Project Report - Management, Access, and Use of Big Data

Indiana University

Revision	Date	Description	Author
1.0	24-Nov-2016	Initial draft	Karthik Vegi
1.1	26-Nov-2016	Added description	Karthik Vegi
1.2	30-Nov-2016	Added Data Pipeline	Karthik Vegi
1.3	02-Dec-2016	Added visualization section	Karthik Vegi
1.4	04-Dec-2016	Review & Wrap up	Karthik Vegi

PROJECT DESCRIPTION

In this project, a book from Project Gutenberg is analysed and the text is visualized using Gephi. NLP techniques are applied on the data to discover the characters in the book and look at the strength of relationships between the characters.

Following are the steps that were carried out to analyse the data and visualize it:

1) Infrastructure Setup

- A new project is created on the Jetstream account
- A new instance is added to the project by choosing an image
- The instance is launched by setting the requirements for CPU and memory
- The instance is launched
- Once the instance is active, the commands are executed on the web shell
- After a secure layer is a setup between the VM and the local machine using Putty, WinSCP is used to securely transfer files back and forth.

2) Technologies Used

- Python
- MongoDB
- Gephi

DATA PIPELINE

As a part of the data pipeline, we shall be performing the following steps to analyse and visualize the text data.

1) Sourcing the Data

- In this project, we analyse two books: Les Miserables and Sherlock
- The book is downloaded from Project Gutenberg using the following command

```
$ wget https://www.gutenberg.org/files/135/135-0.txt -O ~/Projects/book-project/data/les-mis.txt
```

- The book is inserted into mongoDB using PyMongo extension

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

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

2) Extracting the characters from the book

- We automate this process by using a technique called named-entity recognition.
- This technique combines the best of both the worlds of natural language processing and statistical techniques.
- In this project the technique we used is the Python's nltk library
- The characters can be extracted from the book using the following command

```
>>> from lib import *
>>> tagged_texts = tag_texts(mongo_results)
>>> chars = find_people(tagged_texts)
```

3) Data cleaning Phase-1

- As with any data analysis project, we have a lot of junk values that should be cleaned.
- Cleaning is often not a one step process in big data projects, each phase will end with a data cleaning phase to ensure the data is improved at each stage of the data pipeline.
- As a first step, the most obvious characters that appear as junk could be removed from the characters.

```
>>> chars.remove('Jesus')
>>> chars.remove('Christ)
>>> chars.remove('Paris')
>>> chars.remove('Italy')
```

4) Exporting to Visualization Software

- Now that we have the data ready, we can format it and make it ready to be exported to Gephi, the visualization software we are going to use for this project.
- ***Lib.py*** will read the text, analyse the relationships and create a network
- **N** plays a crucial role in this python script as the N value decides the chunk of text that could be analysed at the same time.

```
>>> network = create_network(tagged_texts, chars, N=15)
>>> import networkx as nx
>>> os.makedirs('networks')
>>> nx.write_gml(network, os.path.join('networks', 'les-mis.gml'))
```

- We have the file ready to be exported to the Gephi visualization software

5) Getting started with Gephi

- Install Gephi and use the import wizard to open the **les-mis.gml** file
- The raw data version of the file after import looks as shown below. We shall be performing a lot of data cleaning and improve the visualization in the following steps.

