Scraping Web of Science

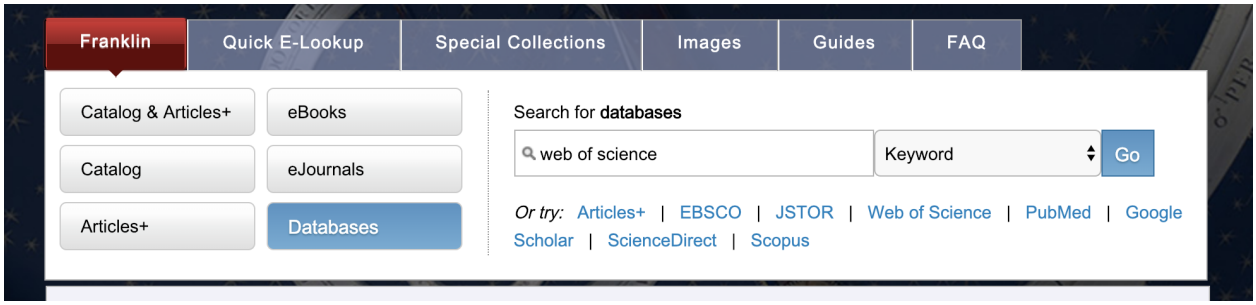
The aim of the protocol is to create a dataset with all the manuscripts published by faculty in your neuroscience department of interest. The dataset of manuscripts will have the name of the authors, the manuscript abstract, and keywords associated with the manuscript. We will then be able to apply natural language processing techniques to ask questions related to the diversity of the science being conducted by individuals and departments on these data.

Step 1: Manually create a list of faculty in the neuroscience department in an excel file (if you don't have excel on your computer, a google drive (https://www.google.com/drive/) spreadsheet will work. The first column should contain the faculty member's first name and the second column should contain the faculty member's surname.

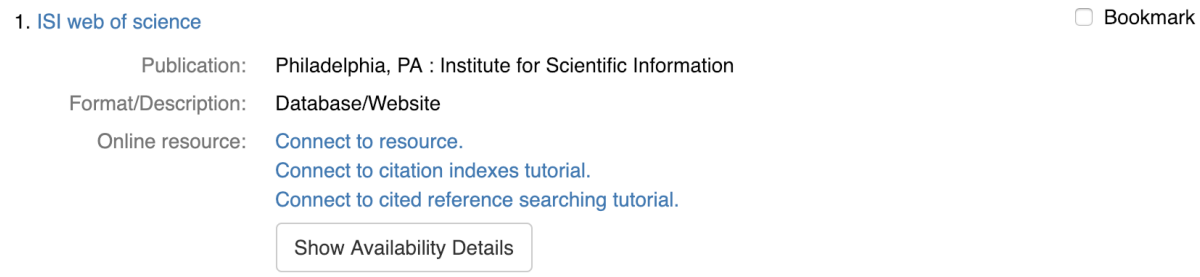
Step 2: Go to web of science and find the web of science institutional code for your neuroscience department.

Go to Penn Libraries: https://www.library.upenn.edu/

Search for web of science in "Databases":



Click on the "ISI web of science" result that comes up:



And then scroll to the bottom of the page that loads and click on "Connect to Resource":

Web link: The Rosengarten Family Fund Home Page

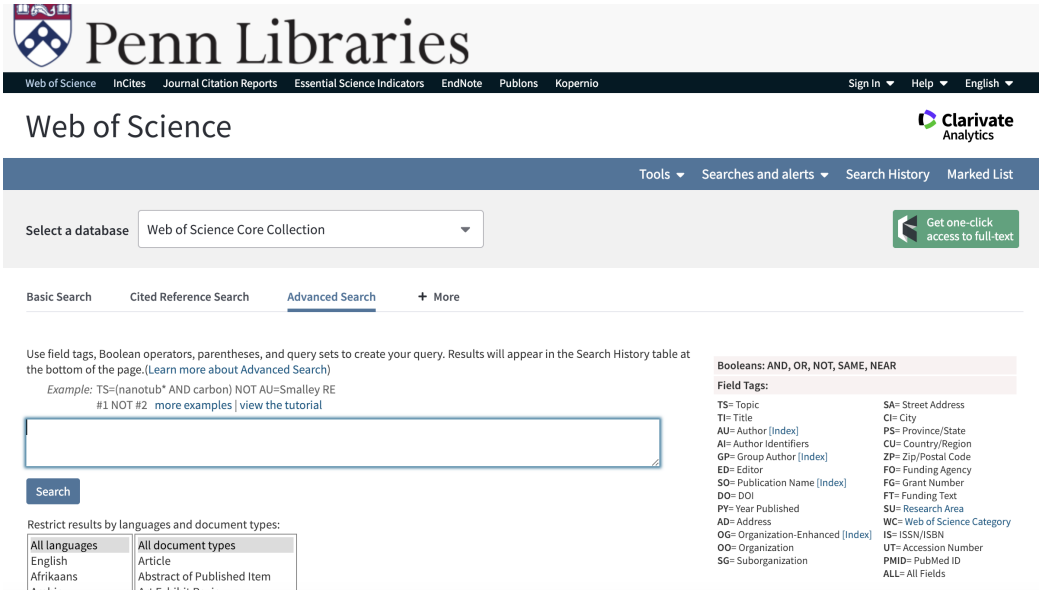
Access Restriction: Restricted for use by site license.

Online: Connect to resource.
http://hdl.library.upenn.edu/1017/6959

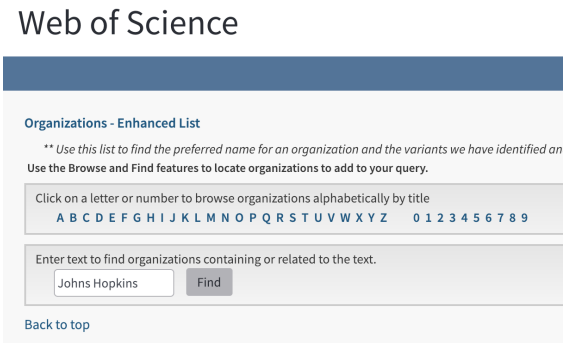
Connect to citation indexes tutorial.
http://hdl.library.upenn.edu/1017/6959/1

Connect to cited reference searching tutorial.
http://hdl.library.upenn.edu/1017/6959/2

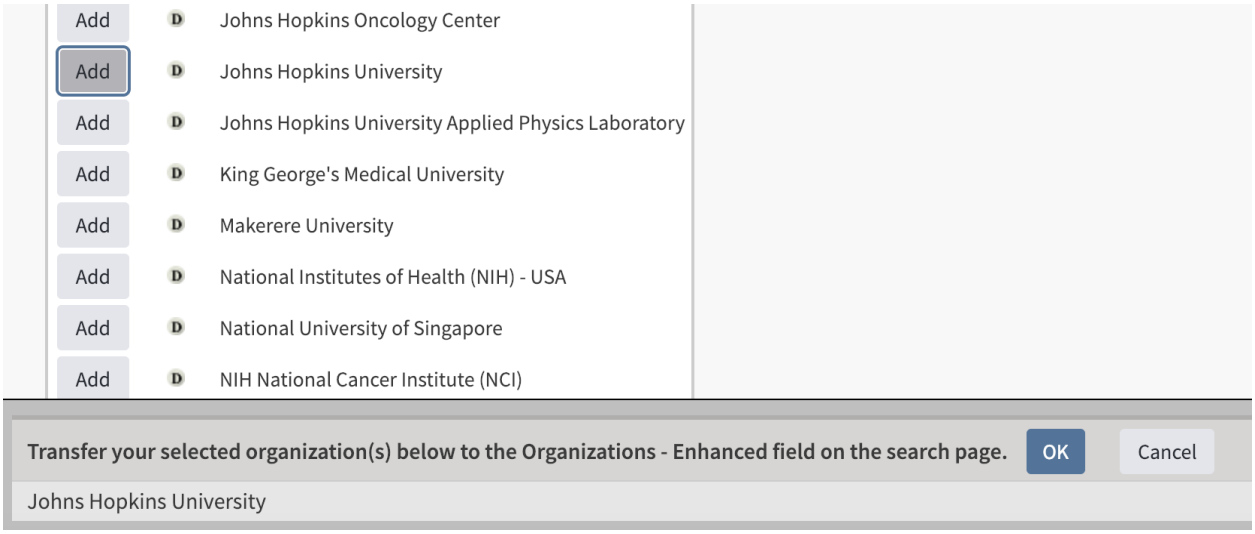
You should get to the Web of Science splashpage. Click on “Advanced Search” and your page will look like this:



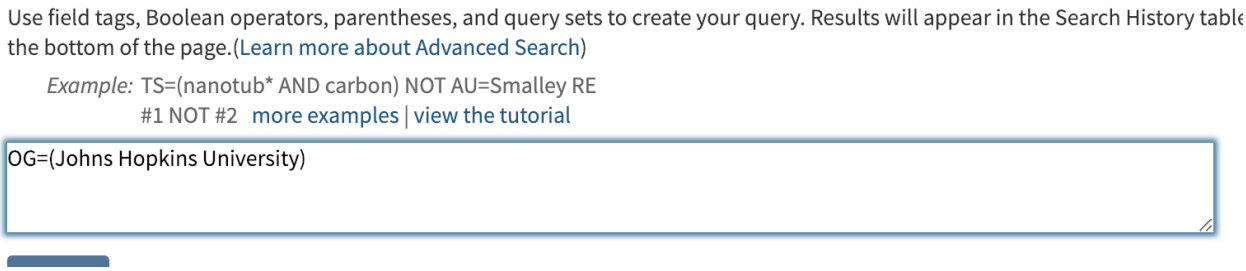
On the right-hand slide, in the “Field Tags” section, click on the [index] after **OG=**Organization-Enhanced. Search for your department in the search bar that appears (here I’m searching for Johns Hopkins):



Find your department's tag, click the "Add" button beside it and click "Ok" where it says "Transfer your selected organization(s) below to the Organizations - Enhanced field on the search page" at the bottom of the page.



This will bring you back to the advanced search and will give you the code you will need for your manuscript search. Here the code we need for Johns Hopkins is OG=(Johns Hopkins University):



Step 2: Use excel to create a block of text that you can then copy and paste later when searching to speed up the process.

Paste the code below into the first row of your completed excel sheet. It can be pasted into any column as long as it is not the first two that you have created using the faculty member names. This is creating a search string we will use later consisting of author names (AU=) and department of origin (OG=[change this based on Step 1]). The text in red will fill automatically with the faculty member's surname (=B1) and their first name (=A1).

AU=	=B1	=A1	AND	OG=(Johns Hopkins University)	OR

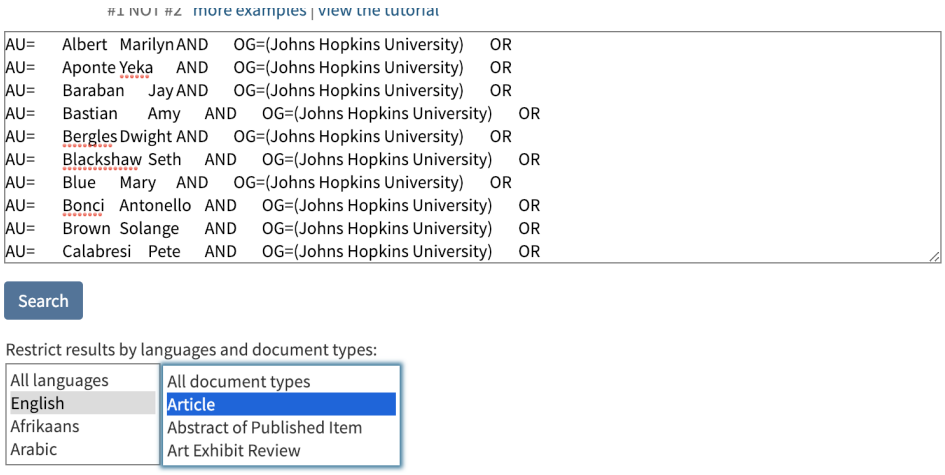
Highlight all 6 columns, grab the bottom-right corner of the rightmost cell with your mouse, and drag it down. It will autofill all the faculty members you entered as you drag it down:

	A	B	C	D	E	F	G	H	I
1	Marilyn	Albert		AU=	Albert	Marilyn	AND	OG=(Johns Hopkins University)	OR
2	Yeka	Aponte		AU=	Aponte	Yeka	AND	OG=(Johns Hopkins University)	OR
3	Jay	Baraban		AU=	Baraban	Jay	AND	OG=(Johns Hopkins University)	OR
4	Amy	Bastian		AU=	Bastian	Amy	AND	OG=(Johns Hopkins University)	OR
5	Dwight	Bergles		AU=	Bergles	Dwight	AND	OG=(Johns Hopkins University)	OR
6	Seth	Blackshaw		AU=	Blackshaw	Seth	AND	OG=(Johns Hopkins University)	OR
7	Mary	Blue							
8	Antonello	Bonci							
9	Solange	Brown							
10	Pete	Calabresi							
11	Peter	Campochiaro							
12	Michael	Caterina							
13	Pablo	Celnik							
14	Jeremiah	Cohen							
15	Carlo	Colantuoni							
16	Ed	Connor							
17	Susan	Courtney-Faruqee							
18	Kathleen	Cullen							
19	Ted	Dawson							
20	Valina	Dawson							
21	John	Desmond							

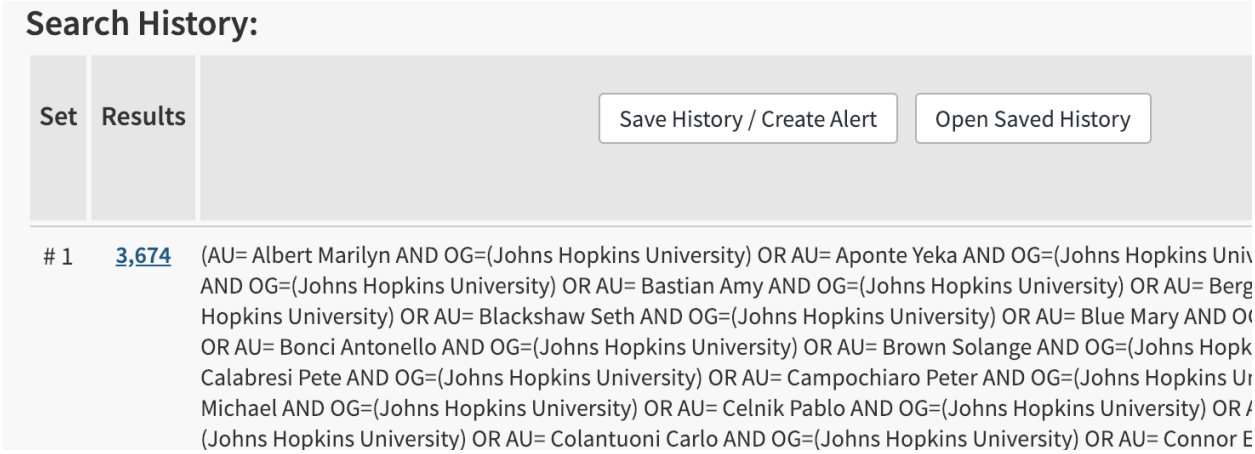
Remove the last cell when you get to the very bottom:

112	Mark	Wu		AU=	Wu	Mark	AND	OG=(Johns Hopkins University)	OR
113	King-Wai	Yau		AU=	Yau	King-Wai	AND	OG=(Johns Hopkins University)	OR
114	Eric	Young		AU=	Young	Eric	AND	OG=(Johns Hopkins University)	OR
115	Don	Zack		AU=	Zack	Don	AND	OG=(Johns Hopkins University)	OR
116	David	Zee		AU=	Zee	David	AND	OG=(Johns Hopkins University)	OR
117	Kechen	Zhang		AU=	Zhang	Kechen	AND	OG=(Johns Hopkins University)	OR
118	Feng-Quan	Zhou		AU=	Zhou	Feng-Quan	AND	OG=(Johns Hopkins University)	
119									

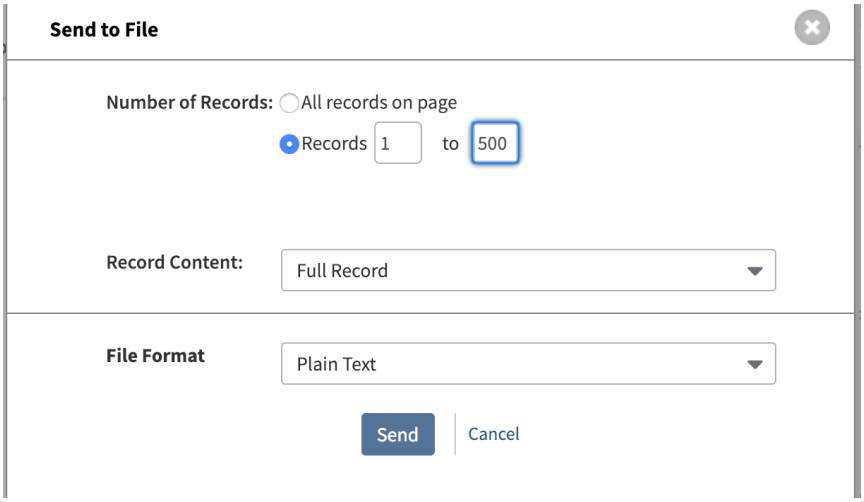
Step 3: Search web of science for all relevant articles.Select all the text in the 6 columns you created, copy it, and paste it into the advanced search in Web of Science, setting the restriction options to search for “English” and “Article” and press Search!



Step 4: Output the resulting data. Click on the search result number (here 3, 674). This will bring you to the articles:



Click on the dropdown option to “Save to Other File Formats”. An option box will show. Web of science restricts exporting to 500 articles at a time so you may have to repeat this process a few times by changing the numbers you place in “Records” - here we are getting articles 1 to 500. Make sure the “Record Content” and “File Format” settings match:



Click “Send” and the plain text document will download.

Building Semantic Networks

This section will guide you through the steps to make a network that can be visualized using Gephi and analyzed using MATLAB. If you haven't already, please download and install Gephi.

Constructing a network from scientific abstracts

1. **Download word embedding model from the link below (file size: 809 MB)**

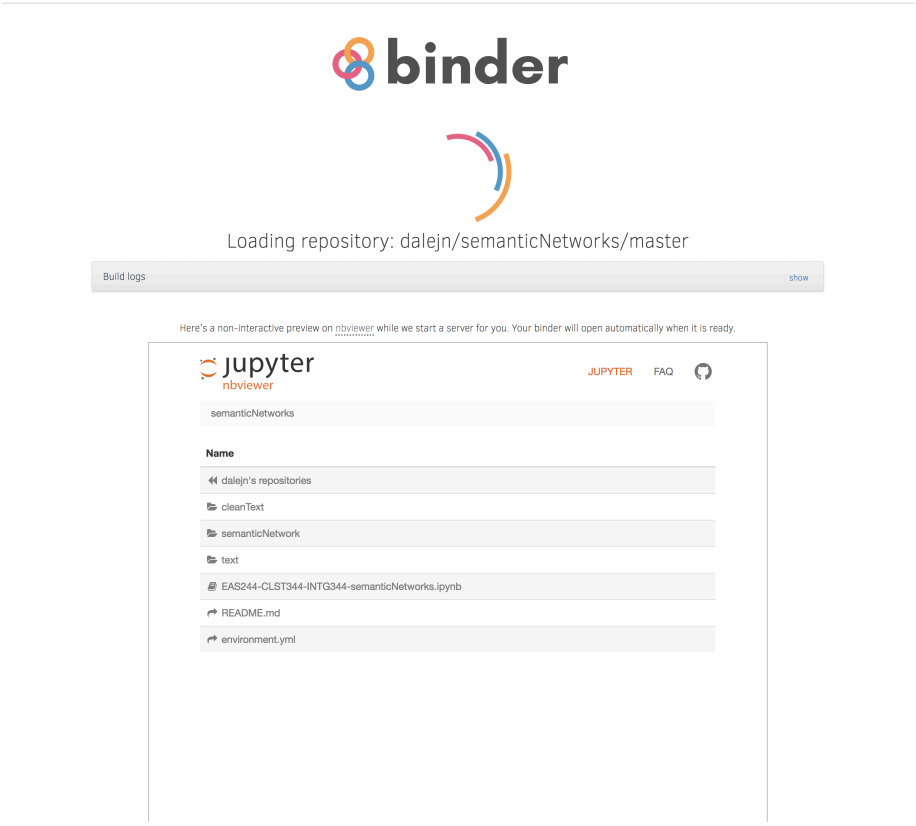
   https://www.dropbox.com/s/je4slqywoh8wnr1/enwiki_5_ner.txt?dl=1

   This is a neural network model of word vector embeddings with 296,630 word vectors representing words' semantic meaning trained on 2,252,637,050 words from Wikipedia. If interested in more details, see last section.

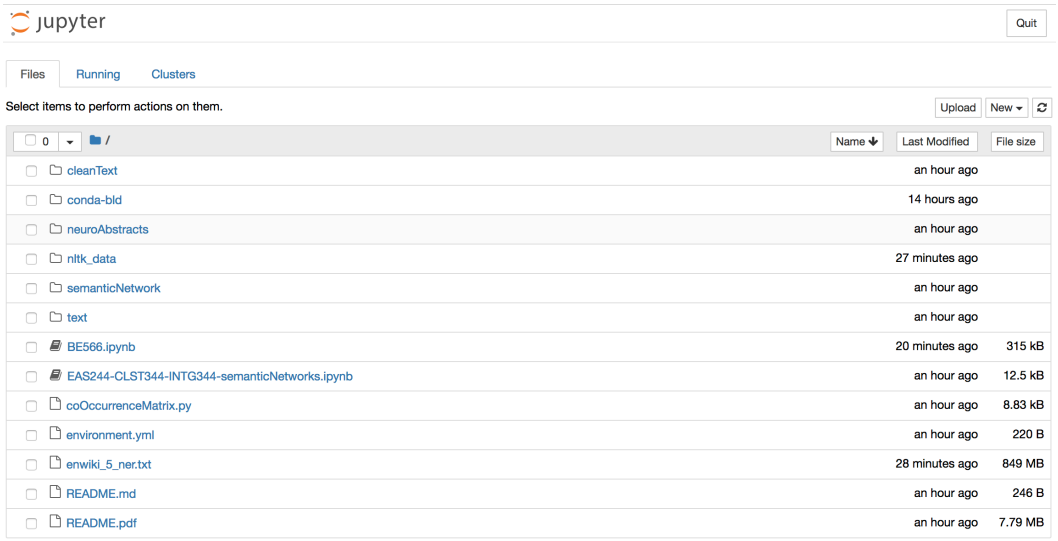
2. **Start programming environment**

   https://mybinder.org/v2/gh/dalejn/semanticNetworks/master

   A. You should see the loading page below. Depending on how many students are trying to access the server, it may take a few minutes or longer to start the programming environment. If it is still loading after 10 minutes, please refresh the page.

