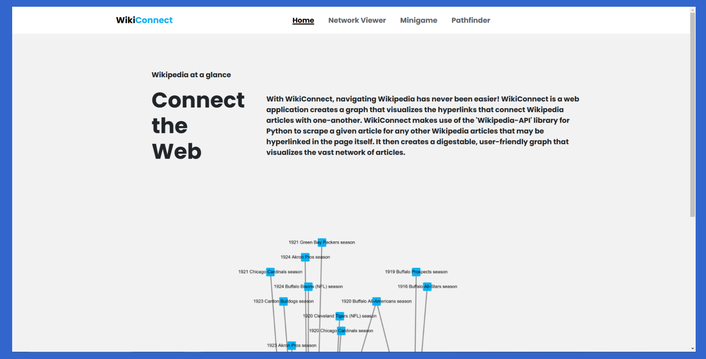
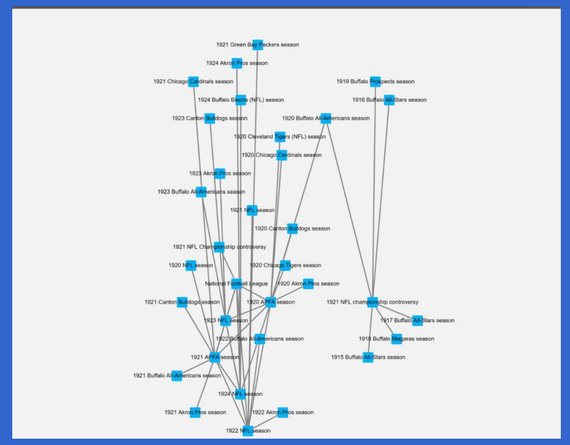
Wikipedia Network Visualizer and Analysis

Tremayne Booker, Jared Vitug
Department of Computer Science and Engineering. Project Adviser : Frank Witmer
4/28/2023

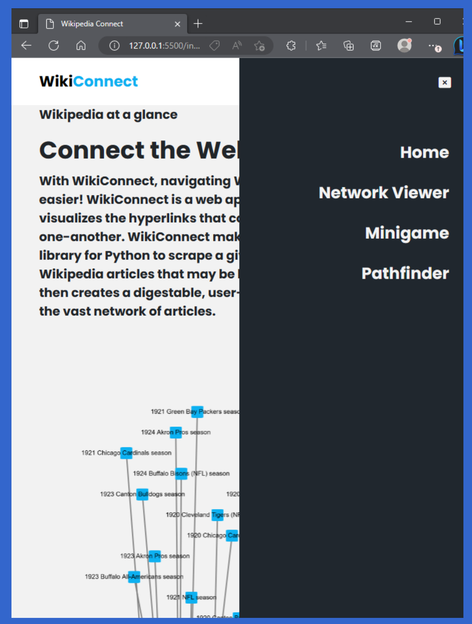
Wikipedia is one of the worlds largest publicly available network structures. With articles as nodes, and the links between them as edges, we allow people to visualize and interact with this network through our website : WikiConnect.



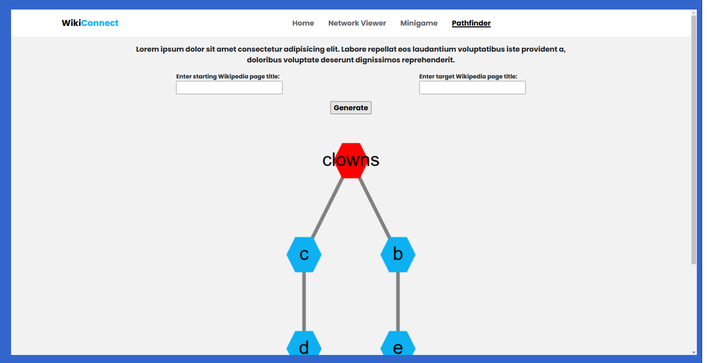
Homepage: This is where users are directed when first visiting the site. Gives a brief description of the website and an example graph.