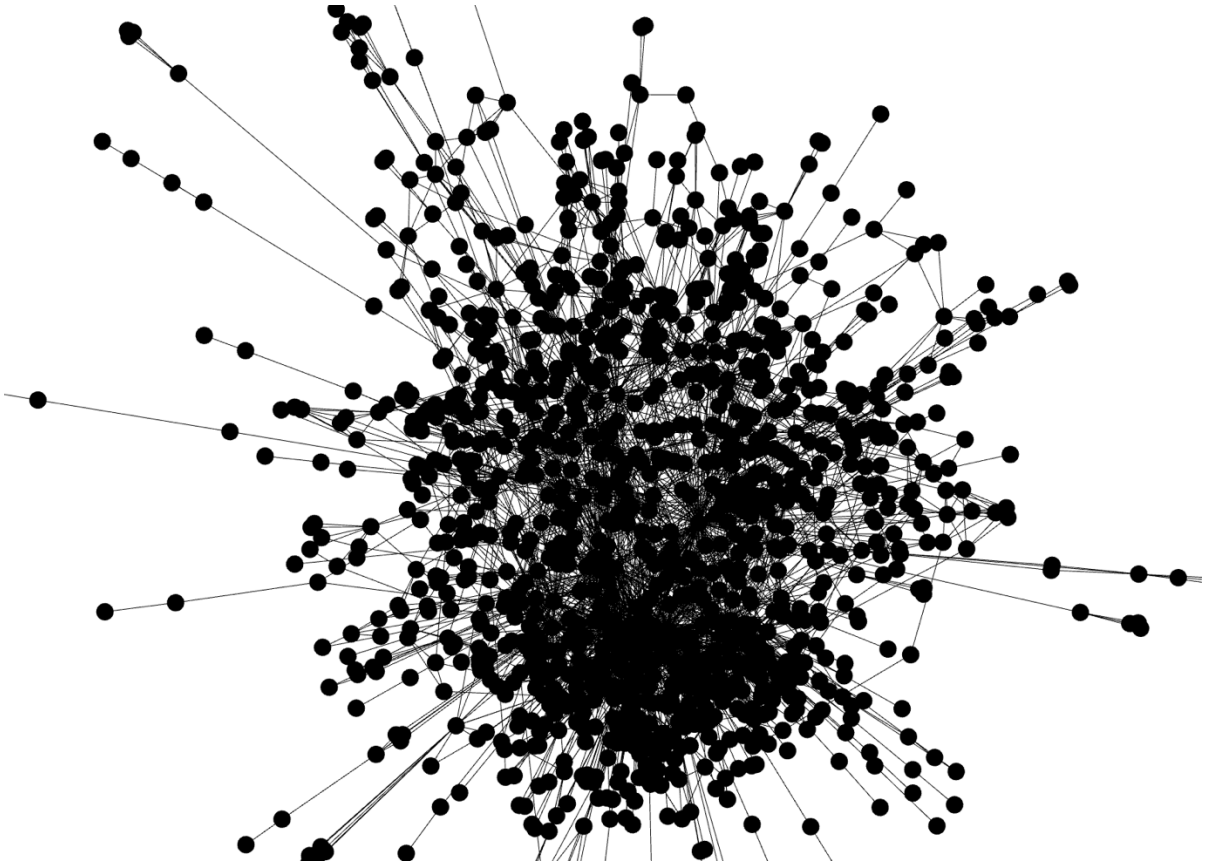


Fig: *Initial version before data cleaning with a lot of chaos*

6) Data Cleaning Phase-2

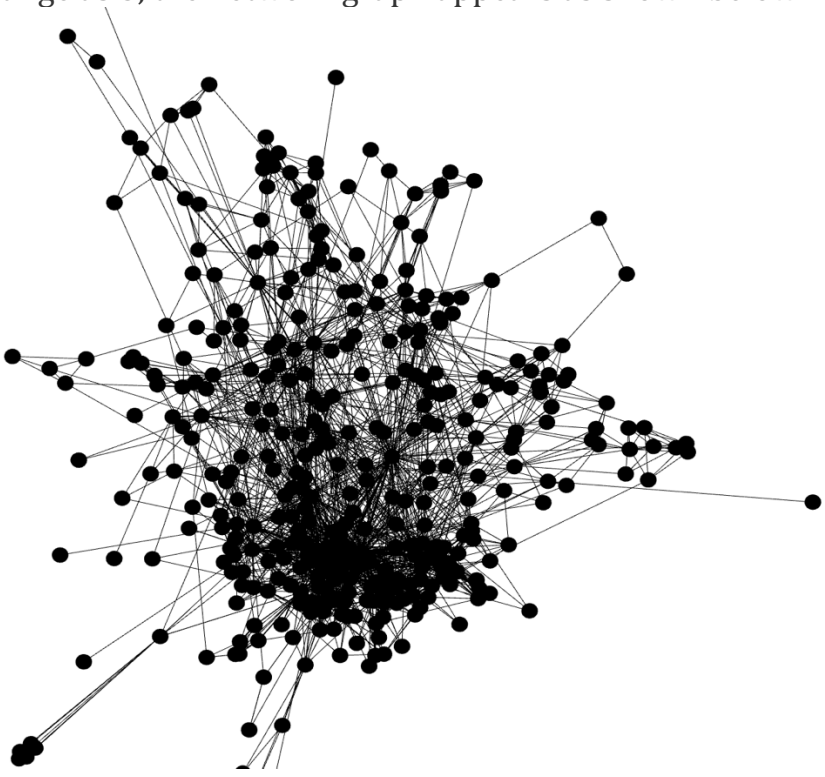
- We see a lot of junk characters in the data. We first remove the junk characters like #;* etc

