Building Cognitive Maps From Large Text Documents

Team: Tristan Weger, Yahya Emara, Ryan Rubadue
Advisor: Professor Ali Minai

University of Cincinnati
College of Engineering and Applied Science

ABSTRACT

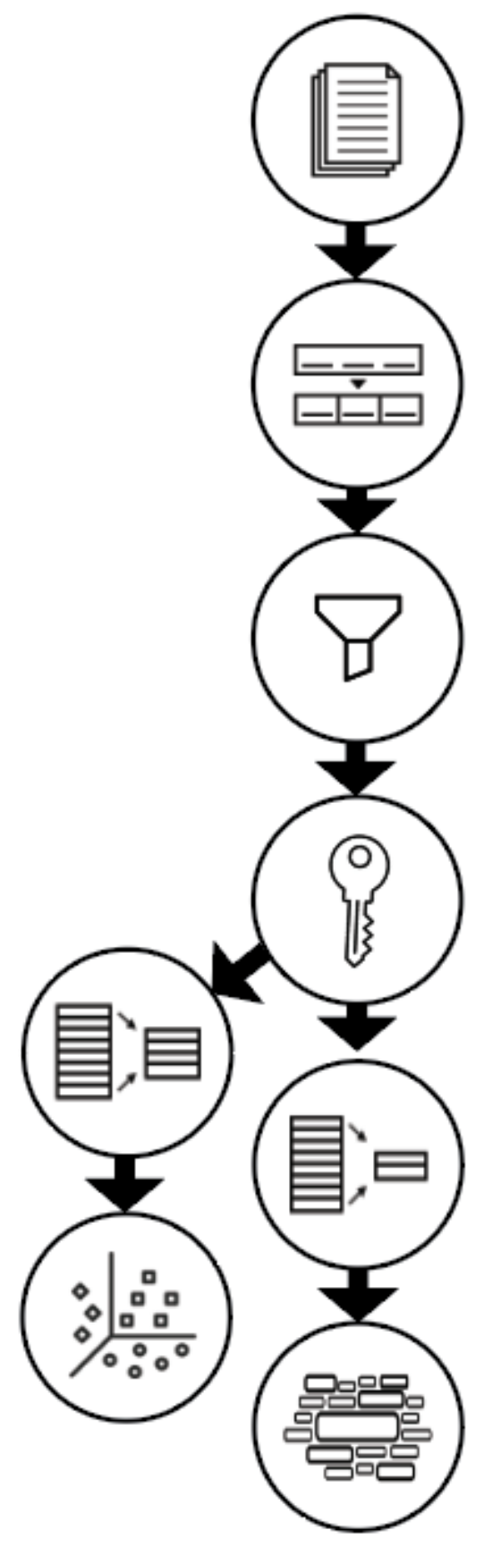
This project aimed to develop a natural language processing (NLP) algorithm using Python that could process large documents and create word clouds and cluster maps by embedding and clustering sentences. The algorithm uses various filtering parameters to remove irrelevant information, and the number of clusters is optimized for each parameter change. The effectiveness of different filtering parameters is then compared to select the most suitable approach. Overall, this project provides a powerful tool for text analysis and visualization, which could have applications in various domains.

OBJECTIVES

The main objectives of this project were to:

1. Tokenize the input text into sentences and embed the sentences as vectors.
2. Apply PCA to the embedded data to reduce its dimensionality while retaining as much information as possible.
3. Optimize the number of clusters for each case using coherence scores.
4. Compare different filtering methods and unfiltered data to determine their impact on the clustering results.
5. Visualize the output of the clustering process as cluster maps and word clouds to provide insights into the patterns and structures of the text data.

ALGORITHM



1. Input long text
2. Deconstruct text to sentence tokens.
3. Filter tokens to remove irrelevant data
4. Embed filtered tokens into high dimension semantic vectors (768 dimensions)
5. Reduce dimensions:
   1. 2 dimensions
   2. 95% variance is retained (~300 dimensions)
6. Visualize outputs:
   1. Word cloud
   2. Cognitive map

TECHNOLOGY

- Python
- UMAP
- Pandas
- NLTK
- Torch
- Sklearn
- SBERT
- Numpy
- Cleanlab

ANALYSIS

In order to achieve the highest quality clusters in our system, it is crucial to optimize the system parameters. The primary method of analysis and validation employed in our approach is the use of cluster coherence scores, which measure the degree of coherence among the topics of the clusters. By analyzing these metrics, we can identify the optimal system parameters that result in the most coherent and meaningful clusters.

