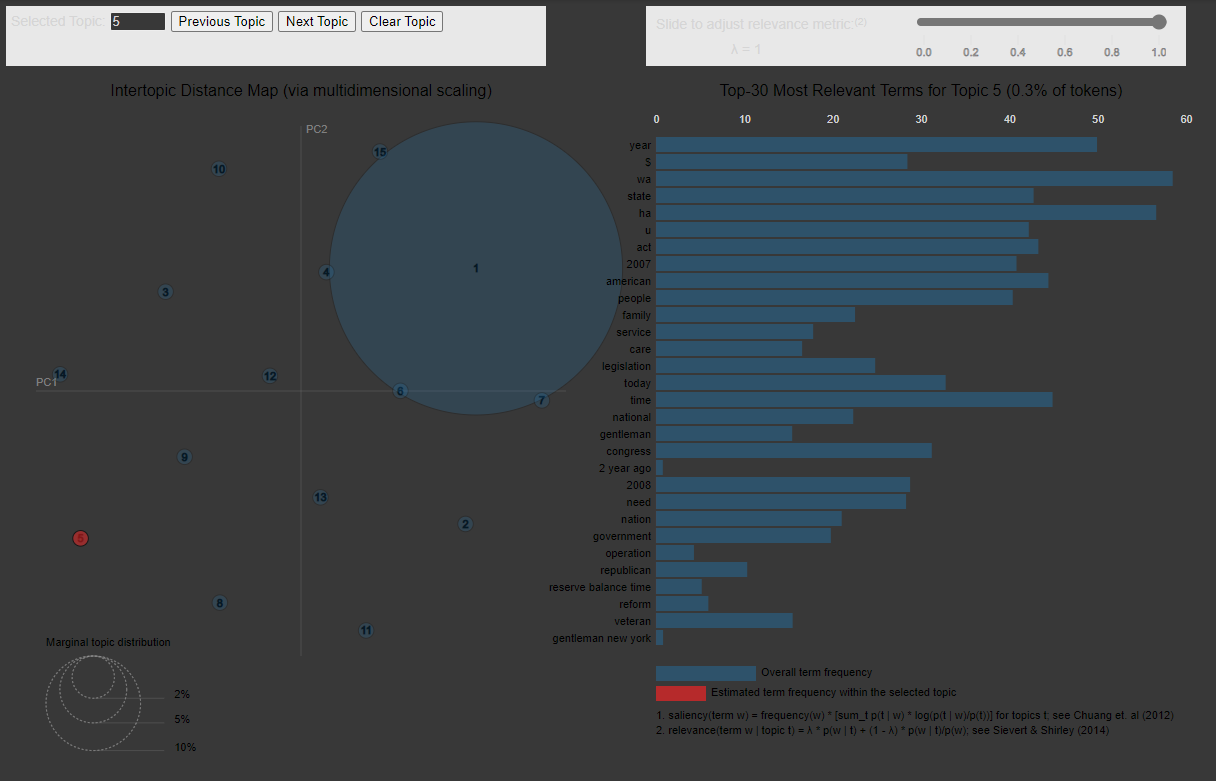


Sample output from third model run

## Model Section and Interpretation

Given the results of the three variation, the second model provides the most decernable topics. The next step is to determin what each of the 15 topics represents.

Python’s LDA Vis package was utilized. It shows the prevalence of the top 30 tokens in the that model has recognized in the topics.



LDA Visualization

It also show the distance between topics and the assurity of the words that help formed the topic. Similar to what was seen in Model three a number of words in the top 30 repeat. However; the 15 topics determined manually from the output are as follows:

1. 'Iraq War'  
2. 'Iraq War Funding'  
3. 'Energy Production'  
4. 'Military Deploymnet'  
5. 'Fomer Military Protections'  
6. 'Military Spousal Assistance'  
7. 'Religious Freedom'  
8. 'School Vouchers'   
9. 'Civilians in Iraq'  
10. 'Energy Cost Domestically'  
11. 'VA Funding'  
12. 'Refugees'  
13. 'Child Healthcare'  
14. 'Health Insurance'  
15. 'Resource Development'

Many of the topics appear as though they could be further consolidated, such as the Iraq related topics, Military Related Topics, and Energy related topics.

The Major topics from the 15 appear to be the Iraq War which is was at the height of controversy in 2006 during the 110th congress and returned Democrats control of congress in the election that year. Military likely also due to the ongoing war. Energy which is sourced from the area of the war. Healthcare for Children and for adults, some of the key words in 14 were age related so likely focused on Medicare costs for seniors and the VA is a military health care service as well.

# Conclusion

Topic modeling is a good feature for quickly searching through larger repositories of text for major themes. The sample in here was done in a few hours, but significantly less time than would be required to read through over 400 speeches.

The practical application as there is more text documents than can be read by an person in a lifetime is ability to search through text for related items and boil down what is needed for research or other purposes.