Data Table

Nodes | Edges | Configuration | Add node | Add edge | Search/Replace | Import Spreadsheet | Export table

Id	Label
8.0	Bicêtre
17.0	Jérôme
25.0	Maubuée
28.0	Fougère
35.0	Thénardier
49.0	Benoît VIII.
64.0	Féletez
72.0	Barthélemy
93.0	Béthisy
95.0	Maître Renard
102.0	Edmond Géraud
118.0	Jérôme Bonaparte
122.0	Curé
128.0	My Lords Charles Brûlart
136.0	Paméla
140.0	Bérulle
145.0	Cîteaux
147.0	Paris Surêne
158.0	Anglès
159.0	Benoît
162.0	Pépin
172.0	Dædalus
187.0	Angoulême

- Applying degree range will remove the less relevant connections from our network graph. You can do this by using the Degree Range filter from the Topology section.
- Using degree range as 5, the network graph appears as shown below.



- This enables us to focus on the more important relationships in the text that we are trying to analyze

7) Data Integration

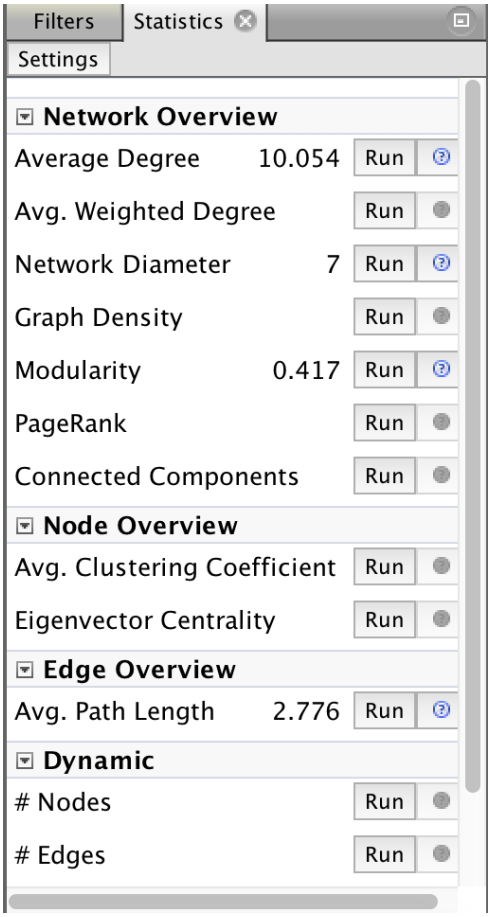
- We can see that a lot of names have been duplicated. For example, Valjean appears as Jean, Valjean which could be merged.
- We select the nodes that could be merged in Gephi and click on merge.

8) Node Centrality

- We use the between-ness centrality to look at the central nodes that are facilitating the other nodes.