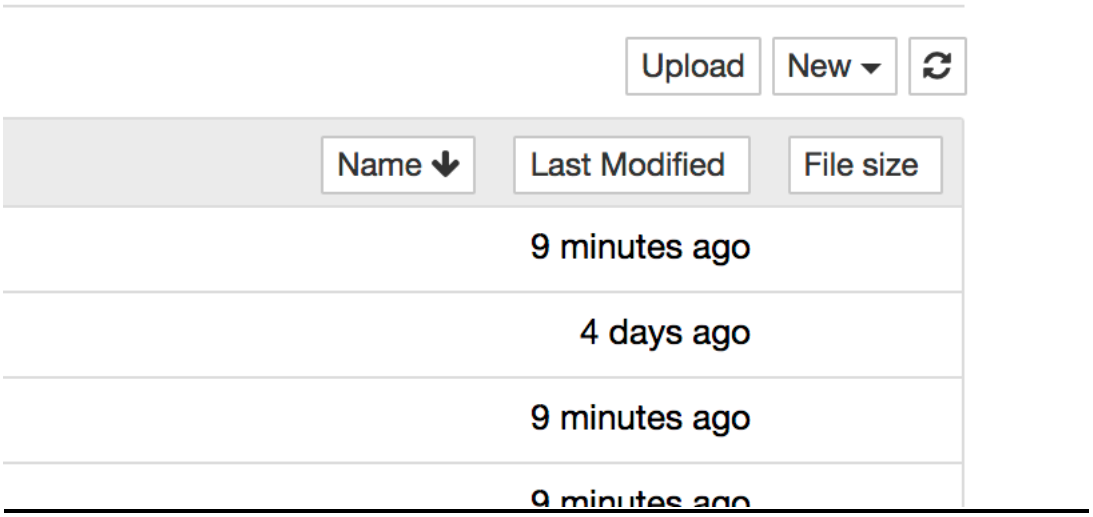


B. When the page is loaded, you will see a screen like the one below. Note, that the environment will automatically shut down after 10 minutes of inactivity (if you leave your window open, this will be counted as “activity”).

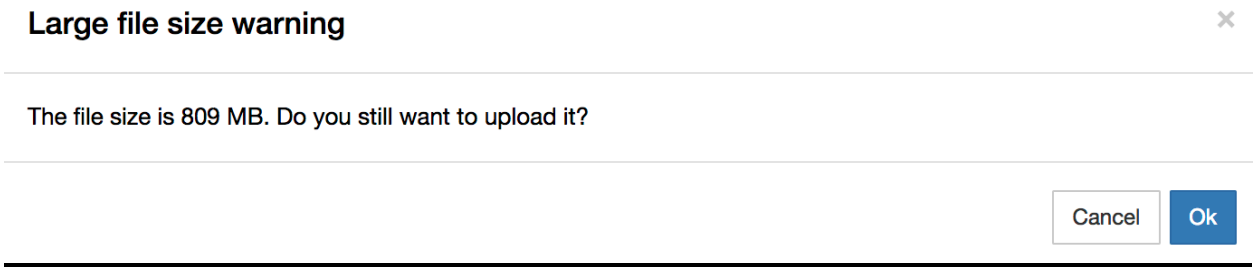


3. Upload the model downloaded from step 1 to the environment

A. Click on “Upload,” go to where you downloaded the file from Step 1 and select it.



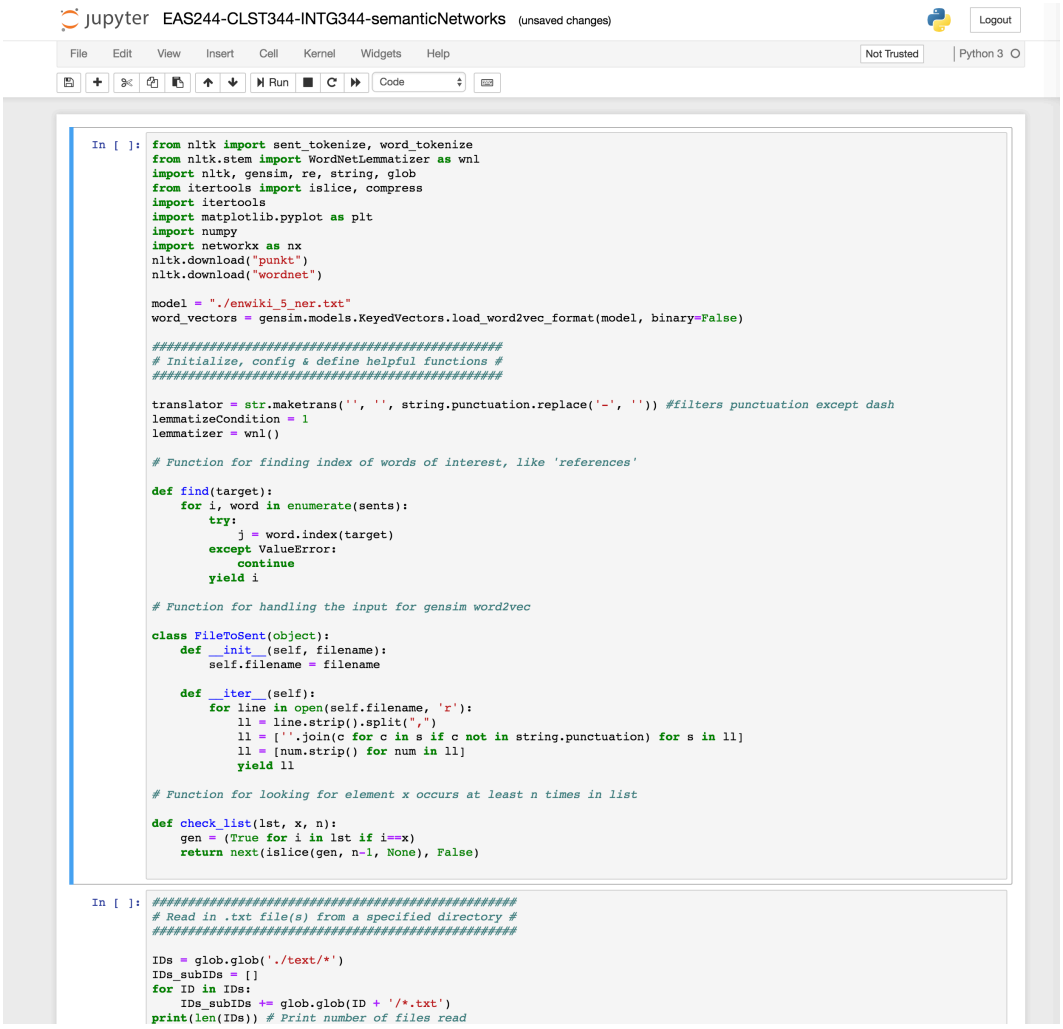
B. Press Ok after selecting the model to upload.



C. Click on the blue Upload button and you should see the percent progress of the upload. Depending on how many students are accessing this environment, this upload can take a few minutes or longer.

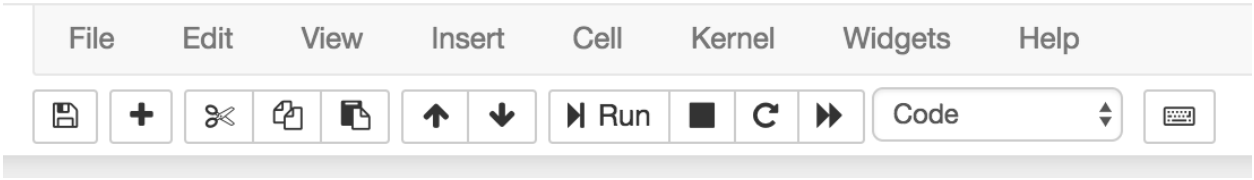
4. **<u>Once the upload has completed, click on the text link "BE566.ipynb" as seen below to open a programming notebook.</u>**

This will open a new tab/page where you should see the below:

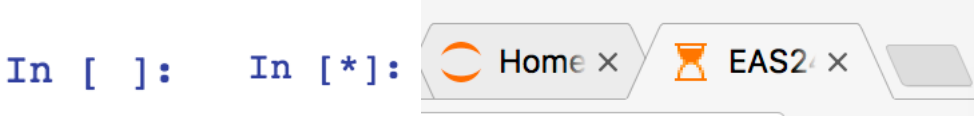


5. Run the code

A. 1st box: loads the relevant packages, modules, and functions. Click on the first box, which should then become highlighted along the borders. At the top of the screen in the toolbar, click the Run button.



