




## Selecting Dataset:

- Finding a good dataset requires a good amount of research into clean text datasets
- Using websites like [Kaggle](#), [Project Gutenberg](#), [20 Newsgroups](#), and others, clean text datasets like the ones shown below can be found

<u>A</u> 	<u>A Party</u> 	<u>A transcripts</u> 
Presidents	Political Affiliation	Transcripts
<b>44</b> unique values	Republican 43% Democratic 34% Other (10) 23%	<b>44</b> unique values
George Washington	Unaffiliated	Fellow Citizens of the Senate and the House of Representatives: Among the vicissitudes incident to l...
John Adams	Federalist	When it was first perceived, in early times, that no middle course for America remained between unli...

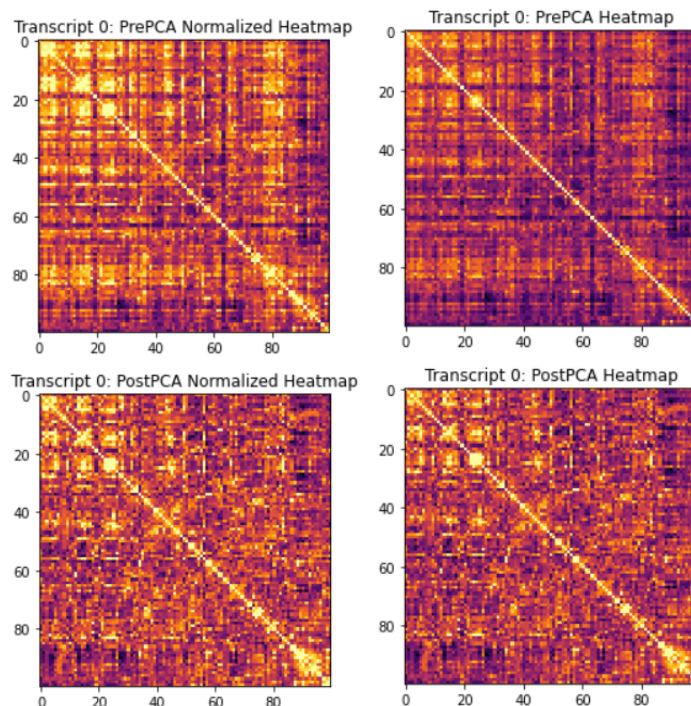
- As we aim to semantically cluster long text documents, datasets that encompass a large variety of ideas is required for optimal system performance
  - Examples of datasets that meet the length requirements are: non fiction books, long speeches, non dialogue format text documents.
  - Datasets to avoid: definitions, vague and shallow text.

# System Setup and Program Running

- Next, we need to input and read the database
  - Make sure that [Python version \(3.6+\)](#) is installed
  - Install necessary libraries:
    - [Pandas](#)
    - [Numpy](#)
    - [Matplotlib](#)
    - [SentenceTransformer](#)
    - [Nltk](#)
    - [Sklearn](#)
    - [Pytorch](#)
    - [String](#)
    - [Cleanlab](#)
  - Import Seg\_algorithm.py in order to use the segmentation algorithm.
- Ensure the program URL variable matches the selected dataset link.
  - Note: Saving dataset to Github and using Github link worked well for us.
- Run the main.py file using python
- File might take several minutes depending on how big the dataset is and the system being used