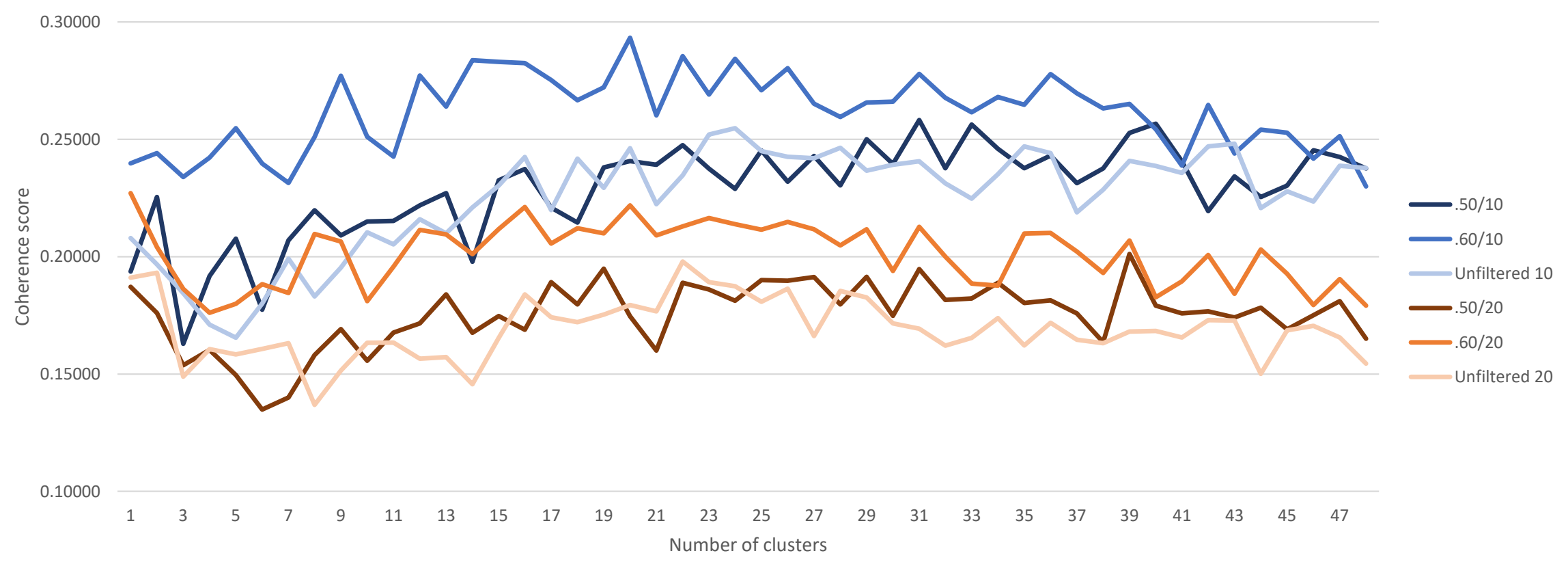


Figure 1: Plot of coherence score per n clusters for 6 data filtering parameter cases.

In Figure 1, we can see the optimal number of clusters for an NLP algorithm plotted against six different parameter settings. Each line represents a different parameter setting, and the x-axis shows the number of clusters. The y-axis represents the coherence score of the topic model. We can observe that for some parameter settings, the optimal number of clusters is lower, while for others, it is higher. By examining the coherence scores at different cluster sizes, we can choose the best parameter setting for our NLP algorithm, resulting in a more effective topic model.