You should see the left-most text change to the right. Your tab in the web browser will also show an hourglass. This means the code in this box is running. This can take a few minutes or more to finish.



B. 2nd box: reads in your text and cleans it. When the code is done running, the asterisk above will be replaced by a number and the hourglass will be replaced by a notebook. Now, click on the second box of code, which should become highlighted along the borders. Run this code by clicking the Run button. This will take a few seconds at most.

C. 3rd box: constructs the semantic network matrix from the shortest N cosine-distances (the strongest N associations). Select the third box. Depending on the number of nodes in the graph (i.e. number of unique words in the text), you may want to change the total number of edges (i.e. connections between the words) in the constructed semantic network. I've set this number to be 10 times the number of nodes, but this can be changed depending on how dense you want the resulting semantic network to be.

```
# The number of connections we want: either as a factor of the number of words or a set number
num_top_conns = len(my_words) * 19
```

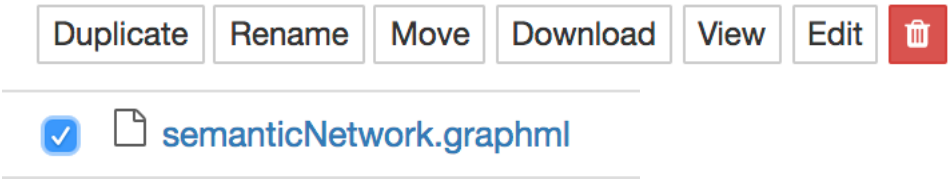
You will see a list of words outputted.

D. 4th box: constructs a co-occurrence network based on 5-gram sliding window (represents a count of words pairs that co-occur with each other in 5-word chunks across the entire text)

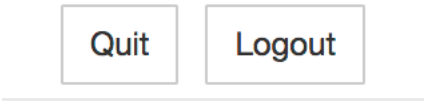
6. **Download the relevant outputted files**

A. You can close the notebook now. Return to the directory screen and click the "semanticNetwork" link.

B. Check a box (**one at a time**) to download **all** the files in this folder. You must select the files one at a time, or else the Download button doesn't appear.



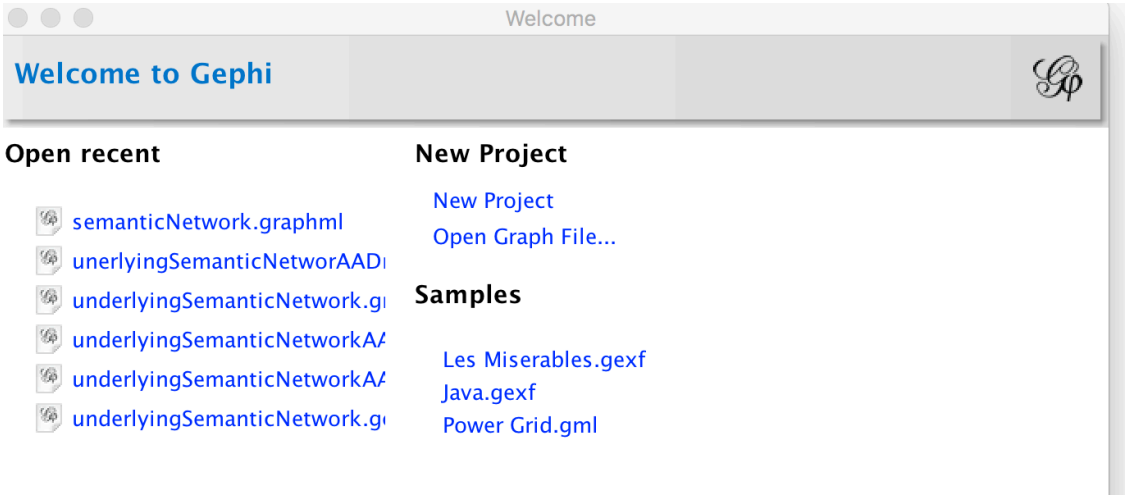
C. Once you've downloaded **all files in this folder**, press Quit at the top right.



7. **Visualize the network in Gephi**

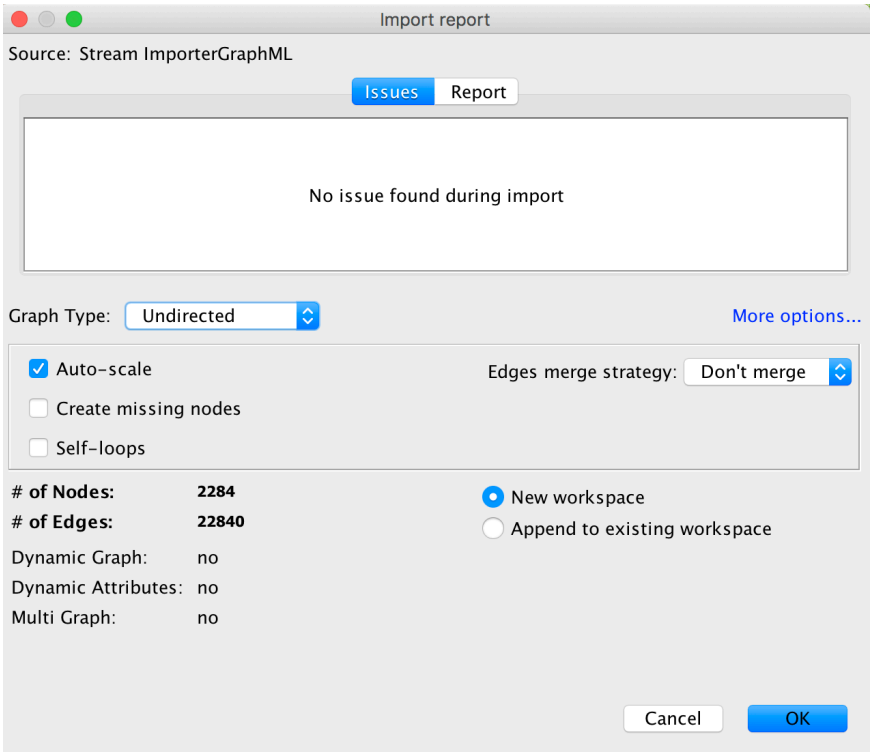
A. Open Gephi. If you haven't already downloaded and installed it, please do so here.

You will see a Welcome screen like below. Click on "Open Graph File…" If you don't see the welcome screen, go to File and Open on your computer's toolbar.

