John Andrews
Dr. Wei Zhang
COP5537
5 December 2024

Project Report

Introduction

American presidential elections have far-reaching effects, shaping society, politics, and even world affairs. Wikipedia, as a platform designed to educate on a wide range of topics, provides detailed information on these elections, enabling readers to access, understand, and contextualize their significance (*Wikimedia Foundation*). The interconnected structure of Wikipedia articles offers a unique opportunity to analyze relationships between topics, transforming articles into data points that reveal deeper insights into how events and individuals are interwoven within broader historical and social contexts. However, while these data points highlight relationships between articles, they can be challenging for users who are not well-versed in the subject. Therefore, creating a visualization that reveals the active links between articles is essential for enhancing user comprehension and engagement while maintaining data points about the articles and links in question.

Dataset Information

For specific data points in Wikipedia, open-source projects and data collections provide tools that enable users to parse and utilize data for their own purposes. However, when dealing with more obscure or newly emerging data points, users often need to collect and curate the data themselves. For this project, I focused on gathering data from 14 major articles on United States presidential elections spanning from 1972 to 2024. These elections were chosen due to their size and complexity, as Wikipedia articles tend to become less detailed the further back in history they go. In the resulting graph, each election is represented as a distinct node, with links between articles forming edges that connect the nodes. This structure reflects the connectivity and associativity within the graph, based on how often a particular Wikipedia article is referenced within the 14 major articles. By organizing the data this way, the graph provides meaningful insights into the relationships between different elections, without creating an overly dense graph by including every possible link between articles.

To collect the data for each article, I developed a Python script using the *BeautifulSoup* module to scrape the 14 major articles. The script identified and extracted all the links (using the *href* attributes) present within each election article. Each link was then added to an adjacency list, where each node represented an article, and a link indicated that the specific article was referenced at least once within one of the major Wikipedia pages. To improve the quality of the data and include more meaningful nodes in the graph, I refined the data collection process to exclude redundant articles. For example, the web scraping script skipped any article containing the name of a U.S. state. This decision was based on the observation that many election articles include individual states' election results or general references to states. Applying this restriction across the 14 articles removed approximately 750 redundant nodes, most of which corresponded to state-level election results. Despite this filtering, the graph still contained significant redundancies. To address this, I imposed additional restrictions by excluding articles whose titles included certain keywords or phrases, such as: *covid-19 pandemic in*, *election*, *presidential*, *list of*, *presidency*, *primary*, *congressional district*, *district election, mayoral election*, *gubernatorial* *election*, *united states congress*, *governor, national convention*, *electoral history*, *state of the union address*, *bibliography*, *inauguration*, *special election*, *electoral college*, *administration.* These restrictions effectively eliminated redundant nodes that appeared in nearly every election article. One notable exception was *covid-19 pandemic*, which, as an outlier, added hundreds of nodes specific to the 2020 presidential election.

Objectives

For my dataset, I aim to accomplish two major objectives. First, I want to create a meaningful visualization of the graph that allows users to explore connections while viewing accompanying data. Second, I intend to answer the following key questions: *What are the most important nodes outside of the presidential elections? What are the major partitions within the graph, and do they reveal any significant distinctions?* Lastly, *Which nodes within the graph play the most critical roles in terms of overall accessibility, acting as bridges between different regions of the network, and exerting influence through their connections to other key nodes?* The first objective is purely about developing a web application that appears navigable and gives users freedom to explore the data. This will allow users to more leisurely explore the data and create inferences about the graph on their own. The second objective is answering questions in a more algorithmic way and will need various algorithms to answer those questions.