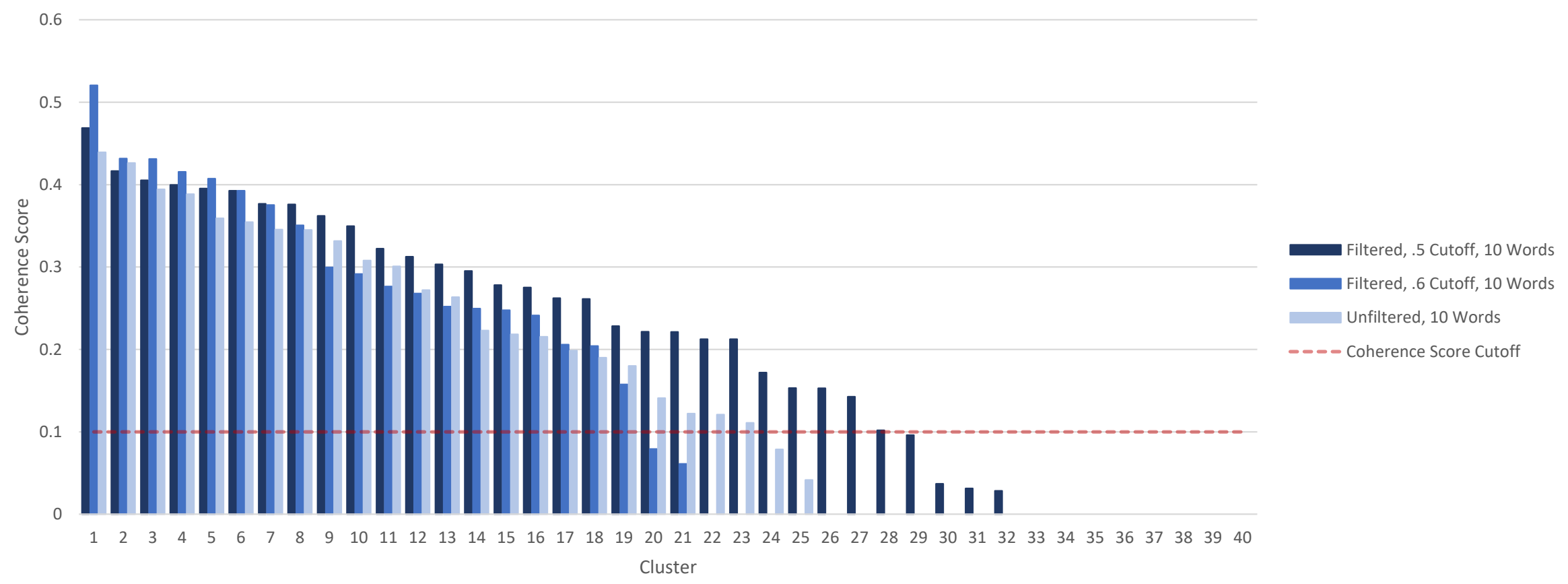


Figure 2: Plot of coherence score for each cluster for the optimal total number of clusters for 3, 10-word pair cases.