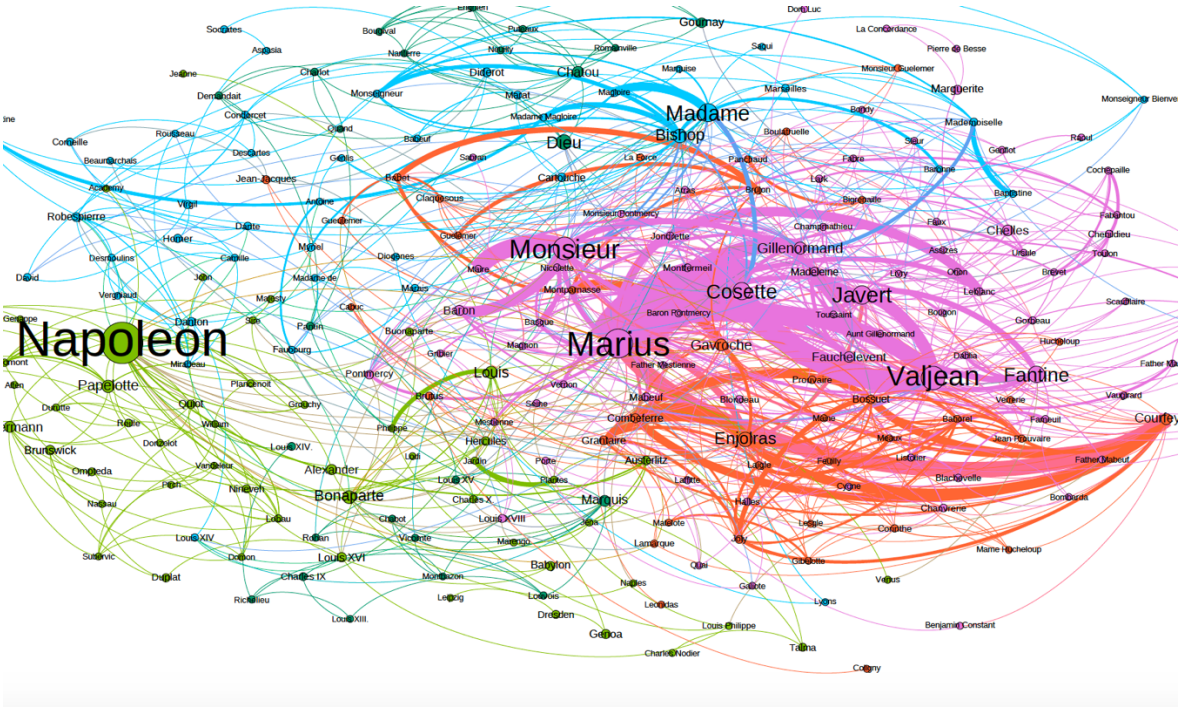
9) Communities

- We then identify the communities in the network to group closely related characters.
- We use the statistics palette to calculate the average degree and the network diameter.



10) Visualization

- Finally, we come to the visualization part of the analysis. After performing all the steps in the data pipeline, we end up with the final network of the characters in Les Miserables which looks are shown below:



LIST OF SOURCES

- MongoDB manual: https://docs.mongodb.com/manual/
- Jetstream user guide: http://www.jetstream-cloud.org/files/Jetstream_User_Guide-3-3-16-11am-Reduced.pdf
- Mongo import: https://docs.mongodb.com/manual/reference/program/mongoimport/
- Exporting data from MongoDB: https://docs.mongodb.com/v2.6/core/import-export/
- Gephi tutorial: https://gephi.org/tutorials/gephi-tutorial-quick_start.pdf
- Gephi video tutorial by Dimitar
- Python for MongoDB: https://api.mongodb.com/python/current/
- Gephi network analysis: https://blog.ouseful.info/2012/11/09/drug-deal-network-analysis-with-gephi-tutorial/
- Les Miserables Wiki: https://en.wikipedia.org/wiki/Les_Mis%C3%A9rables
- Discussed analysis with Megha and Bharat from the class

QUESTIONS

Minimal

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

- Yes, the window size of 15 is good enough to bring out the relations in the characters we are trying to analyze.
- Statistical NLP works based on a context vector which takes the count of the words in the window and matches its relation to the other words in the same window.
- The co-occurrences of the words could be easily captured with a window size that is not too less or not too high. Hence 15 is a decent window size.

(2) 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

A larger window size has its weaknesses and strengths as given below.

Weaknesses:

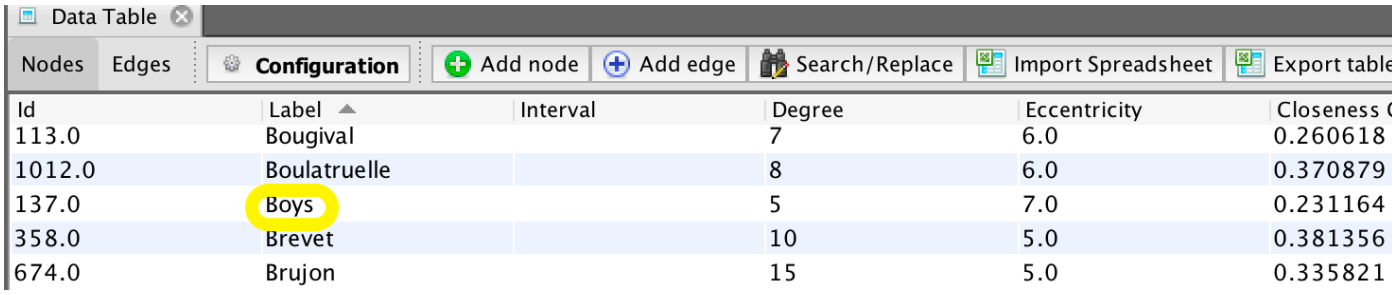
- A larger window means more words in the context vector of the NLP algorithm. This generates a lot of relations as more words come into the picture. The relations will be complex and lot harder to read and analyze.
- The network graph will have more nodes and the complexity of the network graph will be high

Strengths:

- Unknown relationships between the subtle nodes can only be captured with a larger window size. For example, if we are trying to analyze the text of a suspense thriller, the suspense character is mentioned as less as possible in the entire text to maintain the curiosity. This doesn't mean that the character is not important. It is indeed the prime character in the plot. A smaller window size cannot effectively identify characters like these which appear less. Hence a larger window size will help include these subtle characters and bring out a much complex relationship into picture.

