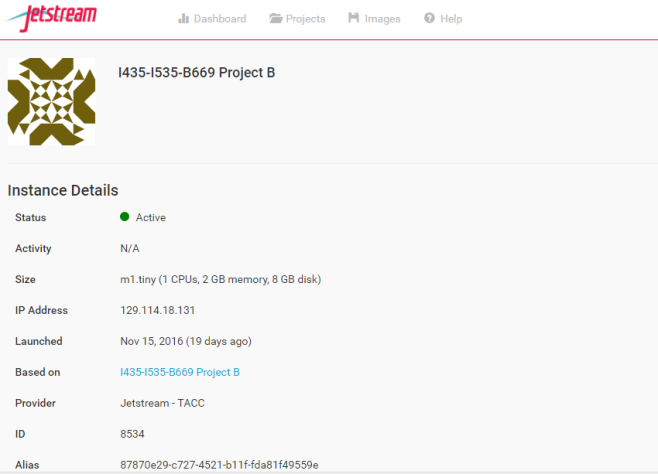
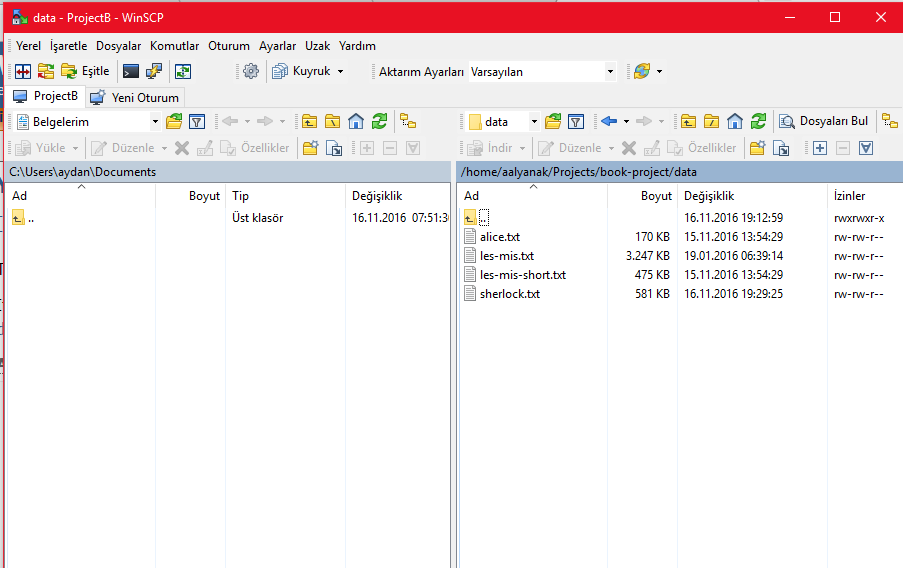
**I535-O Project B: Analyzing Large Text Corpora**

**Setting Up Your Environment**

1. **Use of Jetstream**

I started to this project with setting up my Jetstream instance. I used “*I535-I435-B669 Project B”*image.

I used Winscp for data transfer, and Putty as the web shell like I did in ProjectA. I didn’t find the webshell of Jetstream very handy, didn’t prefer to work in it.

**Figure 1**: The Jetsream Instance and Winscp I completed my project in.

1. **Set Up Directories and Download Code**

I started with creating Projects folder and cloning the git folder inside:

*$ git clone* [*https://github.com/dimitargnikolov/book-project.git*](https://github.com/dimitargnikolov/book-project.git)

Then, I needed to enter into book-project folder, and copy the sample start-up script:

*$ cp sample.bashrc ~/.bashrc*

*$ source ~/.bashrc*

Finally, I installed pymongo with following code:

*$ cd ~/Downloads*

*$ git clone git://github.com/mongodb/mongo-python-driver.git pymongo*

*$ cd pymongo*

*$ sudo /opt/anaconda/bin/python setup.py install*

**Case Study: The Characters Network for Les Miserables**

1. **Adding a Book to MongoDB**

In Project Gutenberg, you can find plain text of free e-books, audio books, CD and DVD projects, mobile reader devices and etc. In the scope of this project, we’re starting by downloading Les Miserables e-book in Plain Text UTF-8 format to data folder:

*$ wget* [*https://www.gutenberg.org/files/135/135-0.txt -O ~/Projects/book-project/data/les-mis.txt*](https://www.gutenberg.org/files/135/135-0.txt%20-O%20~/Projects/book-project/data/les-mis.txt)

Then, we need to insert this book into MongoDB by using python with the following code:

*>>> import pymongo  
>>> from pymongo import MongoClient  
>>> mongodb = MongoClient()  
>>> db = mongodb.projectB  
>>> with open('les-mis.txt', 'r') as f: text = f.read()*

After reading the les-mis.txt, we need to insert it to MongoDB and add some identifiers to the book which will allow us to search the book later on:

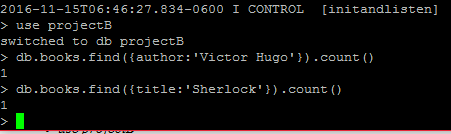
*>>> db.books.insert({'author': 'Victor Hugo', 'title': 'Les Miserables', 'text': text})*

By inserting the plain text of book, author and title, you can doublecheck it by exiting python and logging in mongoDB. You need the following code:

*$ mongo  
> use projectB  
> db.books.find({author:'Victor Hugo'}).count()*

Then it’ll give you the count depending on how many books you’ve inserted into mongo, by Victor Hugo. In my case, I get 1 count for the author “Victor Hugo”.

I’ve also downloaded “Sherlock” by Arthur Conan Doyle from Gutenberg Project for extra Minimal Plus portion.



**Figure 2:** Finding the books we’ve imported into MongoDB

1. **Extracting the Characters from Books**

After importing, we again need to login to python, to tag texts in books. Since I’ve two books inside of Mongo, i need to create 2 result sets:

>>> mongo\_results = db.books.find({'title': 'Les Miserables'})  
>>> mongo\_results2 = db.books.find({'title': 'Sherlock'})

>>> from lib import \*  
>>> tagged\_texts = tag\_texts(mongo\_results)  
>>> chars = find\_people(tagged\_texts) #stands for Les Miserables  
>>> tagged\_texts2 = tag\_texts(mongo\_results2)  
>>> chars2 = find\_people(tagged\_texts2) #stands for Sherlock

1. **Inferring Character Relationships**

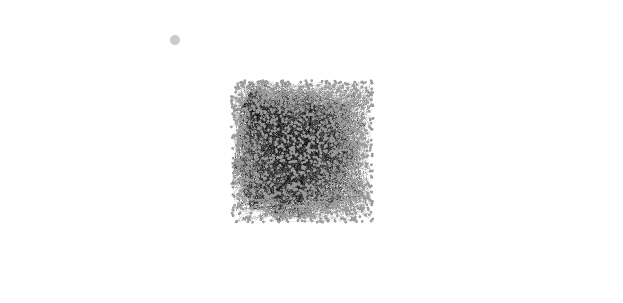
Final step in jetstream is creating the network representation by a window of 15 characters (I’ve also tried to create a N=10 window size for Sherlock):

>>> network = create\_network(tagged\_texts, chars, N=15)   
>>> network2 = create\_network(tagged\_texts2, chars, N=10)

>>> import networkx as nx  
>>> os.makedirs('networks')  
>>> nx.write\_gml(network, os.path.join('networks', 'les-mis.gml'))  
>>> nx.write\_gml(network2, os.path.join('networks', 'sherlock\_10.gml'))

1. **Analyzing and Visualizing the Network of Characters**

I’ve created 3 gml files and now it’s time to do the data cleaning. The first version of les-mis.gml has 1417 nodes and 4488 edges, which makes the visualization nearly impossible.



**Figure 3:** The visualization of gml before the data cleaning

For this purpose, we need to do the data cleaning first. I’ve followed the instructions in the videos. I’ve merged several nodes together, changed the French characters with Search/Replace button in Data Library.

