**Getting started with network analysis**

**Introductory Gephi course for Humanities staff**

1. **Introduction**

Welcome to this workshop! In this hands-on part of the session, you’ll be creating your own retweet network visualization in Gephi. We’ll be working with a large Twitter dataset containing 74.519 tweets posted between 9-10 February 2022. The tweets are related to the Canadian truckers protests (or Freedom Convoy), when large amounts of truckers occupied Canadian cities and highways to protest against vaccination mandates and other COVID measures.

Our dataset represents a slice of the debate that took place on Twitter right in the middle of these demonstrations. We’ll be aiming to find out which communities were involved in the online discussion, which accounts were key players and how the different actors were positioned in relation to each other.

1. **Getting to know the data**

Let’s start by getting to know the dataset.

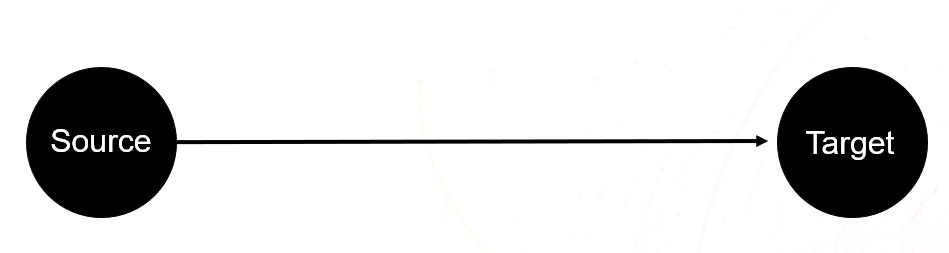
Open the file data\_truckersconvoy.csv in a spreadsheet application of your choice (Excel, Numbers, Google Spreadsheets, etc.) and take a look around. What information can be found in this dataset? What types of insights do you think you can gain from doing a network analysis on this data?

1. **Nodes and edges tables**

Every network in Gephi consists of two data tables: a nodes table and an edges table.

The nodes table contains all information about our nodes.The nodes table always contains an Id column, which is used to identify the node. In this case, nodes represent Twitter users, and the Id column therefore shows their Twitter handle. Nodes tables can be enriched with extra information, too. We’ll be going into that later.

The edges table describes the connections between the nodes. Every edges tables contains a Source and a Target column. In a directed network, the Source is the node from which the connection originates, while the Target is the ‘receiver’ of the connection.



Open the nodes and edges tables in a spreadsheet application and compare the two. What do the connections in the edges table represent?

1. **Importing the datasets**

Time to make a network! Open Gephi and choose to make a New Project. We now need to import our edges and nodes tables. Go to File 🡪 Import Spreadsheet and select edges.csv.

Gephi automatically recognizes that this is an edges tables. Choose Next and wait for Gephi to show the import report.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Here we can decide a few key elements of our network, such as the type of network (directed, indirected or mixed), how parallel edges should be managed, and whether nodes should be able to connect to themselves (self-loops).

Our edges represent retweets. Think about what this means for these parameters and set them accordingly. Not sure? Feel free to ask your instructor to help.

Next, let’s do the same for the nodes table. Import the spreadsheet. If all went well, you shouldn’t need to change any of the parameters. Choose to Append to existing workspace, so that the nodes and edges table are both imported in the same workspace.

**Valuable tips and tricks (courtesy of Marjolein Krijgsman)**

* + - Gephi has no ‘undo’ or ‘redo’ button. CTRL + Z does not work!
    - Gephi does not list and save your actions and settings. **Thus, take a notepad and write down the settings you use,** so you will be able to remember what you did if you open your project later on. The Utrecht Data School in collaboration with Digital Humanities Lab has developed a plugin to automatically list all your setting. You can find and download the Field Notes plugin here: <https://github.com/UUDigitalHumanitieslab/gephi-plugins/tree/fieldnotes>
    - **Save your project often (use the Save As option)!** Also save intermediate results, as there is no undo button. If you make big changes to your visualization that do not turn out well, you will still have your old save.
    - When you run an algorithm on a huge dataset, don’t be afraid let it run for a couple of hours or even overnight.
    - In Gephi you can use various algorithms. ForceAtlas2 is the most used algorithm. If you use Gephi for academic research, you will need to understand how the algorithm works on at least a basic level. To do this, you can read up on the algorithms in the papers written by the developers.
    - For others to understand a Gephi visualization, you will need to provide additional contextual information.
    - Hovering over features in Gephi will result in a small yellow box that explains shortly what the features can do.
    - Don’t be afraid to try out different options and see how your visualizations change accordingly. To master Gephi you will just need to try out different things and will most likely encounter many frustrating problems.