Methods

Let us now delve into each question and identify how we can approach each question and algorithmically find the answer. First, let us consider the question: *What are the most important nodes outside of the presidential elections?*. This question is fairly easy to answer as in essence we can run a *PageRank* algorithm on the nodes and edges within the graph, map each node with a corresponding value based on an arbitrary alpha (we will use .85 as that is fairly generic), then sort based on the corresponding page rank value. Let's consider the next question: *What are the major partitions with the graph, and do they reveal any significant distinctions?*. For this particular question there are a few different partitioning network algorithms that could be used such as *Louvain* (better for large-scale graphs), and *Spectral Cluster* (better for smaller graph or those with well-defined clusters), and *K-Means* (better for graph with complex structures and attributes) (*Memgraph*). Given that our graph is somewhat large and there are ample implementations of this algorithm, we will use the *Louvain* partitioning algorithm to cluster the nodes within the graphs into an n amount of partitions. As for the last question: *Which nodes within the graph play the most critical roles in terms of overall accessibility, acting as bridges between different regions of the network, and exerting influence through their connections to other key nodes?* This question can be slightly interpreted in a few different ways; however, we can use different measurements of centrality and determine which nodes have higher values in differing centralities to determine the overall structure of organization of connections of nodes.

To create a meaningful visualization of the graph and allow users to explore connections, I developed a web application using React. This application displays articles as nodes and represents edges as connections, indicating whether an article is referenced in one of the major election articles. Utilizing the *react-force-graph-3d* library, the application renders nodes and edges in a 3D interactive environment, enabling users to zoom in and out, move nodes freely, interact with the graph, and view node data (*Vasturiano GitHub*). Users can click on a node to be redirected to its corresponding Wikipedia article, hover over nodes to display detailed information, and highlight edges connected to specific nodes. The data for this visualization is dynamically generated by a Python script that scrapes Wikipedia, creating a list of nodes with attributes such as unique node ID, article name, *PageRank* value, overall *PageRank* ranking, closeness centrality value, closeness centrality ranking, betweenness centrality ranking, eigenvector centrality value, eigenvector centrality ranking, number of connected edges, and partition ID (indicating cluster membership). Nodes are color-coded based on their partitions, providing a clear visual distinction between clusters, and their sizes are determined logarithmically by the number of edges they connect to—larger nodes represent more connections, while smaller nodes indicate fewer. This approach provides an engaging, appealing, and informative way for users to analyze and understand the relationships between articles.

Results

The results from the algorithmic analysis provided several clear insights into the data. Excluding the presidential election article nodes (which, as expected, ranked in the top 14) and the restricted nodes, the top 10 nodes identified as most important were *Brokered convention*, *The New York Times*, *Socialist Workers Party (United States)*, *Tipping-point state*, *United States Senate*, *Prohibition Party*, *October surprise*, *Republican Party (United States)*, *Swing state*, and *Convention bounce*. These topics hold significant relevance in the context of U.S. presidential elections; however, some results yield interesting implications. Firstly, the prominence of the *Socialist Workers Party* and *Prohibition Party* over both the *Republican Party* and *Democratic Party*—the two major political parties in the United States—is noteworthy. This discrepancy could be attributed to the web scraper including all available links, regardless of whether they are explicitly visible on the webpage or embedded in the article's metadata. Since Wikipedia articles are continually edited and updated, the inclusion of certain less relevant links may have skewed the rankings. Secondly, the inclusion of the *United States Senate* as a highly important node is intriguing. Many Senate candidates are elected alongside presidential candidates, and the Senate's composition often directly impacts the ability of the sitting president to advance their legislative agenda. In contrast, races for the House of Representatives, as a larger body, may be viewed more holistically, with less emphasis on individual candidates. Finally, terms such as *Tipping-point state*, *October surprise*, *Swing state*, and *Convention bounce* are common phrases associated with U.S. presidential campaigns. Their appearance among the most important nodes aligns with expectations, as these concepts play a critical role in shaping election strategies and outcomes during the course of a candidate's campaign.