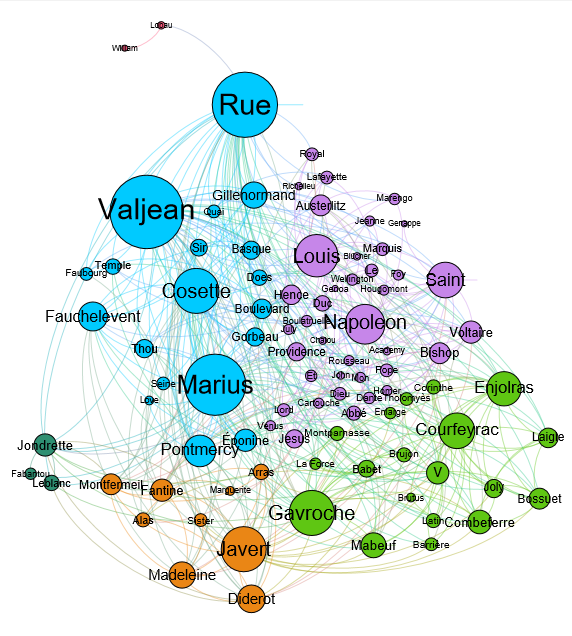
The ASCII code – character conversion list can be found at:

<http://www.petefreitag.com/cheatsheets/ascii-codes/>

By this way, I got rid of all the invalid characters and replaced them with correct values. At the end, I had 1183 nodes and 3232 edges. Which is still a lot.

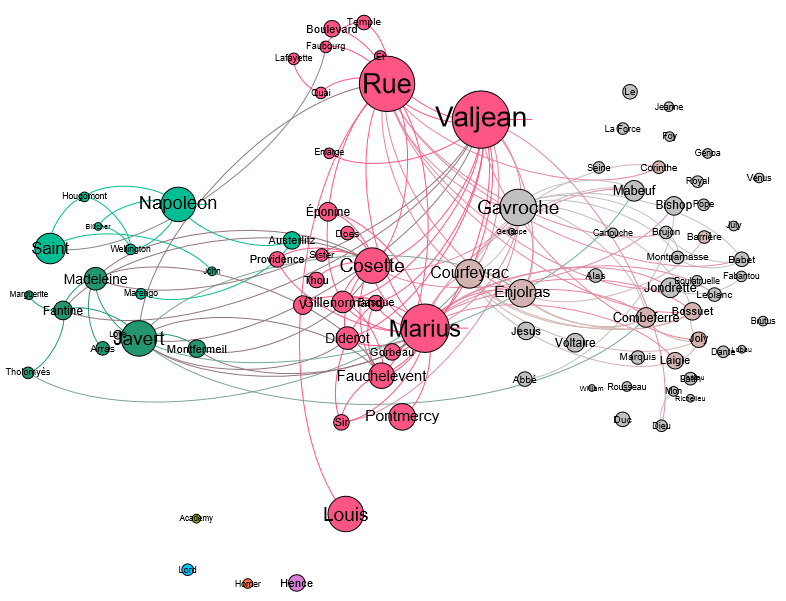
I cleared the nodes under a degree of 10 for visualization and cut the node count down to 91. Calculated the modularity as 0.388 and average path length as 3.118.

At the end, my visualization looked like this:



**Figure 4:** The final visualization of Les Miserables

Since this visualization has too many edges and looks crowded, I’ve also tried to visualize by deleting the edges under a weight of 3.