1. **The shape of a network to come**

Most likely, this is will be staring at your from the screen right now:



No worries! This is the shape of an untouched network. Contained within this Rothko-like monstrosity are all your nodes and edges. In the upper-right corner, in the Context window, you can check exactly how many of them there are.

Let’s start with a basic spatialization of the network. We’ll be using the force-directed lay-out algorithm ForceAtlas2 for this. In essence, ForceAtlas2 treats nodes as if they were connected to each other by rubber bands. Stronger connections means that the nodes are attracted to one another, while loose (or weak) connections result in the nodes pushing each other away. This way tightly connected groups of nodes will cluster together, resulting in a network that can be read quite intuitively.

Go to the Lay-out window and choose ForceAtlas2 from the list of options.

Afbeelding met tafel

Automatisch gegenereerde beschrijving

We want to start by roughly spatializing our network, and running the more precise algorithms later on. Tick the Approximate Repulsion and Dissuade Hubs boxes and click Run. It will look like the Big Bang will take place on your screen. Let this run for a little while, until the network barely moves anymore, then Stop the algorithm.

If you’re feeling experimental, try ticking other options and see how the network changes.

Afbeelding met dag

Automatisch gegenereerde beschrijving

Now that’s more like it! Let’s take a look around your network. You can zoom in and out by scrolling. Holding the right mouse button allows you to drag around the view.

1. **Running statistics**

Now that our network looks like an actual network, we need to run some statistics to prepare for further analysis. Find the Statistics tab on the right-hand side and run the Average Degree and Average Weighted Degree calculations. Gephi will now calculate these metrics for each of the nodes.

Go to the Data Laboratory tab on the top and navigate to your nodes table. Look at that: the results from our calculations have been added to the dataset!

Go back to the Overview. Let’s use our calculations to transform the look of our network. We want to make nodes of users with many retweets bigger, and colorize the nodes based on how many times they have been retweeted. Go the Appearance window on the left. Here you can transform the visual appearance of the network based on your data values.

Let’s start with the size of the nodes. Choose Nodes and the Size option (three circles). Go to Ranking, which gradually increases the size of a node based on a higher or lower ranking of a value. In this case, choose Weighted In-Degree (or: how many retweets the node has received). The Min and Max size are up to you. For the size of this network, 10 and 100 will be appropriate. Now Apply this transformation.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Next, do the same for the color of the nodes by choosing the painter palette icon.

Because our nodes changed sizes, we need to give them some space. Go back to ForceAtlas2, turn off Approximate Repulsion (so that the algorithm will be more precise) and turn on Prevent Overlap. Run the algorithm for a while (this might take some time!). Now the nodes will no longer cover each other.

Our network is becoming more and more readable by the minute!