The partitioning within the graph yields some intriguing insights. Using the *Louvain* method of graph partitioning, the algorithm demonstrated a surprising correlation between the length of presidential terms and incumbent presidencies. From 1972 to 2024, nine individuals were elected to the presidency, and the *Louvain* method divided the graph into seven distinct partitions. Remarkably, five of these partitions correctly grouped nodes corresponding to presidential winners. The sole exception were the partition encompassing the 1976 through 1996 elections, which included the presidencies of Richard Nixon, Gerald Ford, Jimmy Carter, Ronald Reagan, and George H. W. Bush—all acting as president within a fourteen-year span. This anomaly could be attributed to the lack of continuity during this period: Richard Nixon resigned early, Gerald Ford served only two years, Jimmy Carter held a single term, while Ronald Reagan and George H. W. Bush dominated the presidency under one party for 12 years. I hypothesize that Wikipedia articles may not provide sufficient distinction between single-term presidents or those with shorter terms and two-term presidents. This could be due to various factors, such as the cultural significance of American presidents, the tendency for election years to emphasize specific topical issues, or even the sheer duration of a president's time in office. Another fascinating observation is that the algorithm clustered the 2016 and 2024 elections within the same partition, despite being separated chronologically by the 2020 election. Both elections marked Donald Trump's non-consecutive terms as president. A possible explanation for this grouping is that Joe Biden's single-term presidency in 2020 was distinct enough to form its own cluster, potentially due to unique challenges faced during his term, such as the COVID-19 pandemic, the Russo-Ukrainian War, the 2023 Israel–Hamas conflict, large legislative initiatives, and inflation.

The valuations and rankings of the centralities (closeness, betweenness, and eigenvector) revealed intriguing insights about the graph's structure and organization. For *closeness centrality*, which measures how close or accessible a node is to all others, it was surprising to find that none of the presidential elements ranked within the top 10. Instead, the top 10 nodes included: *United States, United States House of Representatives, President of the United States, Vice President of the United States, Unpledged elector, Tipping-point state, Super Tuesday, Main Page, Conservatism in the United States, and Red states and blue states.* These results are among the most predictable in the context of American elections, as all these terms are either fundamental to the electoral process or closely associated with presidential elements. For *betweenness centrality*, which measures how often a node acts as a bridge along the shortest paths between other nodes, and *eigenvector centrality*, which evaluates a node's influence based on the importance of its neighbors, the rankings were similar. In both cases, all presidential elections ranked highly, dominating the top of the lists. Excluding the presidential elements, the top 10 nodes were identical to those in the closeness centrality ranking: *United States, United States House of Representatives, President of the United States, Vice President of the United States, Unpledged elector, Tipping-point state, Super Tuesday, Main Page, Conservatism in the United States, and Red states and blue states.* However, as we move further down the rankings, the results begin to diverge. Differences in the rankings of nodes based on betweenness and eigenvector centrality highlight the distinct ways each metric captures the network's structural and organizational nuances.

Conclusion

The analysis of the Wikipedia pages for the past 14 presidential elections revealed intriguing insights into the structure of their links and the interconnected nature of Wikipedia articles. Through graphical algorithmic methods, such as node clustering, *PageRank*, and centrality measurements, the study uncovered patterns and highlighted key relationships that shape the organization and content of Wikipedia. Additionally, the development of the 3D Wikipedia graph visualizer introduced a dynamic and interactive platform for users to explore and interact with the article graph. By enabling free exploration of the graph, the visualizer offers an engaging and creative way for users to satisfy their curiosity and uncover valuable information about presidential elections as represented in Wikipedia.

Citations

Graph clustering algorithms: Usage and comparison. Memgraph. (n.d.). https://memgraph.com/blog/graph-clustering-algorithms-usage-comparison

Vasturiano. (n.d.). *Vasturiano/react-force-graph: REACT component for 2D, 3D, VR and AR Force directed graphs*. Vasturiano GitHub. https://github.com/vasturiano/react-force-graph

Wikimedia Foundation. (2024, November 13). *About*. Wikipedia. https://en.wikipedia.org/wiki/Wikipedia:About