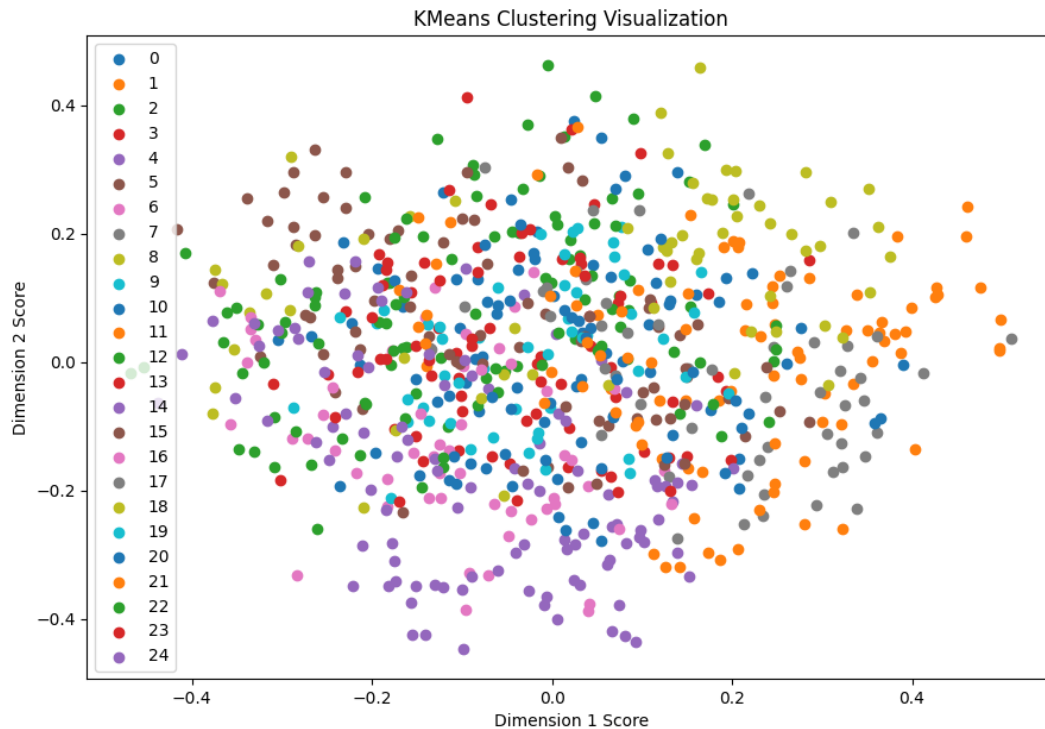
# Understanding Results

- There are multiple opportunities for data comparison throughout the program.
- There are outputs of segmentation, cosine similarity heatmaps, embeddings, filtered sentences, excluded sentences, plots of clusters of semantically similar and different ideas.
  - Examples: PrePCA and PostPCA of raw and normalized cosine similarity maps can be seen below. The data is also saved into a .csv matrix so exact number values can be used to further evaluate how the system identifies sentences.



Pre PCA						
	0	1	2	3	4	5
0	1	0.336	0.412	0.335	0.412	0.329
1	0.336	1	0.447	0.441	0.613	0.45
2	0.412	0.447	1	0.386	0.579	0.539
3	0.335	0.441	0.386	1	0.495	0.237
4	0.412	0.613	0.579	0.495	1	0.555
5	0.329	0.45	0.539	0.237	0.555	1

Post PCA						
	0	1	2	3	4	5
0	1.001	0.028	0.045	0.085	0.039	-0.052
1	0.028	1	0.075	0.222	0.367	0.121
2	0.045	0.075	1	0.054	0.208	0.184
3	0.085	0.222	0.054	1.001	0.228	-0.143
4	0.039	0.367	0.208	0.228	1	0.208
5	-0.052	0.121	0.184	-0.143	0.208	1



- Comparing segments of related sentences to the heatmap can be a method of evaluating how successful the program was
- Evaluating the excluded sentences by identifying if they are unrelated to the text can help evaluate progress
- Finally, evaluating Clusters of similar/different ideas can be helpful in evaluating the success of the program in semantic embedding of long text documents

## Troubleshooting Overview

### Result Optimizations

*Application is filtering useful sentences from corpus for computations or failing to filter non-useful sentences*

- There are several static values that can be set for application usage which determine how exactly sentences are filtered. If more/fewer sentences are determined to need filtered, these values can easily be modified.

*Computing results is taking too long*

- The analysis done by this application is extremely computation heavy and will likely take a significant amount of time to run. If the computer used for running the program does not have relatively strong performance, our team recommends running the program on Colab and have provided Colab files on the repository. Another option is to modify the number of documents being analyzed for the current run iteration.

*The number of clusters seems suboptimal*

- Methods within the code are provided to determine the optimal cluster size and may be used to benchmark the relative performance of the current amount of clusters that is being used. The number of clusters may then easily be modified if it is determined to be suboptimal.

*Heatmaps are difficult to read/interpret*

- Values within the code related to the color scale of the heatmap and numerical scale to determine the shading may be modified.

Critical Errors

*Application is failing to read input data*

- The input data read by the program is expected to be in the form of a CSV. Double-check the number of documents the program is expecting is not greater than the total number of documents within the imputed corpus.

*Application crashes during execution*

- The program is extremely computation-heavy and consumer-grade devices may experience difficulty running the code. Our team recommends running the program on Colab and have provided Colab files on the repository.