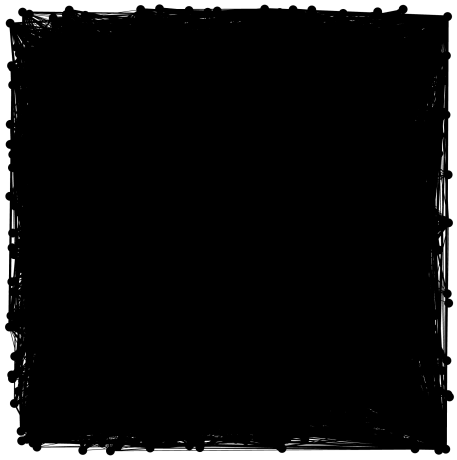


B. Navigate to where you downloaded semanticNetwork.graphml and open this file.

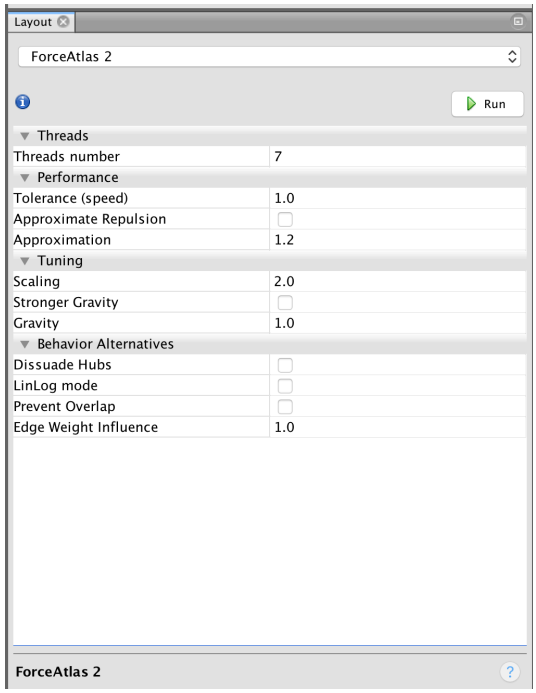
C. A new screen should pop up. Default settings should be fine but please check that they match the ones below and then press OK. Note: If you are loading coOccurrenceNetwork.gexf, there will be many "Issues" listed—you may ignore them and press OK.



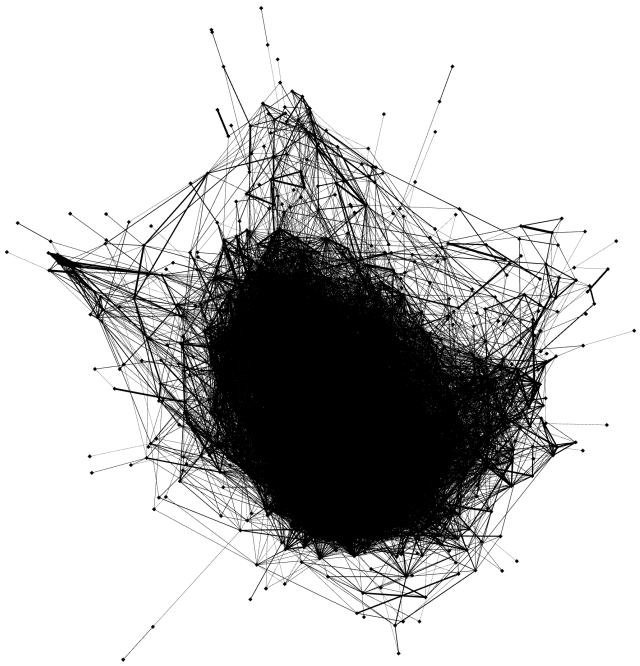
D. The network should be visible. If not visible, you may need to go to Gephi's menu at the top and select Window > Graph. You can scroll your cursor over nodes to see connections. We'll now clean this up to make it more interpretable.



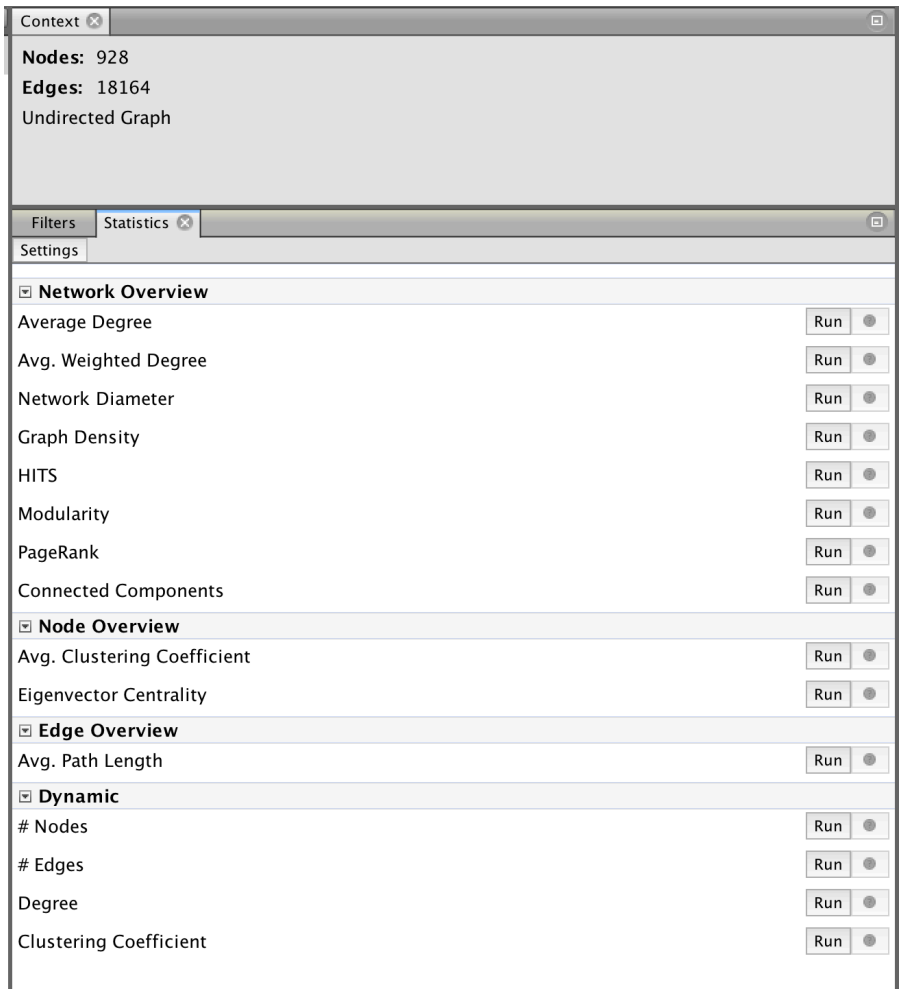
E. On the bottom left, click on "---Choose a layout" and select "ForceAtlas 2." Then, press the Run button, which will then be replaced by a Stop button.

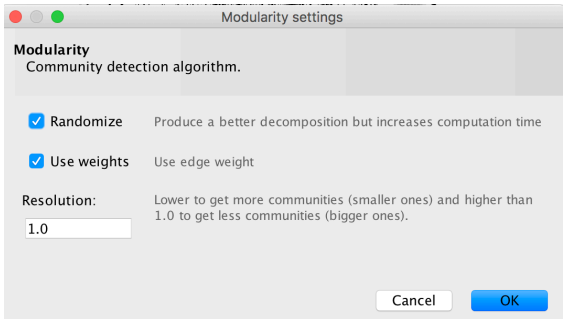


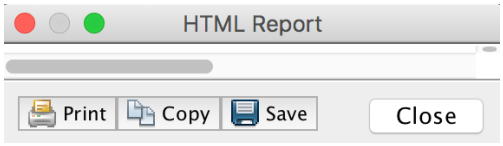
F. Use your mouse wheel to zoom in and out. Hold right click and drag to move the network around. When the network looks like it has stabilized (not moving much), press the Stop button.



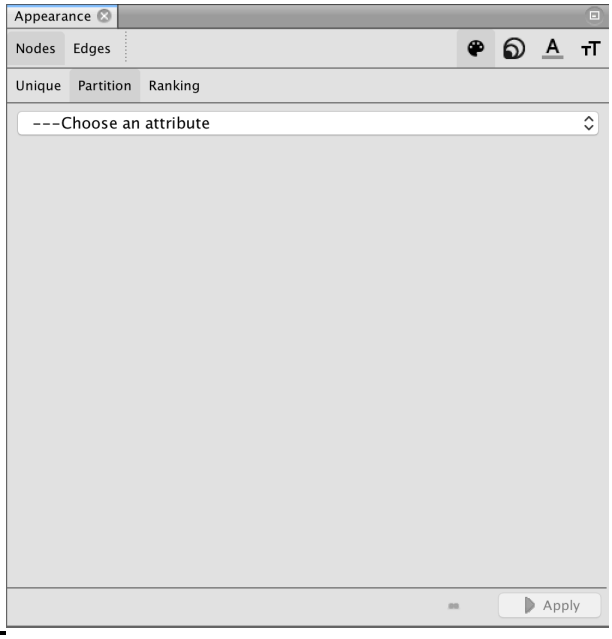
G. On the right-side toolboxes, click on the “Statistics” tab and then click “Run” for the Modularity measure. A box will pop up; click OK. Another pop-up labeled HTML Report will appear; click Close.



