HW8

Topic Modeling

Bing-Je Wu

IST 736 – Text Mining

Table of Contents

[Introduction 2](#_Toc41127607)

[About the Data 2](#_Toc41127608)

[Data Processing 2](#_Toc41127609)

[Model 4](#_Toc41127610)

[Results 6](#_Toc41127611)

[Conclusion 7](#_Toc41127612)

[Reference 7](#_Toc41127613)

# Introduction



The United States Congress is the bicameral legislature of the federal government of the United States, and consists of two chambers: the House of Representatives and the Senate. The Congress meets in the United States Capitol in Washington, D.C. The members of the House of Representatives serve two-year terms representing the people of a single constituency. The number of congressional representatives can be vary depending on the census results. Each state, regardless of population or size, has two senators. The congress meet will be hold at least once each year. The congressional records provide transcripts of discussions and policy proposals. However, these records are often not organized in hierarchical structure based on topics. The ability to explore topic trends can enhance the understanding of Bills and Resolutions.

Analysis and Models  
The Latent Dirichlet Allocation (LDA) algorithm from Scikit-learn was implemented for topic modeling. A transformer, ‘CountVectorizer’ and custom stemmer were applied for tokenization.

## About the Data

The 110 Congress dataset consists of four subfolders, 110\_f\_d (female and democrat), 110-f-r (female and republican), 110-m-d (male and democrat), and 110-m-r (male and republican). Each folder contains different number of text files. The female and democrat folder, female and republican folder, male and democrat folder, and male and republican folder have 50, 18, 202, and 159 respectively, 429 documents in total. Each document is in XML representation.

Data Processing  
The dataset contains XML tags, such as <DOC> and <TEXT>. The useful text content is stored within the <TEXT> and </TEXT>. There are many ways to extract the text content. Here the regular expression method was implemented to extract text content from the raw string by using the user-defined function, ‘get\_text\_from\_tags’. Also, a lambda function was built to remove the file extension to store the filename only.

**Figure 1.** Sample of the raw 110\_baldwin\_x\_wi.txt

A screenshot of a cell phone

Description automatically generated

**Table 1**. *Dataset*

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data Type |
| filename | The name of the file | String |
| text | Text content of each file | String |

**Table 2.** *User defined functions for data pre-processing*

|  |  |  |
| --- | --- | --- |
| Function | Type | Explain |
| get\_text\_from\_tags | Defined function | Extract the text inside the <TEXT> and </TEXT> tags, remove '\n', '``' and "''" signs |
| filename | lambda | Remove the ‘110\_’ and ‘.txt’ from original filename string |

**Figure 2**. *Dataframe before pre-processing*

A screenshot of a newspaper

Description automatically generated

**Figure 3**. *Dataframe after pre-processing*

A screenshot of a social media post

Description automatically generated

In order to run an LDA model, a document-term-matrix is required. A custom tokenizer, ‘my\_tokenizer\_stemmer’ was set to the tokenizer option as the input of the ‘CountVectorizer’. The rest of the setting were ‘max\_df=0.95’, ‘min\_df=5’, ‘english stopwords’, word at least 3 letters with boundaries as the token pattern. There were 15,732 tokens in total.

**Table 3**. *User-defined functions for model building*

|  |  |
| --- | --- |
| Function | Explanation |
| my\_tokenizer\_stemmer | The function works as tokenizer and takes a string as its input. Then, it modifies the string in following steps:   1. Instantiates ‘PorterStemmer’ from ‘nltk.stem.porter’ 2. Removes every character except uppercase alphabet letters, lowercase alphabet letters and dash 3. Converts the string to lowercase 4. Returns a list of words after stemming |

Model  
An LDA model was built with 10 topics as its setting running in 100 iterations. The output of the model is a matrix that contains the weights of 429 documents relative to each of the 10 topics. The component matrix has the weight of 15,732 terms (tokens) relative to each of the 10 topics. Three types of visualization were shown on Figure 4, Figure 5 and Figure 6 to help analyze the result of model.