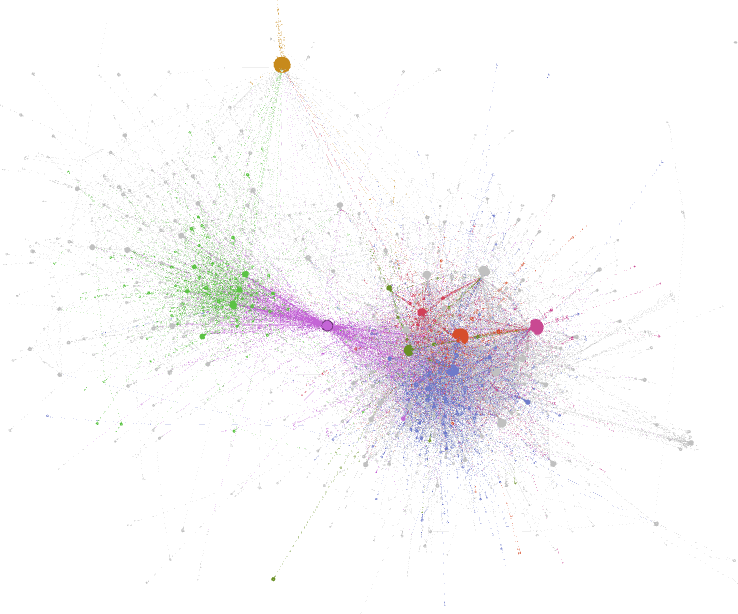
1. **Modularity classes**

One of the main reasons to do network analysis is to detect communities in a network of actors. Communities are groups of nodes that are intimately connected. It depends on the context what these groups represent. In our case, users that form a community very often retweet each other, which in most cases points to a high level of agreement. This can help us detect ideological or topical communities within a debate.

In Gephi, communities can be detected by the Modularity option under the Statistics tabs. Run Modularity. You will be asked to set a resolution. This will influence how big or small the resulting communities will be. The higher the number, the bigger the communities. Let’s start with the default resolution, 1. Run the algorithm.

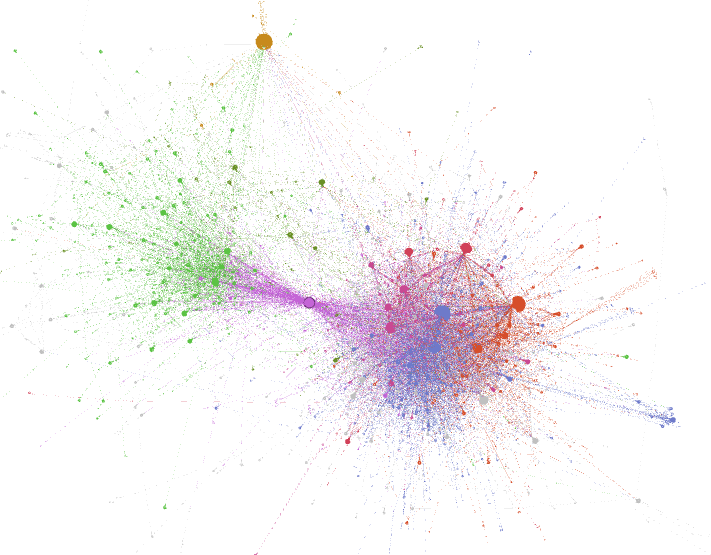
The results screen will show you the number of communities that Gephi detected. That’s quite a lot! Look at the network. Why do you think that is?

Colorizing the network based on the communities is one of the staples of network analysis. It helps us reveal the underlying structures of the network. Go to the Appearance tab 🡪 Nodes 🡪 Color 🡪 Partition and select Modularity Class. Partition (as opposed to Unique or Ranking) allows us to assign a unique color to each different value for a metric. Go to Palette… and choose a nice set of colors for your communities. Now apply the colors. The result will look something like this:



*The network after running the Modularity analysis with a resolution of 1.*

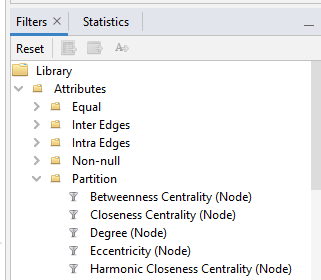
Sometimes, the number of communities makes it hard to parse the network. In that case, it might be useful to increase the resolution of the Modularity algorithm to find less, but bigger communities. Repeat the above steps, but with a larger resolution (for example, 1.5 or 1.8). See if the results improve. Please note that you need to regenerate the palette after each iteration of your Modularity calculation!



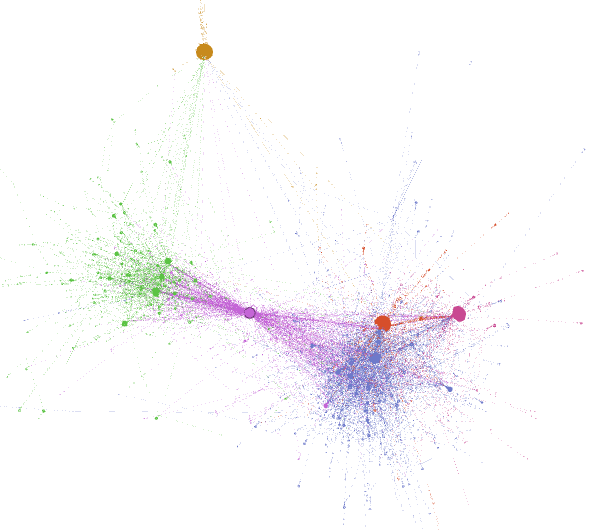
*The network after running the Modularity analysis with a resolution of 1.9.*

