

# Report on ProjectB : Book Analysis

Jianwen Liu

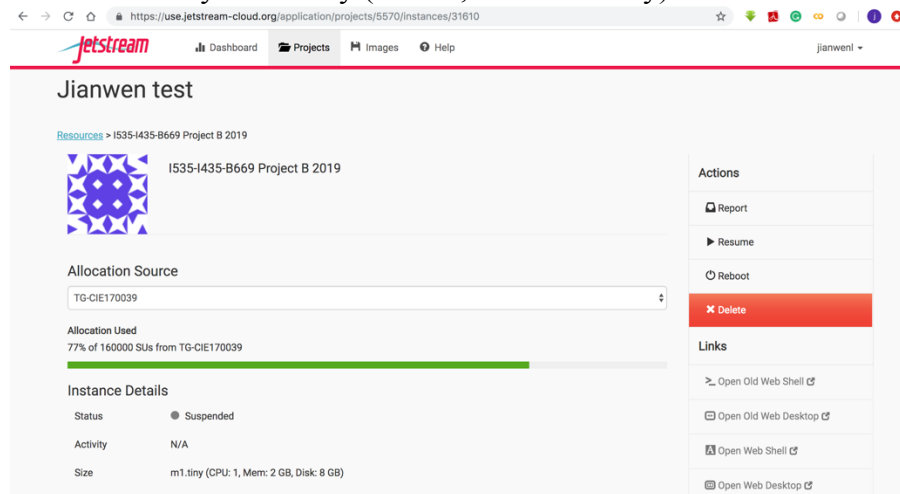
## 1.Introduction

An network visualization and analysis on information is the crucial. It is the first step for graph mining. Since knowledge graph is the heated topic in NLP, building a fundamental graph and implemented the further analysis on this graph for text is becoming important to the data scientists' work. With the Jetstream, NLTK and networkx, we built a network to visualize and analysis the relations among characters in books.

## 2. Implementation and description

### 2.1. Set up directories

Within the Jetstream, we create a project called Jianwen test, and select the image: 'I535-I435-B669 Project B 2019' and style: m1.tiny (1 CPU, 2 GB memory) as follows:



Then I set up the directory and used git to download the code:

```
$ mkdir ~/Projects
$ cd ~/Projects
$ git clone https://github.com/dimitargnikolov/book-project.git
$ cd book-project
```

Then I download the corpus for “les miserable” using external link, it turns out to be from the project Gutenberg corpus, which is a famous corpus:

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

### 2.2. Building pipeline

First, I verify the MongoDB authentication, for simplicity, I keep the keyword “i535y2019” for MongoDB:

```
$ mongo -u admin -p please_change_this_password  
  
> use admin  
  
> db.db.changeUserPassword("admin", "i535y2019")
```

### 2.3. Running Data Pipeline

we insert the corpus of “les-miserable.txt” file from project directory to the MongoDB with a Python script. Exactly, we use pymongo as wrapper to get connected to MongoDB:

```
>>> import pymongo  
>>> from pymongo import MongoClient  
>>> mongodb = MongoClient('mongodb://admin:i535y2019@localhost:27017/')  
>>> db = mongodb.projectB  
>>> with open('data/les-mis.txt', 'r') as f: text = f.read()  
... <PRESS ENTER HERE>  
>>> db.books.insert({'author': 'Victor Hugo', 'title': 'Les Miserables', 'text':text})
```

### 2.4. Extracting the Characters from a Book

Now, in the mongoDB, we used the nltk to extract characters for names.

1) firstly, we used a find MongoDB command to retrieve book in the database with the title Les Miserables and check it with “explains”:

```
>>> mongo_results.explain()  
  
{u'executionStats': {u'executionTimeMillis': 0, u'nReturned': 2, u'totalKeysExamined': 0, u'allPlansExecution': [], u'executionSuccess': True, u'executionStages': {u'needYield': 0, u'direction': u'forward', u'saveState': 0, u'restoreState': 0, u'isEOF': 1, u'docsExamined': 2, u'nReturned': 2, u'needTime': 1, u'filter': {u'title': {u'$eq': u'Les Miserables'}}}, u'executionTimeMillisEstimate': 0, u'invalidates': 0, u'works': 4, u'advanced': 2, u'stage': u'COLLSCAN'}, u'totalDocsExamined': 2}, u'queryPlanner': {u'parsedQuery': {u'title': {u'$eq': u'Les Miserables'}}}, u'rejectedPlans': [], u'namespace': u'projectB.books', u'winningPlan': {u'filter': {u'title': {u'$eq': u'Les Miserables'}}}, u'direction': u'forward', u'stage': u'COLLSCAN'}, u'indexFilterSet': False, u'plannerVersion': 1}, u'ok': 1.0, u'serverInfo': {u'host': u'i535-i435-b669-project-b-2019', u'version': u'3.6.11', u'port': 27017, u'gitVersion': u'b4339db12bf57ffee5b84a95c6919dbd35fe31c9'}}
```

