CSC111 Project Proposal: Comparing and Mapping Wikipedia's Articles

Gabe Guralnick, Matthew Toohey, Nathan Hansen, Azka Azmi

Tuesday, March 16, 2021

Problem Description and Research Question

Wikipedia is an online encyclopedia used by millions of people to learn just about anything. From quick look-ups to in-depth research, Wikipedia is an excellent tool for finding information.

Wikipedia sorts its articles into various topics and categories. These categories range from sweepingly broad to very specific. Take, for example, the page about William Clark (who you may know from the Lewis and Clark expedition). His article falls under categories *United States Army Officers*, a very large category, as well as *People from Caroline County, Virginia*, which is significantly smaller. Wikipedia articles also have many subcategories, creating a hierarchical structure. For example, the *People from Caroline County, Virginia* category is a subcategory of the *People by county in the United States* category.

But in an age of seemingly limitless information easily accessible over the Internet, it becomes difficult for casual users to determine what information is "worth" learning: which topics are actually influential and important, and which ones are more in the background?

Wikipedia itself does not differentiate between the "importance" of each of its articles. This approach is helpful in preventing any sort of discrimination on their part, but it prevents users from getting an idea of what people, places, or events are the most influential in human history. Because of this, for our project we hope to create a system for determining what topics are the most important.

This is a difficult question to answer, however; you could say that "United States" would be a very important article due to the country's large-scale involvement in global politics, but from an astronomer's perspective it's not very relevant. For this reason, our goal is for our solution to be user-driven: any user of our program will be able to input the category that they're interested in, and then our program will determine the most important topics within that category.

One potential criteria for the influence of a topic is how frequently its article is referenced by others on Wikipedia. Intuitively, if an article is linked to often by other articles, it is probably more influential to other topics, of higher quality, and more useful. As mentioned before, since Wikipedia contains a massive amount of different articles and topics, we'll focus on a given (user-selected) category. For our project, our group will investigate this research question: **What are the most frequently referenced articles across Wikipedia pages of a given category?**

Computational Plan

For our project, we plan on using the Wikipedia API to access article information, via the `wikipeda-api` Python library. This library will allow us to create `WikipediaPage` objects based on article titles. This class has various properties, including `links`, `back-links`, and `categories`, which will all be vital in allowing us to model the relationships between articles and categories. These properties are exactly what their names suggest: the `links` property provides a dictionary of titles to the related `WikipediaPage` objects that linked to *by this article*, `back-links` is a similar dictionary except that it includes articles that link *to this page*, and `categories` is another dictionary containing the categories the article belongs to.

Using the three properties outlined above, we will construct graphs of related articles (within a given category), which will be stored as `networkx.Graph` objects (or potentially as `networkx.DiGraph` objects). Each vertex will represent a Wikipedia article, and an edge between vertices signifies that the two pages link to each other. Then, we can perform different computations on these graphs, such as: finding the article in the category with the most back-links overall, finding the article that links to articles from the most other categories, searching for the article with the most back-links from within the selected category, or determining which other category has the most articles in common with the selected category, to give a few examples.

Given enough time, we may also try to model the hierarchical structure of Wikipedia's categories as a tree. This modeling (and possible visualization) will supplement the information given by our graph visualization to give us a better idea of the relationships between articles and categories.

Due to the nature of the API and the scope of the data, there will likely be some limitations on what categories we can feasibly operate on, and the types of computations we can do, as excessively large categories like "Human activities" may prove too large to traverse effectively. There is a good chance we will need to make use of `sys.setrecursionlimit` to increase Python's maximum permitted recursion limit, though we will also have to pay attention to memory usage when doing this.

Once we have modeled our data, we will use `plotly` to visualize the resulting graphs. Plotly includes various options for modifying the appearance of these visualizations, including node sizes, colours, edge widths, etc., so we can use these options to visualize the results of different questions we might want to ask, such as the article with the most back-links, by setting the node size to the number of back-links to the article.

References

Majlis, M. (n.d.). Wikipedia-API: Python Wrapper for Wikipedia. PyPI. https://pypi.org/project/Wikipedia-API/

Network Graphs. (n.d.). Plotly.com. https://plotly.com/python/network-graphs/

NetworkX — NetworkX documentation. (n.d.). Networkx.org. https://networkx.org/

Wikipedia contributors. (2021, March 13). William Clark. In Wikipedia, The Free Encyclopedia. Retrieved 15:43, March 15, 2021, from https://en.wikipedia.org/w/index.php?title=William_Clark&oldid=1011838536