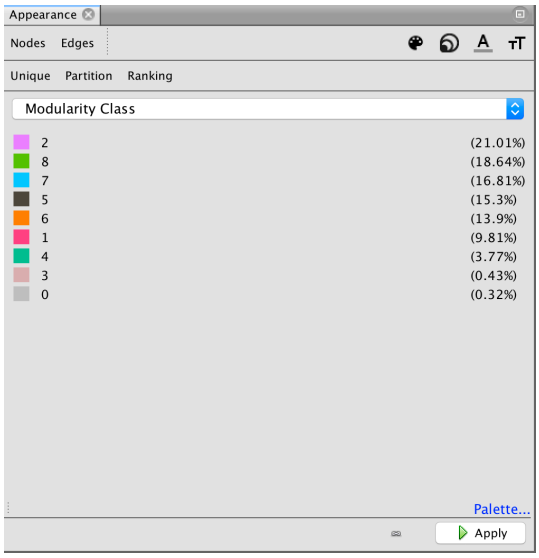




H. At the top-left toolbox labeled Appearance, click on the “Partition” tab.

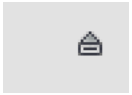


I. Click on "—Chose an attribute" and select "Modularity Class." Click "Apply."



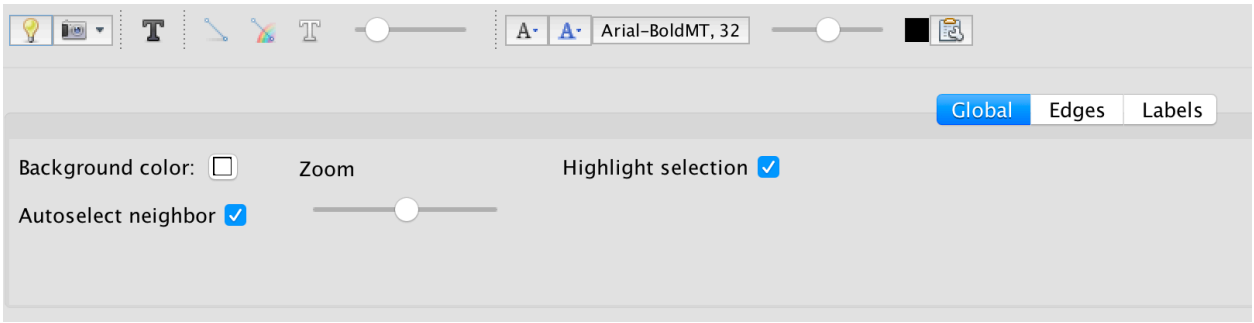
Your graph should now be colored by module.

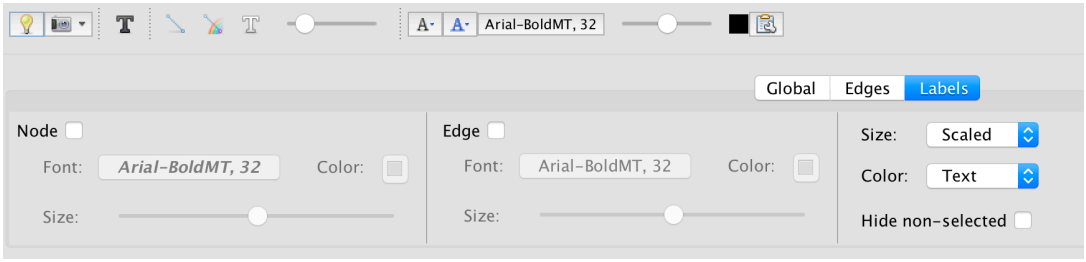
J. Label each node with the unique word it represents. At the bottom, there is a toolbar; click the small button all the way to the right.

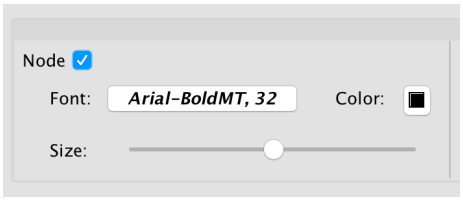


← Small button

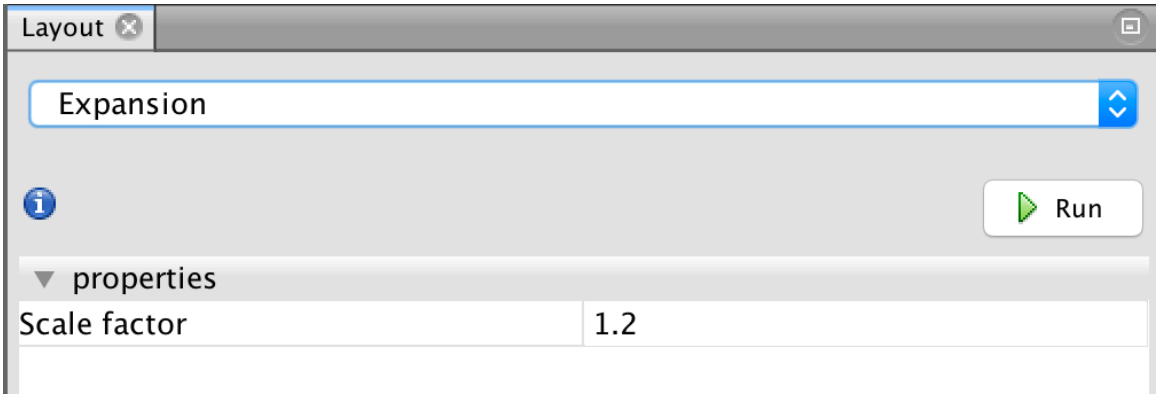
K. Click on the “Labels” tab and click the checkbox for Node.