Creating a network with a window size of 25, increased the no: of nodes in the graph but included some subtle characters that were missed. One character that was found

was **boys**. Per Wikipedia, the two unnamed youngest sons of the Thenardiers, whom they send to Magnon to replace her two dead sons.



Id	Label	Interval	Degree	Eccentricity	Closeness
113.0	Bougival		7	6.0	0.260618
1012.0	Boulatruelle		8	6.0	0.370879
137.0	Boys		5	7.0	0.231164
358.0	Brevet		10	5.0	0.381356
674.0	Brujon		15	5.0	0.335821

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

- The network graph is attached as a pdf in the zip file by the name ***les-miserables.pdf***

Minimal Plus

(4) When you analyzed texts of your own choosing that you're familiar with or interested in, did you glean any insights from this type of analysis that would be harder to glean from a simple read-through?

- When it comes to analysis in the real world, subject knowledge plays a crucial role to draw insights and help translate hunches to knowledge with the help of data. While working for **Les Miserables,** since we had no knowledge of the play and the characters that are a part of the play, data cleaning was very difficult task.
- Language barrier is also evident in this case as we couldn't differentiate between names, places and other things.
- When it came to analyzing Sherlock, background knowledge helped immensely to understand the relations and data cleaning.
- This reiterates the fact that domain knowledge is very important while doing analysis on data.

(5) Include a copy of the graph (or portion of it) that you generated for the characters in content you chose (PDF)

- The network graph is attached as a pdf in the zip file by the name ***sherlock.pdf***

(6) An archive containing the text(s) you chose to analyze in the second part of the project (ZIP)

- The archive has been added as an attachment in the zip file by the name ***sherlock-archive.csv***

Extra Credit

(7) When you extract the characters, create the network representation and apply the network analysis algorithms, there is some fine-tuning of the algorithms that needs to happen. Try exhaustively cleaning your list of characters, adjusting the parameter values for the length of the text window, or the number of communities. How do the results differ? Did you need to do a lot of fine-tuning to produce a visualization that was useful and easy to understand? What ways of automating this fine-tuning can you think of?

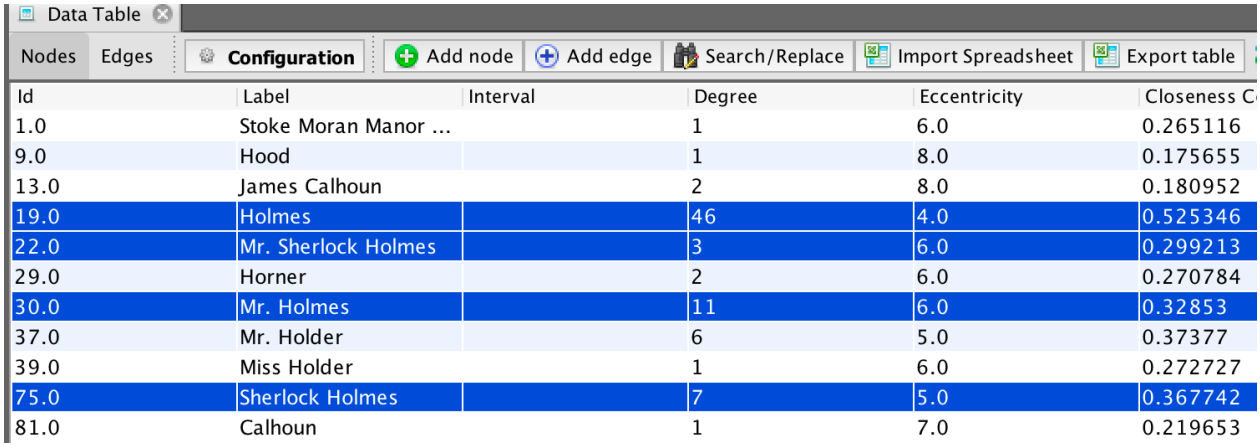
Analysis of Sherlock before and after cleaning:

Before Cleaning

- There were a lot of character which are not integral to the play and these characters only increase the network graph and make it less meaningful.
- The network graph was not easy to read; the relationships were hard to recognize and the main characters became less visible because there were a lot of relationships.
- Clearly, the raw data had a lot of chaos, the parameter values need to be adjusted and the network analysis algorithms should be run to bring out the important characters.