**Figure 4.** *Topic distribution for a sample document, ‘110\_brown\_x\_fl.txt’*A screenshot of a cell phone

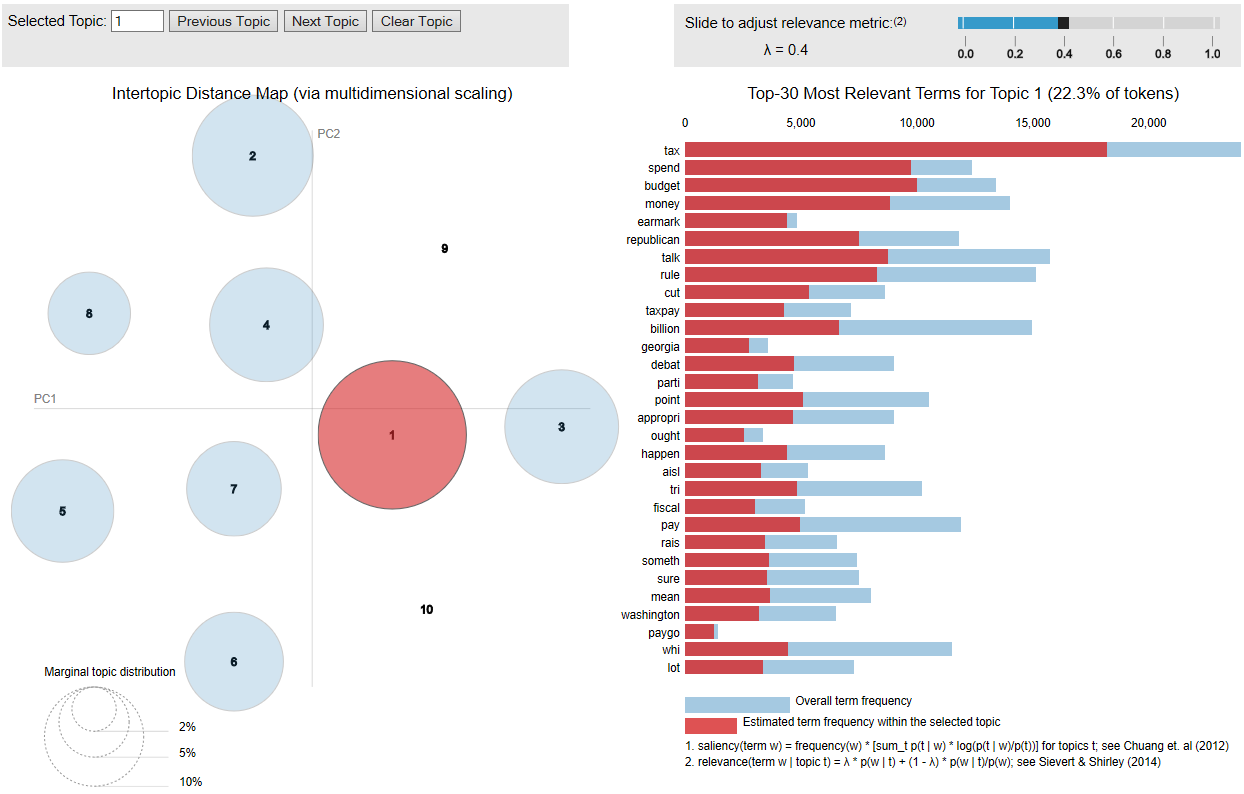
Description automatically generated

**Figure 5.** *Top 15 words for each of 10 topics*

A close up of a white wall

Description automatically generated

**Figure 6.** *Snapshot of the interactive visualization from pyLDAvis library*



# Results

The dataset was successfully modeled for 10 potential topics. The ground truth of the topics is unknow. It can only be deduced by the calculations of the distances between topics and words. From the Figure 5, the topic 1 could be talking about “health and education regarding the energy”. Topic 2 could be addressing the “border issues regarding the immigrant”. Topic 3 could be stating “government improvement project”. Topic 4 and 5 are similar. It could be surround “children’s health and education”. Topic 6 could be talking about “benefits regarding health, education, tax and insurance”. Topic 7 focus more on “tax and budget”, Topic 8 could be discussing “veterans in California”. Topic 9 could be addressing about “terrorist attack” and the topic 10 could be talking about “energy, gas and oil”.

Figure 6 shows the topic models in different representation. The circle on the left indicates the topic prevalence in the descending order. The number one means the most popular topic in the corpora and the number 10 is the least popular topic, among others (the topic numbers are not corresponding to the topic numbers in Figure 5). The distance between circles shows the similarity of each topic. However, it is only the approximation to the original topic similarity matrix because of two dimensions fashion. From the first glance, it seems like the 10 topics not far from each other. It indicates that each of topic somehow similar to at least one topic. There are two topics that are the least popular in the corpora. And the most popular topic, topic order number 1, has relevant terms, such as “tax”, “budget”, “money”, “spend”…etc. Those terms are relative to “tax and budget” which is not surprise for having congress to talk about money issue at higher priority.

# Conclusion

The 110th United States Congress was a meeting of the legislative branch of the United States federal government, between January 3, 2007, and January 3, 2009, during the last two years of the second term of President George W. Bush. The major event during that time was “Support for the Iraq War”. Thus, it explains few of the topics are talking about “benefit of veterans”, “military”, “border issues” and “terrorist attack”. Some other bills were proposed and discussed although some were not enacted during that time . Those bills were “Energy Independence and Security Act of 2007”, “Housing and Economic Recovery Act of 2008”, “Energy Improvement and Extension Act of 2008 “, “Tax Extenders and Alternative Minimum Tax Relief Act of 2008”, “Family and Consumer Choice Act of 2007”, “Medicare Prescription Drug Price Negotiation Act of 2007” and “State Children's Health Insurance Program”…etc. The LDA algorithm successfully help human explore topic trends and understand the session of Bills and Resolutions.

# Reference

United States Congress. [Web Link](https://en.wikipedia.org/wiki/United_States_Congress) ; 2 Topic modeling using Python and pyLDAvis. [Web Link](https://www.youtube.com/watch?v=SF50IK5XgKA)  
3 Hacking Scikit-Learn’s Vectorizers. [Web Link](https://towardsdatascience.com/hacking-scikit-learns-vectorizers-9ef26a7170af) ; 4 110th United States Congress. [Web Link](https://en.wikipedia.org/wiki/110th_United_States_Congress)