Coronavirus AI Final Report (Group 3)

Introduction

Coronavirus has forced the world into paralysis. The majority of central Europe has gone into lockdown to prevent any further spread of the virus[[1]](#footnote-1). The virus is the focal point of everyone in the world today, thanks to the spotlight it receives from world media. This ever-growing problem encouraged our group to try and help in any way we could to try and find some form of answers to assist in the prevention of the spread of the virus. As we discussed in our project proposal, we acquired two datasets from the internet. We have used these datasets along with tools inside SNAP to create a visual representation of countries that would be highly likely to become infected with the virus.

Our Approach

We had a clear idea of what we were going to do from the start. We knew who was doing what and how we were going to do it. Even though we did miss out some aspects of our original plan, we still managed to gain the analysis and answers we thought we would receive from our work.

Datasets:

For our Flights dataset, we combined two sets of data, routes.dat and countries.dat from the OpenFlights site[[2]](#footnote-2) to create a parsed collection of data that only had the necessary fields. We combined them into one file having just the flight source and flight destination. To do this, we used C# due to the simplicity of its file handling and data structures. We had one issue with this dataset, this being that it stopped receiving updates in 2014, but we agreed to use it in the interest of time. After creating this dataset, we decided that for the problem we were aiming to sort, we didn’t need to account for domestic flights so we created a small method in C# that would remove the domestic flights and export it as a CSV to make it easy to read with C++.

Our second dataset was to do with the current COVID-19 cases. We created a Python program to get data from a website[[3]](#footnote-3) which gave Snap something to use. After we realised, we wanted more since it was day dependent, we moved on.

The third dataset was hosted on GitHub[[4]](#footnote-4) and contained the number of confirmed cases, recovered cases and deaths due to coronavirus. in parts of the world. Various publications cited the source for use in calculating the number of confirmed cases in the United States[[5]](#footnote-5).

SNAP:

Out of all the snap options that we had, we decided to use CoDa. CoDa was not only an option of necessity but also a choice of interest. The necessity part was since when we looked at the folders for all the integrated snap programs, a multitude of them wouldn’t handle a TNEANet network, and there were no conversions to a graph. Despite this, we looked online for resources that could help, but unfortunately, most of these were in SNAP for Python and were challenging to transpose into SNAP for C++. An idea we had for solving the problems surrounding networks in SNAP was to overload the SNAP functions manually. We tested this idea with the GraphViz function allowing us to edit the DOT file more, such as adding more parameters to Nodes such as labels, colours and more. Overloading functions was quickly resolved but making the data structure stable took more time. When it came to interest, we chose coda because our overall project showed the connections between nodes that had Covid-19 and the nodes that didn't. Coda gave us a different analysis of the data, instead of showing us the relationships between all the nodes coda gave us the option of seeing the overlap of communities and where the connections are which reinforced our predictions.

JavaScript:

In this project, we used JavaScript with NodeJS to do a variety of tasks. The next job was the process of the primary COVID-19 dataset. A file called server.js which handled these server-side operations. This downloads individual CSV files which contain the data for each day, respectively. This data was split into individual regions such as provinces of China and counties of the United States, meaning these fields were combined to paint a national picture.

We also perform calculations and other data analytics, such as: calculating the daily percentage increase in confirmed cases, the number of active instances and estimating how many cases are undiagnosed. We calculated estimated examples by assuming a case fatality rate of just under 1%; this was based on the scale seen in China outside of the severely affected Hubei province. So, if a country had a death rate of 10%, we assumed there were ten times as many cases as having been confirmed.

This data could then be exported to files to be used by the Snap part of the project and served to the website via the Express API. The site is used to display information by country, continent and across the world in a timeline manner. We utilise the data that we have parsed from the dataset by creating a slider function on the website’s homepage.[[6]](#footnote-6) We can then visualise the increase in confirmed cases, deaths and number of recovered cases.

Our Results:

The results we received were very predictably since they all shared a consistent pattern. Most resulting countries were connected to an infected country, i.e. a country that has confirmed cases of COVID-19, through a flight route. This is what made the results predictable. It was predictable since each country had some form of connection to a country that is infected, so it was highly likely that the connected countries would have some kind of infected traffic going back and forth in the foreseeable future. This effect was particularly evident in nations that had flights arriving from Italy.

Improvements to our approach

Datasets:

We could have spent more time looking for/creating an up-to-date flights dataset, but with the time we had, we wanted just to have a working set of data. We also would have taken into account domestic flights as they are a big part of spreading the virus, but our goal was to track and predict the spread between countries so were deemed necessary.

SNAP:

As said previously, we had many problems with SNAP when it came to the Network type we wanted to use. We would have converted more of the SNAP programs to work with networks, but with the limited manpower, we would have found it challenging to do as not only did you have to overload all the functions, you had to fix all the data structures as well. Also, we would have made it more like a choice based program to give more flexibility and more accessible to view all actions possible.

JavaScript:

Overall, we are delighted with the website we created. However, there are improvements we would have wished to make. For starters the dataset despite being corroborated by trustworthy sources, it is full of inconsistencies. Many of these are down to reporting from the individual countries themselves and conflicting information from different data sources such as the World Health Organisation (WHO) and individual countries’ health ministries. For example, this can lead to the number of confirmed deaths due to COVID-19 in Australia declining.

If we had time, we would have tried to clean up the dataset ourselves or used versions of the same dataset available online that have been cleaned up already. However, we weren’t confident in switching to them due to recent changes to the CSV format that broke our code and probably broke theirs as well. Perhaps the ultimate solution would be to create our dataset based on the WHO’s PDF situation reports or some other reliable collaborator.

We also wished that we could have found a solution to integrate SNAP data into our website, including displaying the routes used on the map and displaying risk factors for each country for international transmission. Presenting this data together would have made for a more concise and user-friendly solution.

Overall:

We feel we should have worked more within Snap to process data and that we should have had more people on that job since clearly, they had many struggles with Snap. These struggles are due to a lack of documentation and the incompleteness of Snap for C++: Networks had little support with any programs besides the basics, meaning processing a network required making your own programs and functions.

Conclusion

To summarize, by using tools within SNAP in concurrence with our datasets, we have managed to create a network that we could use to predict the next outcome of the most significant ongoing crisis in the world today. Our main goal was to provide some insight into this crisis to help countries that may not have any confirmed cases of COVID-19. Looking back at the project now, we believe that we have achieved that goal and in doing so, opened much insight into the spread of COVID-19.

Word Count – 1518 words

1. <https://www.bbc.co.uk/news/world-europe-52025553> [↑](#footnote-ref-1)
2. <https://openflights.org/data.html> [↑](#footnote-ref-2)
3. <https://www.worldometers.info/coronavirus/> [↑](#footnote-ref-3)
4. <https://github.com/CSSEGISandData/COVID-19> [↑](#footnote-ref-4)
5. "Coronavirus map of the US: latest cases state by state | World ...." 29 Mar. 2020, <https://www.theguardian.com/world/ng-interactive/2020/mar/29/coronavirus-map-of-the-us-latest-cases-state-by-state>. Accessed 29 Mar. 2020. [↑](#footnote-ref-5)
6. <https://rsewell.tech> [↑](#footnote-ref-6)