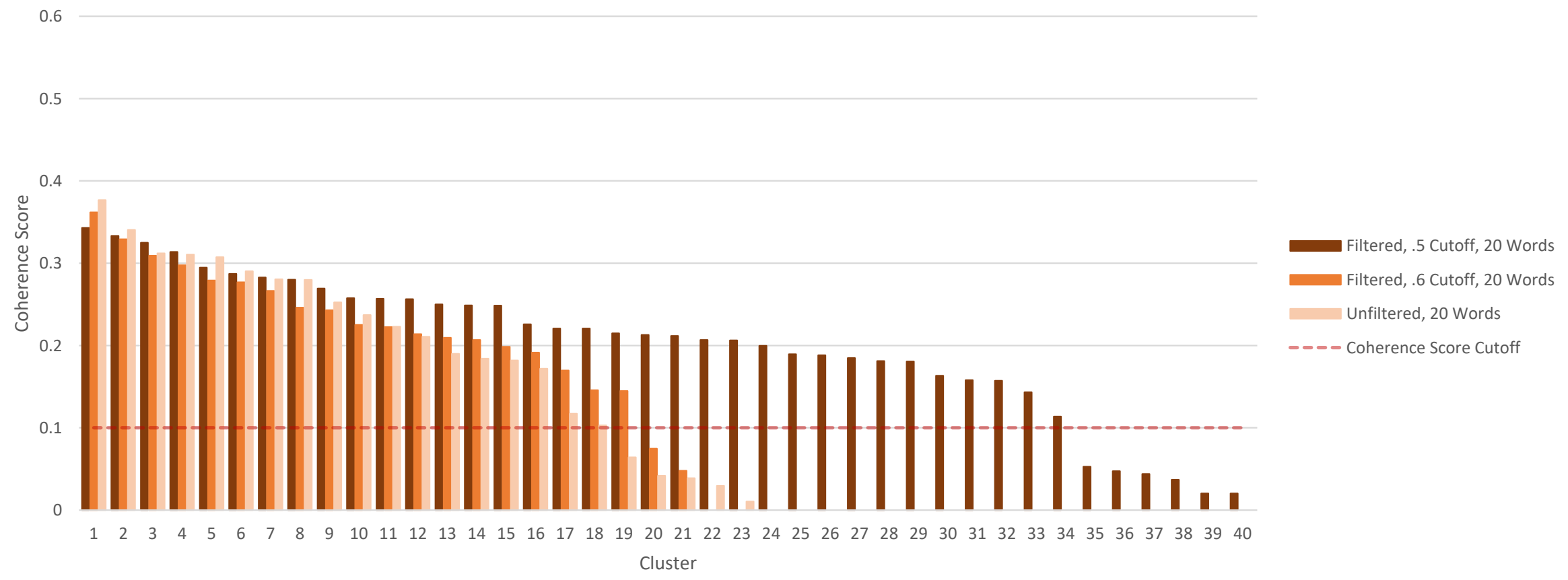


Figure 3: Plot of coherence score for each cluster for the optimal total number of clusters for 3, 20-word pair cases.

Figures 2 & 3 display the coherence scores of each cluster out of the optimal number of clusters for the six different parameter settings. The scores are plotted in descending order to show trends in the distributions along with the introduction of a cluster removal based on a cluster coherence score threshold below 0.1. These analysis methods can help us fine-tune our algorithm and achieve more accurate and informative topic modeling results.

RESULTS

Our algorithm employs a series of techniques, including sentence filtering, sentence embedding, and PCA95 reduction, to generate cognitive maps and word clouds that effectively identify and organize topics within the data. The results demonstrate that our approach is effective in identifying and grouping related topics within large text datasets, and the visualizations produced by the algorithm provide an intuitive representation of the underlying topics. These findings highlight the potential of our approach to facilitate efficient and effective analysis of unstructured text data.