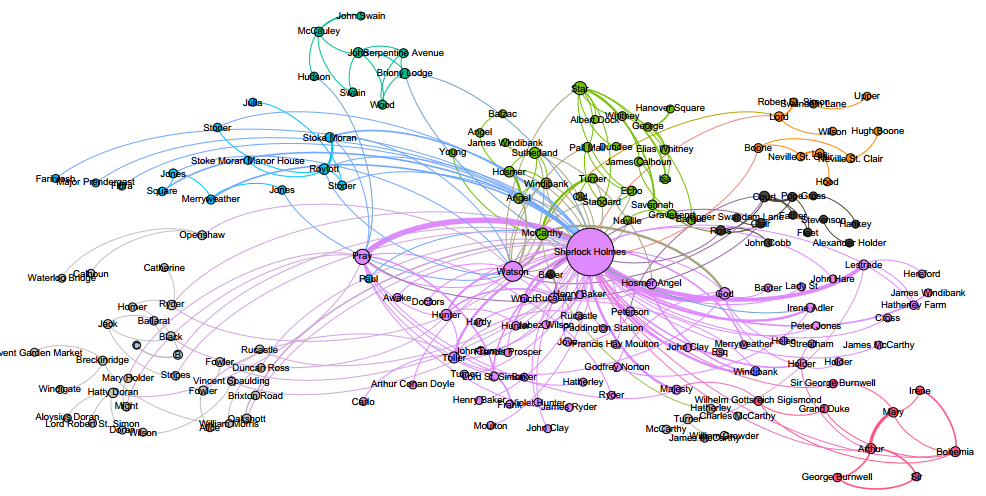


**Figure 5:** The visualization of Les Miserables under clearing some edges

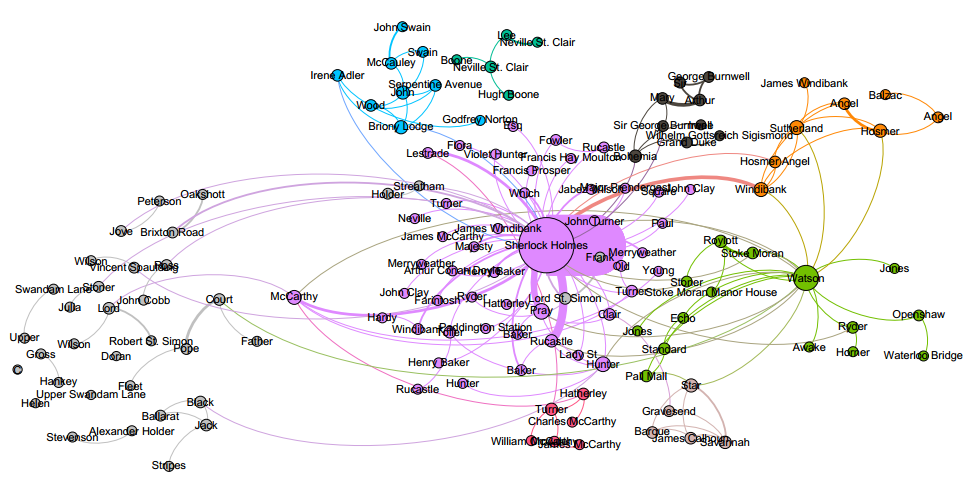
With this visualization, I’ve losted my modularity groups and had lots of unconnected nodes. Which is why, I didn’t continue with this. I’ll continue with the graph in Figure 4.

1. *Is a window of size 15 a good window size for the characters that you think are related?*

I also did an analysis (for Minimal Portion) of Sherlock to see the difference between size 10 and 15. The difference can be seen in the graphs:



**Figure 6:** Visualization graph of Sherlock in window size 15



**Figure 7:** Visualization graph of Sherlock in window size 10

I don’t know about Les Miserables, but for Sherlock, I know that Sherlock Holmes and Watson are the most related characters. N=10 made them the most important nodes of two different groups (purple and green) but in N=15, they’re the 2 biggest node of purple group. In that case, I can say that N=15 is a good size for the related characters.

1. *What are the strengths and weaknesses of a larger window size? Give an example of a relationship that was missed because of a window size of N=15*

Window size 15 creates more nodes and edges which is making the graph crowded. It’s not obvious that which are the connecting nodes inbetween modularity groups.

Because of this crowdedness, I’m missing the large relations between Valjean-Thenardier, Marius-Epinone, Valjean-Myriel (Myriel is an important node of a group in the assignment) and more.

1. Include a copy of the network graph (or portion of it) that you generated for the characters in Les Miserables from Gephi (PDF)

Please find the graphs in:

Aydan\_Alyanak\_ProjectB\ Les\_Mis\_N15.pdf

Aydan\_Alyanak\_ProjectB\ Sherlock\_N10.pdf

Aydan\_Alyanak\_ProjectB\ Sherlock\_N15.pdf

**RESOURCES:** The course materials, tutorials, videos, piazza discussions, the courseware of assignment, websites such as stackoverflow and google searches for script/code related consultancy.