An example network generated starting from the "National Football League". The first connections are 1920 APFA season, 1921 APFA season, 1922 APFA season, etc. before sprawling outwards



Mobile Version : A view of the mobile version of the website. All of the features work well within mobile and desktop.



Pathfinder: A feature to be implemented. Users can enter in a source article and a target article and the website displays the shortest path between the two.

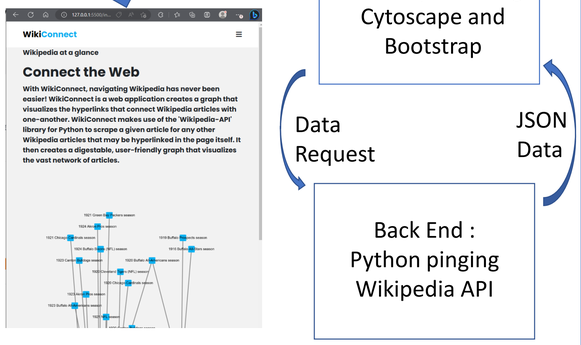
Architecture

The website has two main compononents : the front end and back end which interact with each other by passing JSON files via flask.

The front is coded mostly in Javascript. Stylization was done with "bootstrap" and "fontawesome". Although most of the heavy-lifting was through the plugin "Cytoscape". Cytoscape is a plugin that allows for a multitude of options to represent data as networks. Our graphs use the "Cola" layout. Cytoscape uses force-simulations to animate the graphs as they are generated or moved, with the cola layout giving control over these simulations. The graphs also require two other plugins, "Popper" and "Tippy" to create info boxes when hovering over nodes.

The back end is mostly in python. It handles requests for data. Whenever the front end needs to create a graph, it asks the back end what data is required to create it. It does so by collecting data from Wikipedia's API.

These are connected by "Flask" : a python package to integrate front and back ends. Flask allows javascript code to directly use python functions, as long as they return values acceptable by Javascript. For this, the back end send JSON files.



Data Collection

To perform analysis of the Wikipedia network, the entire network needed to be downloaded. Doing so was more difficult than originally planned as Wikipedia has a limiting throttle for API accesses. Data was collected over 12 days across multiple computers. These were stored in dataframes which were later combined to complete the network.

UAA College of Engineering
University of Alaska Anchorage

Network Analysis

Unfortunately, the entire network could not be analyzed due to system restraints. There are approximately 14 million pages on Wikipedia, each containing dozens of links. When constructed into a network there are dozens of billions of edges between these pages, and each edge requires at least a few bytes of data and a few hundred to store nodes. Therefore, to fit the network in memory, it requires upwards of a TB of RAM. However, analysis was performed on two subsections of data. These subsections were 1,600,000 and 1,000,000 articles large, and analysis was performed using a python package named "NetworkX". The aspects analyzed were :

Flow hierarchy; the fraction of edges that do not form cycles
Reciprocity; likelihood two nodes link to themselves
Centrality; the fraction of edges that connect to it
Assortativity; the likelihood any two connecting nodes connect to other, same nodes.

1,600,000 articles between "Densil Theobald"-"Georg Dedichen"
Flow Hierarchy : .833
Reciprocity : .0834
Most Central Node : ISBN (Identifier) .0212
Assortativity : -.107

1,000,000 articles between "!"-"ADS_13017
Flow Hierarchy : .605
Reciprocity : .340
Most Central Node : United States .0122
Assortativity : -.062

Further Work

The biggest improvement to be done is implement a larger than memory model to analyze the entire network rather than subsections. This will let us implement a "Shortest Path" finder, the framework of which can be seen in the "Pathfinder" image. Packages like "Dask" allow easy larger than memory management and will hopefully help wrangle the data to be more manageable.

Acknowledgements

We would like to thank our capstone professor and project adviser Frank Witmer for his help with our project. Also, ANSEP for providing a place to study and work. And of course, to Wikipedia for being an amazing educational tool with free information!