Report (Work in progress.)

PREFACE

I have chosen to work on Project 2: The PubMed literature database.

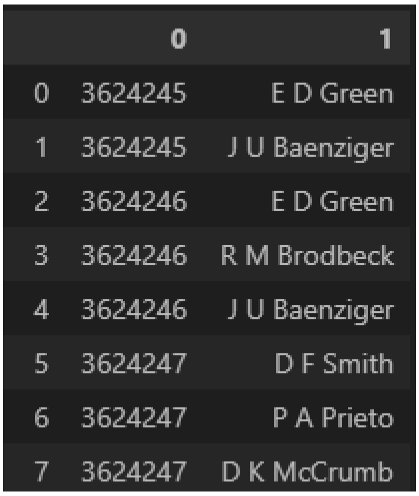
I started this assignment quite late and it has proven a lot more challenging and time consuming than I initially expected. Maybe I am doing this completely wrong but it will a 100% take me a lot more than just two days of coding and I sincerely doubt it can be done within this timeframe unless someone already has a lot of prior knowledge of graph theory. And even then, after looking online people have written entire papers on even a single one of these questions while applying it on a smaller scale.

However, I did answer ~~some~~ one of the questions and already created some of the data structures required to answer the other questions. Which I hope will be enough for me to pass the overall programming course.

I do enjoy this assignment so I will definitely work on it during the summer holiday, and then hand it in again later with hopefully a few more questions answered. For now I have answered the questions that I could manage to answer and for the others described how I plan on answering them.

DATA STRUCTURES

To answer the question I created different kinds of data structures. I did this by creating functions that loop over the xml files and then save the data that I want to pickle files. I applied these functions using multiprocessing.



	0	1
0	3624245	E D Green
1	3624245	J U Baenziger
2	3624246	E D Green
3	3624246	R M Brodbeck
4	3624246	J U Baenziger
5	3624247	D F Smith
6	3624247	P A Prieto
7	3624247	D K McCrumb

Figure 1: Part of author-paper dataframe

Author – Paper dataframe

The first data structure I created was a author paper dataframe. Where one column contains all the authors and co-authors and the other column contains the respective PubMedID that they have written. (see figure 1)

I created this dataframe by combining all the pickles created from parsing the xml files into one Dask dataframe.

The required XML parser is named *article_author* and can be found in *pickle_maker.py.* The function that creates the Dask dataframe can be found in *Assingment6.ipynb*

Author – Paper graph

Using the same pickle files I also created a bipartite graph which connects authors to papers they wrote and vice versa. Within the graph there are 72759 individual subgraphs. With the largest subgraph containing 95% of the full network nodes.

Most graph theory functions only work on connected graphs and I decided that looping over 72759 individual graphs just to obtain 5% more data was not worth it. So I only used the largest subgraph and did not look at the others.

This function and additional information can be found in *Assignment6.ipynb*

Citation graph (NOT Finnished)

In addition to a author – paper graph I also created a Citation graph. Where each paper is connected to a paper they reference.

Author – co-author graph. (Not started)

Creating a graph of all authors connecting two if they have worked on the same paper.

QUESTIONS

In this chapter I will answer all the given questions and explain how I answered them. Or explain how I plan on answering them. One problem I ran into while trying out things was that I so far I have only used the NetworkX library for graph theory and it happens to be extremely slow. And answering some of these questions requires a lot of computational power/calculations, which will require me to use another faster library if I want to finish running it within a year.

So far I have only answered question 1.

How large a group of co-authors does the average publication have?

I used the created dataframe with the IDs and the authors. And I divided the total authors with the amount of unique PubMedIDS to give me an answer of **4.2 average authors per paper.**

Do authors mostly publish using always the same group of authors?

To answer this question I wanted to use a modularity function. As this is the most common method used for finding communities (Newman, 2006). If there are few communities or very large communities it would mean that authors do not always use the group. If there are a lot of small communities it would mean that authors often use the same group of authors.