Steps of Cleaning and Fine-Tuning

- **Layout algorithms:** First we try to get the graph more appealing by running the layout algorithms namely Force Atlas, label adjust and no overlap. We also adjust the repulsion strength to maintain proper spacing. These steps increase the visibility of the graph.
- **Network Analysis:** The network analysis algorithms bring out the relationships between the characters. We apply network diameter to compute the average shortest path between the nodes, average degree to find the number of connections each node has and node closeness centrality to analyze how close a node is to the other given nodes. These algorithms bring clarity to the relationships between the characters.
- **Data Integration:** A lot of characters have been duplicated and merging nodes will make sure to avoid redundancy in the network graph.

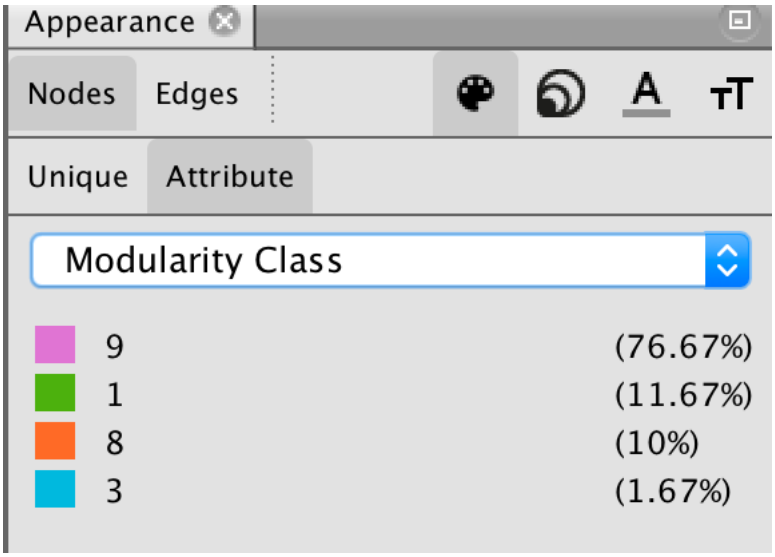


Data Table

Nodes | Edges | Configuration | Add node | Add edge | Search/Replace | Import Spreadsheet | Export table

Id	Label	Interval	Degree	Eccentricity	Closeness C
1.0	Stoke Moran Manor ...		1	6.0	0.265116
9.0	Hood		1	8.0	0.175655
13.0	James Calhoun		2	8.0	0.180952
19.0	Holmes		46	4.0	0.525346
22.0	Mr. Sherlock Holmes		3	6.0	0.299213
29.0	Horner		2	6.0	0.270784
30.0	Mr. Holmes		11	6.0	0.32853
37.0	Mr. Holder		6	5.0	0.37377
39.0	Miss Holder		1	6.0	0.272727
75.0	Sherlock Holmes		7	5.0	0.367742
81.0	Calhoun		1	7.0	0.219653

- **Communities**: Running the modularity statistic helps find closely related characters and forms communities. After finding the communities, we can assign a color to each community to make the communities more visible.



After Cleaning

- The prime character in the play, Sherlock Holmes comes to the forefront with all the important characters like Watson, Moriaty and Lestrade appear close to the prime character.
- The communities stand out making the relationships clearer.
- A lot of fine-tuning and help of network analysis algorithms was needed to bring out a clear analysis on the texts.

Ways to Automate Fine-tuning

- Shell scripts plays an important role in the automation process and we could use them along with a scheduler that runs the network algorithms and statistics one by one.

- If any errors occur, the script could send out an automated email for the quality analyst to consider the discrepancies and fix them.
- It should be noted that each analysis is different and the order of algorithms that we choose to apply differs. Utmost care has to be taken to ensure that we are applying a set of automated analysis only on related texts.

(8) Include a copy of the graph (or portion of it) that you generated for the characters in content you chose that went through the cleaning that you carried out (PDF)

- The network graph is attached as a pdf in the zip file by the name ***sherlock-cleaned.pdf***

(9) An archive containing the text(s) you chose to analyze in the second part of the project (ZIP)

- The archive has been added as an attachment in the zip file by the name ***sherlock-cleaned-archive.csv***