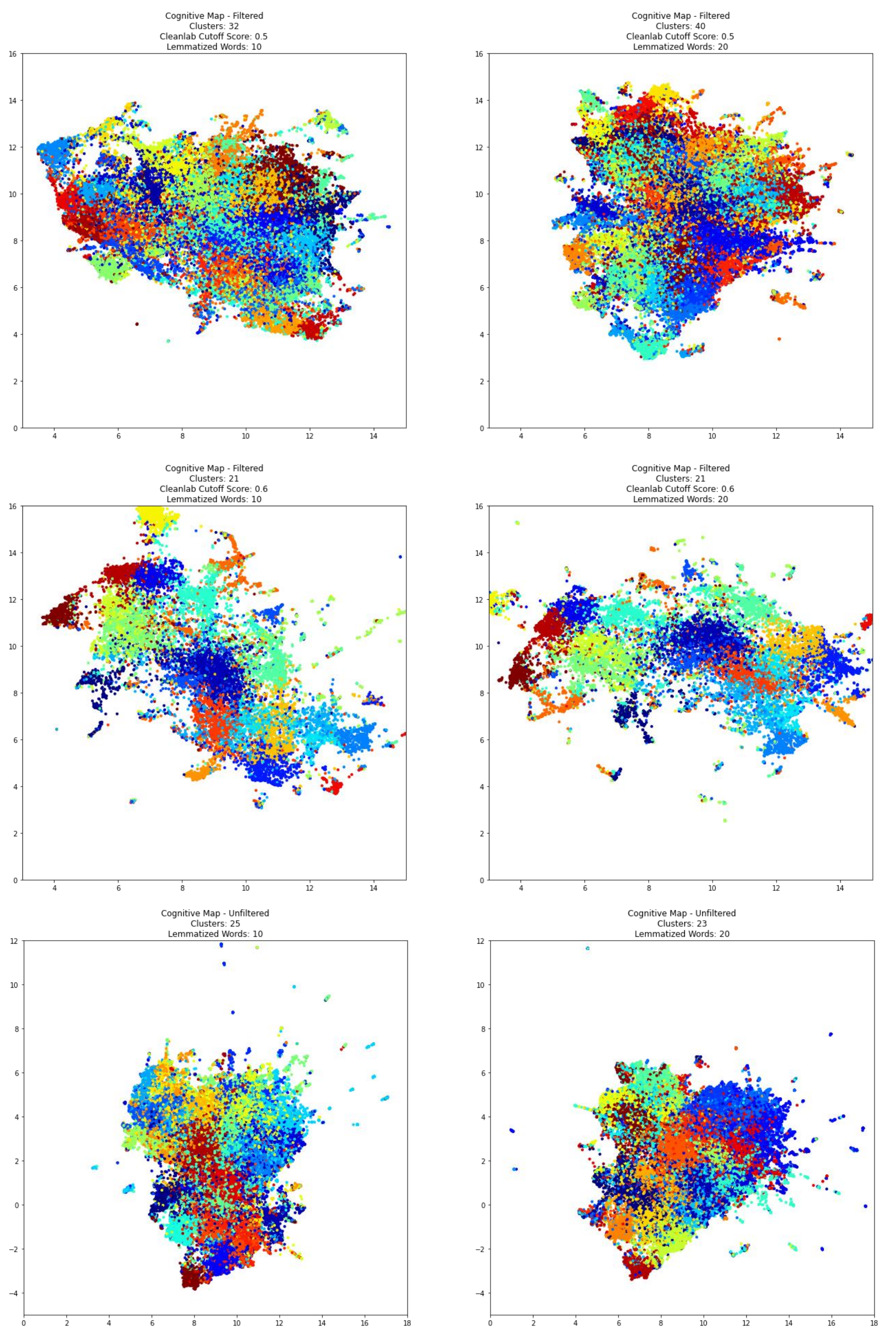


Figure 4: UMap cognitive maps for six different parameter settings. The optimal number of clusters for each case along with the parameters are listed in the plot titles.

UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique used in machine learning and data visualization. As seen in figure 4, a unique cognitive UMAP plot has been generated for each of the parameters based on the optimal number of clusters for our algorithm. Each plot represents a different parameter setting. The UMAP plots visualize the similarity and relationship between the clusters in a two-dimensional space. By examining these plots, we can gain insights into the structure and organization of the clusters and their underlying topics. These insights can help us better understand the corpus of text and refine our topic modeling approach. Overall, these cognitive maps serve as a tool for validating the effectiveness of our filtering techniques in in retaining similar topics.

RESULTS (cont.)

The word clouds generated from our NLP algorithm provide a visual representation of the quality of the filtering levels used in our model. By comparing the best, average, and worst word clouds, we can observe the differences in filtering efficacy, particularly in terms of word importance as determined by the algorithm. To verify the reliability of our results, we also tested our model on "The Origin of Species" and found that it effectively identified a diverse range of topics in the text, as evidenced by the generated word clouds. We also investigated the performance of the LDA topic modeling technique and found that the resulting word clouds were significantly less coherent than those produced by our algorithm. This suggests that our NLP approach is more effective in identifying coherent and meaningful topics in large text data.