1. **Using filters**

Our network is still a bit messy. We can use filters to clean things up. First, let’s start by only keeping the main clusters, eliminating the ‘noise’ on the borders of the network. Go to the Filters tab 🡪 Attributes 🡪 Partition.

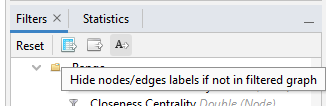


Find the Modularity Class (Node) filter and drag this to the Queries window underneath. By clicking the boxes next to the Modularity Class ID’s, you can select which communities you want to keep. Start the Filter and select the biggest communities, starting from the top. The Context window on the top right shows you the percentage of nodes and edges that you have filtered out. You network will now look a lot more clean.

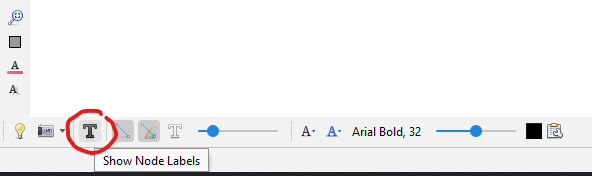
****

Always be aware that filtering is a very deliberate process. You inherently exclude data from your visualization once you start filtering. If you use filters in your analysis, be very clear in describing what you filtered and why.

Filters can be used for different means as well. For example, we can use a filter to view the names of the biggest nodes in the network. Go to Filters 🡪 Attributes 🡪 Range and drag the Weighted In-Degree filter to the Queries window. This filter allows us to filter based on a range of Weighted In-Degree values, meaning the amount of retweets an account received. Run the filter an start dragging the slider on the left. When you only see big accounts in your network, click the A with an arrow in the Filters window:



This option makes sure that only nodes that are now visible will show their labels (in this case, their Twitter handles). Turn on the labels by clicking the black T on the bottom of the overview. You can change the size and appearance of the labels with the options on the right. Try playing around with these until you’re happy with the result.



You can now turn on the Modularity Class filter again, which will return the network to the view that we made earlier.

1. **Exporting the visualization**

Now that we have a readable network, we can export our visualization as an image file so that we can share it with others. Go to the Preview tab at the top of the window. Here we can change how our network appears in the image. The structure of the network won’t change here, only how it shows up.

Play around with the options here and see what they do. Please note that you need to click Refresh for the changes to show up. Find a combination of settings that result in a network that you think is nice and clear.

Once you’re happy with how your visualization looks, we can export the image. Go to Export on the bottom left, and choose PNG. Click options and increase the resolution to be very high (for example, 16000 x 16000 pixels). Because networks can often contain tiny details and letters, you want your image file to be of high quality so that you can zoom in without loss of detail. Export the image.

1. **Exporting your data**

Finally, let’s get the results of our network analyses out of Gephi. Go to the Data Laboratory, select the nodes table and click Export Table. The data that was visible in your nodes table will now be exported as a CSV file, which can be used for further analysis in different environments (e.g. Excel, Tableau, Python, R).

1. **What’s next?**

This tutorial taught you the basics of Gephi by guiding you through the creation of a network visualization. You know how network data is structured, how to import that data, spatialize the network, run statistics, change the appearance of nodes based on the results of these metrics, use filters to clean up the network and export the result as an image. These steps are all you need to tackle all kinds of different networks in Gephi.

However, we have barely scraped the surface of what Gephi can do. There are many more different metrics, visualization options, and lay-out algorithms waiting for you. Gephi is very well documented, and hovering over options will give you a clear description of what each option does. It is therefore very easy to start experimenting with the possibilities of this program. Want to make nodes bigger based on their number of followers? Want to colorize the most connected nodes within the network? It’s all possible.

Try to come up with questions that you want to ask this dataset, and try to answer them using Gephi. Need input? Just ask Jeroen!