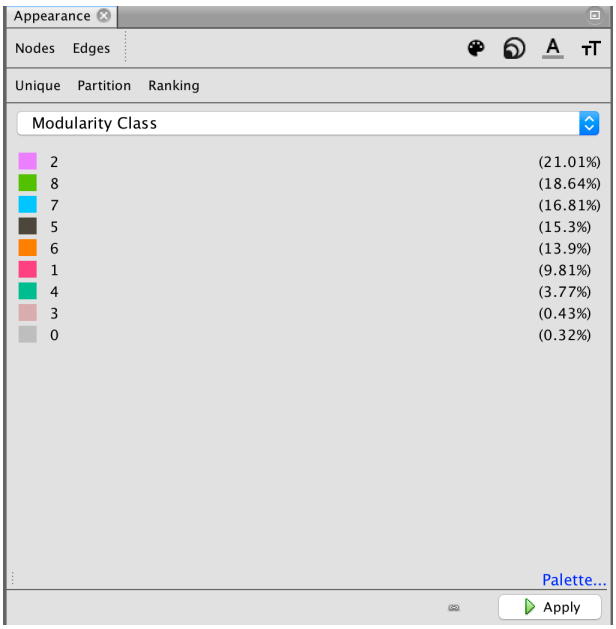


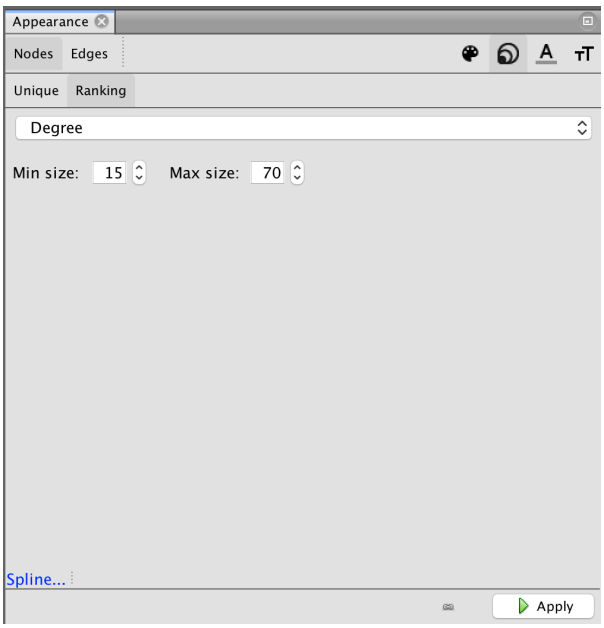


L. Spread out the graph to read the text more easily by going to the "Layout" pane again (the same place where you applied "ForceAtlas 2") and select "Expansion." Click on Run a few times until you're satisfied with the spread of the text labels. Then, zoom out on the graph using your mouse wheel.

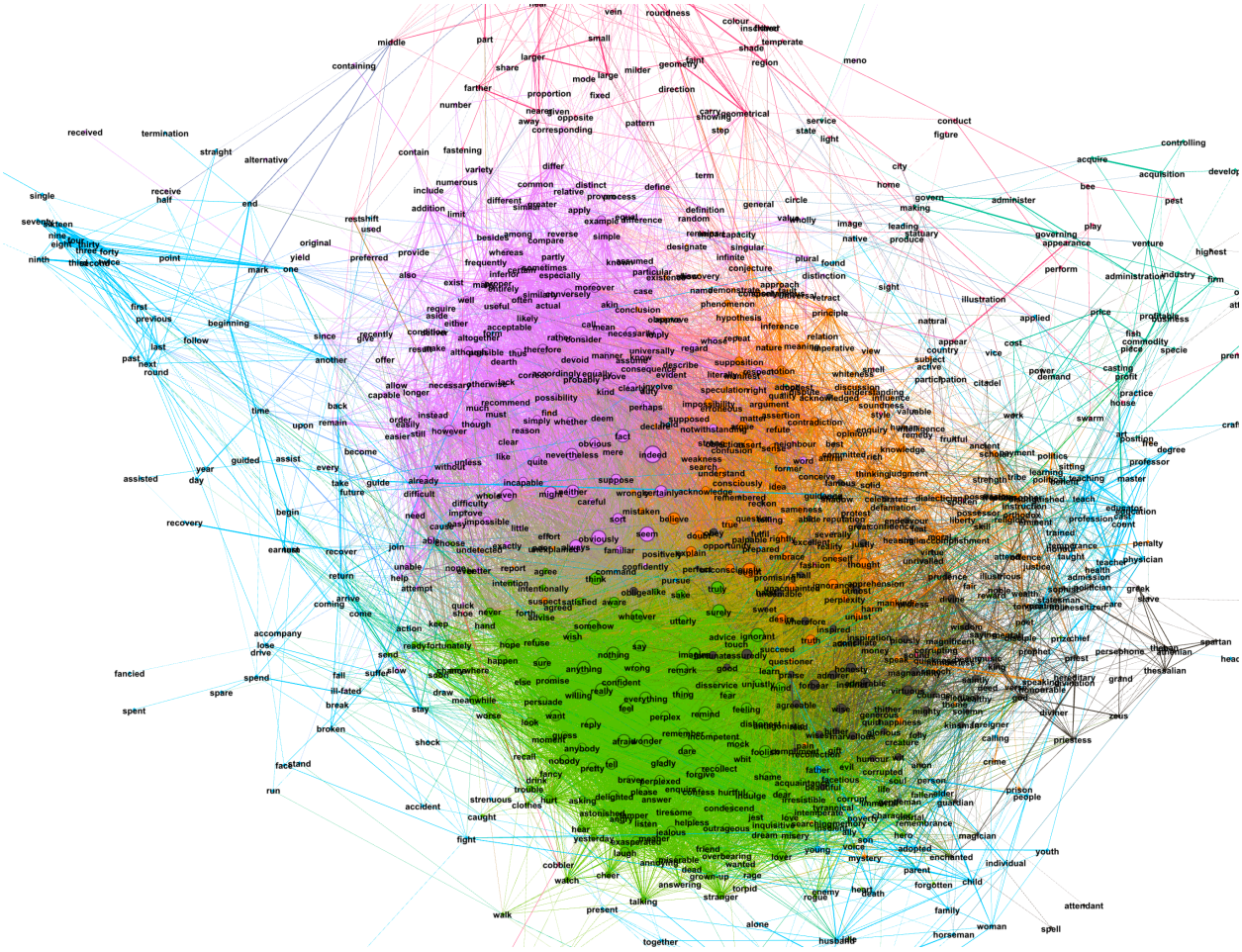


M. Finally, navigate back to the "Appearance" pane and click the  button. Click on the "Ranking" button. Then, select "Degree" from the dropdown menu and change the "Min size" and "Max size" to the ones below. Click "Apply".





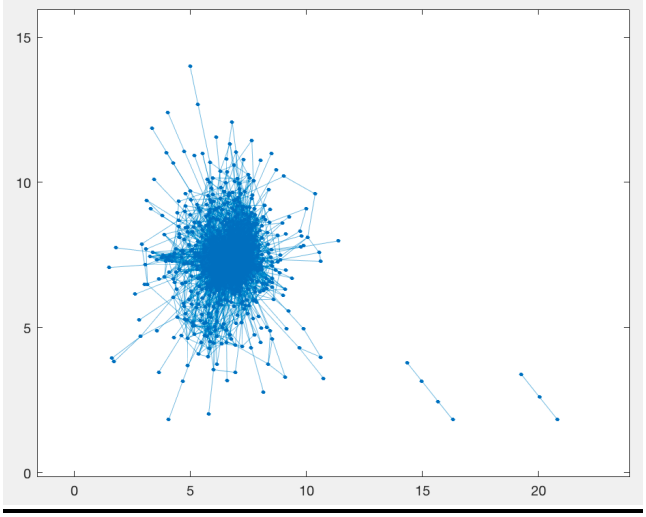
N. Explore your graph! You can scroll over individual nodes to see what words are connected.



O. Repeat step 7 with the coOccurrenceNetwork.gexf. How do the networks differ?

8. <u>Visualize/analyze the network in MATLAB.</u>

A. Please see semanticNetworkAnalysisReadme.pdf for further instructions.



More details

1. **Word vector embeddings model**
   Pre-trained model from http://vectors.nlpl.eu/explore/embeddings/en/models/
   Removed all "universal part of speech" tags from the corpus

   =====
   General model information:
   dimensions: 300
   window: 5
   iterations: 5
   vocabulary size: 296630
   id: 3
   ====
   Training corpus:
   ===
   description: English Wikipedia Dump of February 2017
   url: https://dumps.wikimedia.org/
   tool: Wikipedia Extractor
   case preserved: True
   stop words removal: NLTK
   id: 2
   tokens: 2252637050
   lemmatized: True
   tagset: UPoS
   tagger: Stanford Core NLP v. 3.6.0
   NER: True
   public: True
   ====
   Training algorithm:
   name: Continuous Skipgram
   url: https://github.com/RaRe-Technologies/gensim
   tool: Gensim
   version: 2.1
   command: None
   id: 0

Math underlying word2vec: https://arxiv.org/pdf/1411.2738.pdf,
Nice visualization/demonstration of some toy training data: https://ronxin.github.io/wevi/
Google's original word2vec paper: https://arxiv.org/pdf/1301.3781.pdf