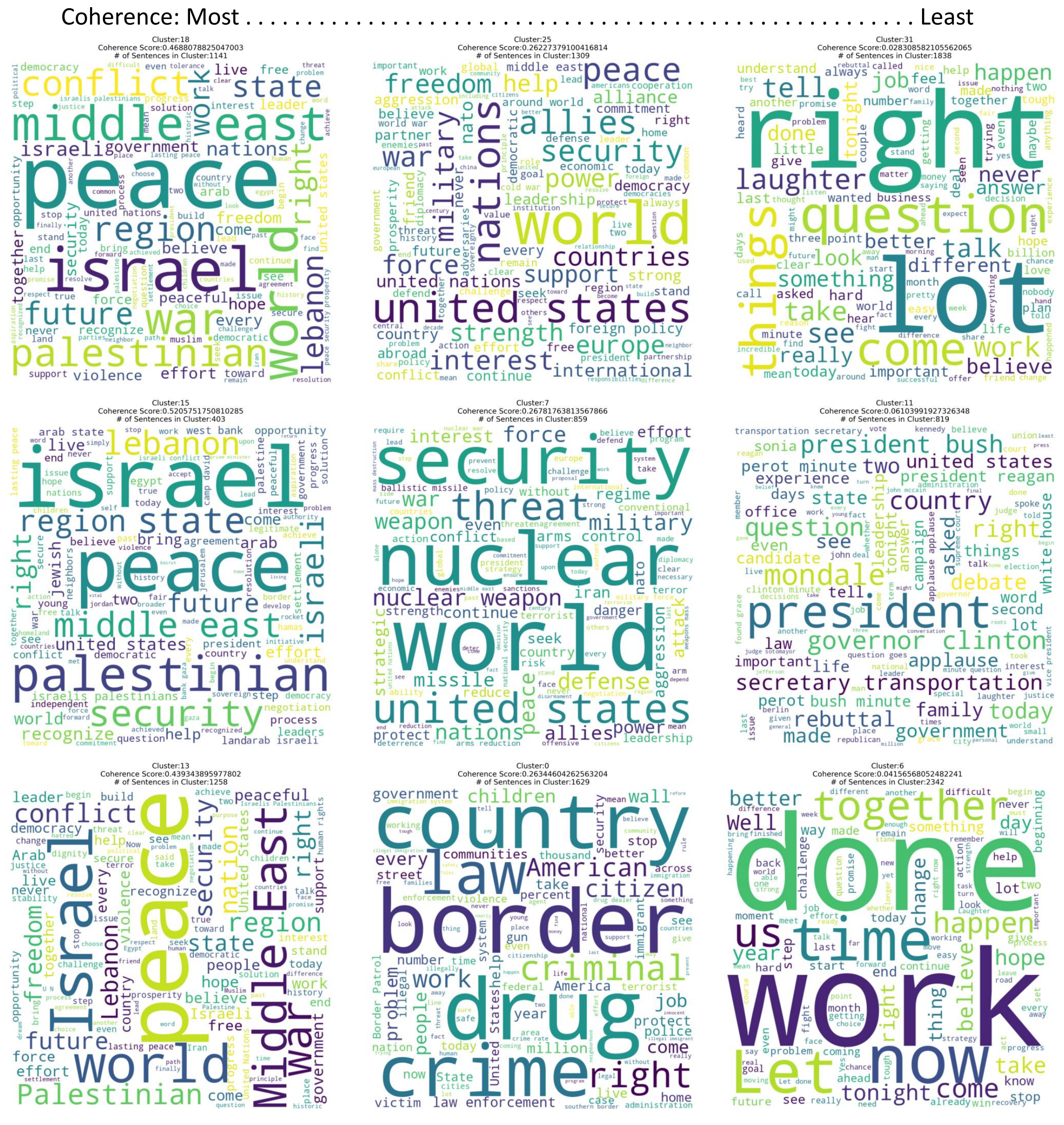


Figure 5: Words clouds of clusters. All are from 10-word pairs parameter. Top row is .5 cleanlab cutoff, middle row is .6 cutoff and bottom row is unfiltered.

CONCLUSION

The ability to break long text documents into clusters and display meaningful data is aided by using filtering techniques to remove non-useful text. Beyond simply comparing word cloud output several metrics like coherence scores and cluster visualization may be used to validate this improvement in performance.

The filtering and clustering techniques yield high quality results not limited to the presidential speeches dataset, but also for the secondary Origin of Species dataset. By testing across multiple datasets of different topics, we demonstrate the algorithm is not specific to one dataset.

Additional research topics related to our findings include summarizing the topics each president talked about and analyzing how the specific tone used for a topic changes over a timeframe.