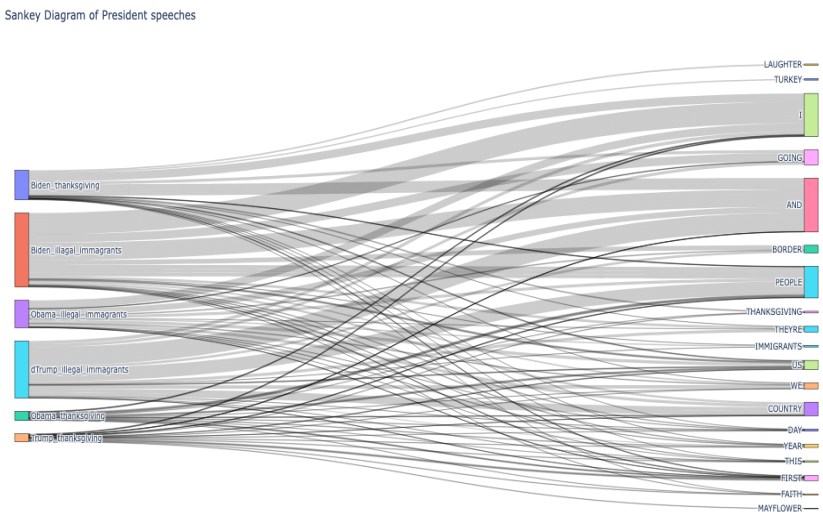
Group Member: Qixiang Jiang, Yidi, Wang, Jingkai Wang
Professor John Rachilin
DS3500 HW3 Report
Feb 27, 2023

Text File Processing:
Specifically, this step involves the usage of a famous natural language processing library, i.e., NLTK, which has a great number of built-in tools. For example, we didn't tokenize the raw text using our own code, instead, we invoked word_tokenize(), an API in NLTK, which simplifies this step. Also, the stop word list was not downloaded manually, instead, it was loaded with NLTK's support.

Furthermore, the generic file parser returns a dictionary where the keys are some useful statistics, such as word count in a file, average word length, and sentiment score of the content.
Word count is very straightforward. Average word length is calculated by total word length over the count of words. In terms of sentiment score, we have two word lists, the one being positive words and the other being negative words. While we are processing one file, the occurrence of a positive word contributes to a positive value, and the occurrence of a negative word contributes to a negative value. In this way, we can sum the total sentiment value of a file.

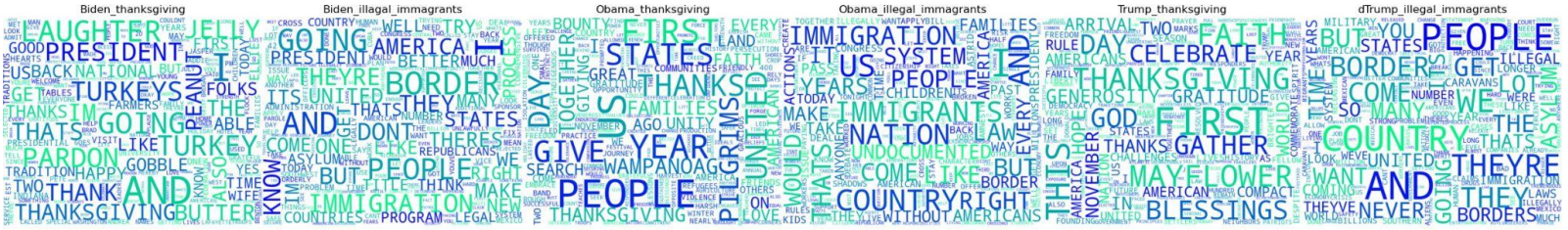
Sankey:
The Sankey diagram can help us visualize the relationships between the speeches and the most common 5 words in those speeches. The nodes are speeches and the most common 5 words across the speeches, and the link weights are the frequency of each word.

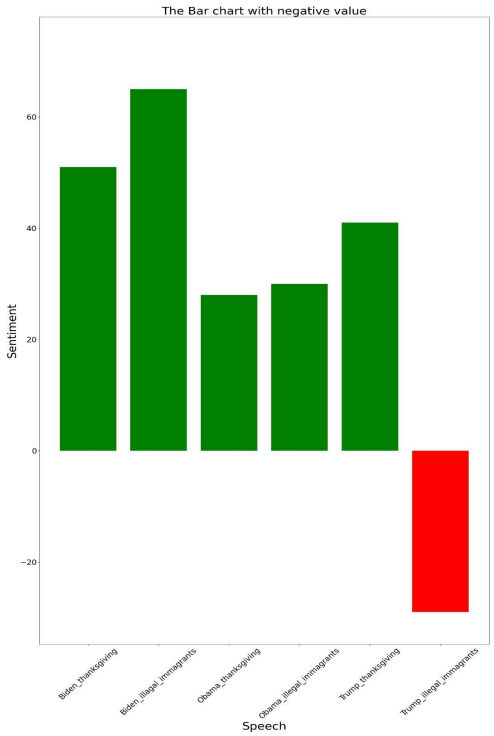


By using a Sankey diagram to visualize these relationships, we can get a better understanding of the connections between the speeches and the most common 5 words. For example, we would like to discover that certain speeches tend to use certain words more frequently or that there are common themes or patterns across the speeches.

Word Cloud:

To create the word cloud visualization, based on the processing for text words, the text is tagged using the NLTK library, and a list of individual words is obtained. Then, I used the Counter function from the collection library to calculate the frequency of each word in the list. This helped us to create a new dictionary where each word is a key, and its frequency is its corresponding value. Finally, I used the WordCloud function in the WordCloud library to create a word cloud visualization of words and their frequencies. The word cloud representation shows the most frequent words in the text data in a visually appealing way, with the size of each word proportional to its frequency. It is obvious that the three presidents all have more positive words, such as "thank," "celebrate," 'together," 'turkey," etc., and Obama uses more serious and formal words compared two other presidents. While speeches around illegal immigrants are relatively natural, even negative.





Bar Chart: The third visualization is a bar chart. Given that each president likely has a unique perspective on these events and issues, we calculate sentiment scores and use the matplotlib library to generate a bar graph that distinguishes negative and positive values with different colors. Higher green bar means more positive content in this speech, while red bar means it is a negative speech.

Contribution: Our goal is to divide the work equally among team members. To that end, we selected political speeches by the last three U.S. presidents on two specific topics. Jingkai Wang created the generic file parser of the words in the txt file, which not only reads and tokenizes the file content but also cleans the data, including removing unnecessary whitespace, punctuation, stopping words, and capitalization. Qixiang Jiang found the presidential speech we

used on Thanksgiving and illegal immigration on the White House website. Given that each president is likely to have a unique perspective on these events and issues, we calculated the sentiment score, and Yi di used the matplotlib library to generate a bar chart with different colors to separate negative and positive values. In addition, she changed the size of the x - and Y-axis characters to make the visualization more attractive. In addition, we intend to create a word cloud for our analysis. Jingkai Wang built word cloud visualization based on 'no_stopword_tokens' by using processed data. After that, the three of us created the Sankey diagram to visualize the relationship between the speech and the five most common words in those speeches. Once we decided on the type of text file and visualization, we set about implementing the necessary functionality.

Git Repo link: https://github.khoury.northeastern.edu/yidi/DS3500_HW3

Reference

The United States Government. (n.d.). *Speeches and remarks archives*. The White House. Retrieved February 27, 2023, from https://www.whitehouse.gov/briefing-room/speeches-remarks/