The reason why so far I have not been able to answer this question is because finding modularity is difficult, especially for large scale graphs. I did find an article: *Community Detection in Large-Scale Bipartite Biological Networks* (Calderer & Kuijjer, 2021). Which lists multiple methods including non modularity methods, but I will have to do a bit more research to see if I can use one of them.

Do authors mainly reference papers with other authors with whom they've co-authored papers (including themselves)?

Have not yet found a concrete method to solve this.

I did find an article: *A small world of citations? The influence of collaboration networks on citation practices* (Wallace, Larivière, & Gingras, 2006) This article examines the proximity of authors of those they cite using degrees of separation.

What is the distribution in time for citations of papers in general, and for papers with the highest number of citations? Do they differ?

The XML file does not give me information regarding how often and on what time a specific paper has ben cited as far as I could tell. I could look at the time distribution of the used references but I'm not sure how interesting this is.

Is there a correlation between citations and the number of keywords that papers share? I.e. papers which share the same subject cite each other more often.

Make a citation graph and add the keywords as attributes and see if the connections/edges have more overlapping keywords than average/try some sort of correlation test.

For the most-cited papers (define your own cutoff), is the correlation in shared keywords between them and the papers that cite them different from

This question needs the same method as 5, except I need to find a cutoff for most cited papers which can be easily found by looking at the node degree.

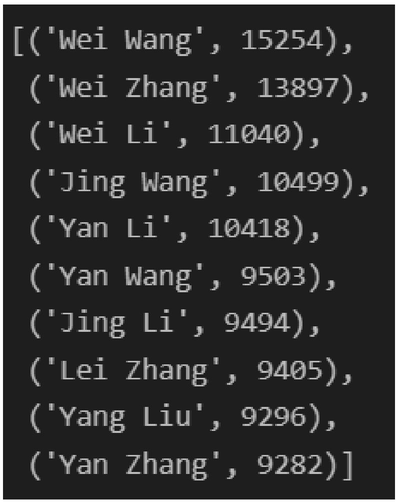
Own Question: What % of papers does not have a reference list

A Lot of papers seem to be lacking a reference list so I wonder what % this is. I can answer it by dividing the amount of papers with a reference list by the total amount papers * 100.

OBSERVATIONS

Chapter with interesting/useful information that I found.

After created the author – paper graph I checked for the nodes that have the most connections/degrees. This will either show the author with the most papers or the paper with the most authors. What I found was the results shown in figure 2. It shows Authors with more than 10 000 papers. This makes it seem like they are the busiest writers in the world. Which Is most likely not true as they are also all very common Chinese names, meaning that there are probably a lot of different authors publishing under that name. However the PubMed xml files contain no unique author id as far as I could tell so there is no way to remove/split duplicate names from the dataset.



```
[('Wei Wang', 15254),
 ('Wei Zhang', 13897),
 ('Wei Li', 11040),
 ('Jing Wang', 10499),
 ('Yan Li', 10418),
 ('Yan Wang', 9503),
 ('Jing Li', 9494),
 ('Lei Zhang', 9405),
 ('Yang Liu', 9296),
 ('Yan Zhang', 9282)]
```

Figure 2: Nodes with the most degrees.

REFERENCES

Calderer, G., & Kuijjer, M. L. (2021). Community Detection in Large-Scale Bipartite Biological Networks. *Applications and Methods in Genomic Networks* .

Newman, M. E. (2006). Modularity and community structure in networks. *PNAS*.

Wallace, M. L., Larivière, V., & Gingras, Y. (2006). A small world of citations? The influence of collaboration networks on citation practices. *Science and Engineering Indicators 2006*.

To add:

https://www.researchgate.net/publication/5915523_Module_identification_in_bipartite_and_directed_networks https://www.frontiersin.org/articles/10.3389/fgene.2021.649440/full

https://www.sciencedirect.com/science/article/pii/S0378437119322642?casa_token=uKtpa6Qm-sIAAAAA:_8ySg7xyYQ-5sNj3CrzNqT4btTKndE0Cr0td31jOu9KHHna4HrG7alp9OHW69hqkkeGLEph3pfU

https://www.researchgate.net/publication/235883078_Network_Effects_on_Scientific_Collaborations