2) Then we firstly used the lib packages and download the NLTK functions to local:

```
>>> from lib import *
>>> nltk.download('punkt')
>>> nltk.download('averaged_perceptron_tagger')
>>> nltk.download('maxent_ne_chunker')
>>> nltk.download('words')
```

we then used tag\_texts and find\_people to extract the people characters:

```
>>> tagged_texts = tag_texts(mongo_results)
>>> chars = find_people(tagged_texts)

>>> chars
```

The characters are as follows:

```
, u'Flora', u'Mongrais', u'Gindre', u'Saint Paul', u'Helen', u'Wellington', u'Madame', u'Saint Louis', u'Cadran', u'Barbari', u'Orl\xe9ans', u'Bossuet', u'Vincent', u'Gaster', u'Monsieur Mabeuf', u'Mother Pr\xe9sentation', u'Gourgaud', u'Night', u'Malebranche', u'Ponceau', u'Henri Fonfr\xe8de', u'Baruch', u'Madame Dubarry', u'Saint Peter', u'Monsieur Bombarda', u'Buonaparte', u'Virgil', u'Vancouver', u'Bastide', u'Phyllis', u'Joly', u'Venice', u'Sirven', u'Louis XVIII', u'Queen Isabella', u'Louis Blanc', u'M\xe9gisserie', u'Celtic', u'Madame Cottin', u'Hernani', u'Regnier', u'Tholomy', u'Arcis', u'Benvenuto Cellinis', u'Adamastor', u'Garden', u'Genflot', u'Christi', u'Ennius', u'Dismas', u'Ponsonby', u'Marat', u'Th\xe9nardier Jondrette', u'Bibles', u'Nassau', u'Sir Hudson Lowe', u'Delille', u'Attila', u'Coligny', u'Porte', u'Champeaux', u'Savoyard', u'Jean Valjean', u'Beaumarchais', u'De Chappedelaine', u'Naheury', u'Baron Marius', u'Henquinez', u'Lake Lauzet', u'Terentius', u'Ponine', u'Urbain Fabre', u'Ludwig Snyder', u'Pedasus', u'Puyraveau', u'Vignes', u'Sabinus', u'Soult', u'Bavoux', u'Picpus Street', u'Pierre', u'Mother Gibou', u'Ave Maria', u'Blancard', u'Cracovie', u'Sister Agatha', u'Charles Myriel', u'Il', u'Io', u'Faux', u'Gondren', u'Skill', u'Levis', u'Buonaparte', u'Femme', u'Fiden
```

3) finally, we used remove function to clean the raw data. We remove “A”, “Madame” and “Paris” as shown because they mean “A” is for article, “Madame” is for “madam” in French and “Paris” is just the name of the city.

```
, u'Ronde', u'Mathieu', u'Arthurs', u'Nicolette', u'Marseilles', u'Lutzen', u'Add', u'Fauvent', u'Caius Gracchus', u'Chouan', u'Exodus', u'Le N\xe9fite', u'Garat', u'Quiot', u'Mars', u'Tarsus', u'Opponach', u'Below Cosette', u'Labarre', u'Mademoiselle Dog-lack-name', u'Nuncio', u'Pirch', u'Virgin', u'Andr\xe9 Chenier', u'Monseigneur', u'Whither', u'Horace', u'Jean Prouvaire', u'Will Boulatruelle', u'Mavot', u'Ramponneau', u'Vousiergue', u'Mademoiselle Mars', u'La Haie-Sainte', u'Vertigo', u'Gilbert de Por\xe9a', u'Diana', u'Naviguer', u'Bahorel', u'Diane', u'Guillons', u'Alg\xe9sir', u'Plautus', u'Campan', u'Dom Mabillon', u'Tenait', u'Merlonus Horstius', u'Dict\xe9e', u'C\xe9sar', u'Fantine', u'Leipzig', u'Monsieur Pabourgeot', u'John Brown', u'Sonnerie', u'Camille', u'Vergniaud', u'Junot', u'J\xe9rome Bonaparte', u'Robespierre', u'Hermogenus', u'Mademoiselle Cosette Fauchelevent', u'Crillon', u'Fontenoy', u'Letter Slang', u'Redivivus', u'Rio Maior Marques', u'Monsieur de D', u'Baronne de T.', u'Fatigue', u'Tr\xe9sor', u'Arnauld', u'Marie', u'Blacheville', u'Milton', u'Lesgueules', u'Sabot', u'Terreur', u'Bombarda', u'Bauduin', u'Mathurins', u'Contra Gracchos Tiberim', u'G\xf6tzberg', u'Richefeu', u'Azelma', u'Nivernais', u'Perrault', u'Mary', u'Antipope Gregory', u'Majesty', u'Alba', u'Plancenoit', u'Spinoza', u'Delancey', u'Pelagius', u'Place de', u'Ditch', u'Montebello'])
>>> chars.remove('A')
>>> chars.remove('Madame')
>>> chars.remove('Paris')
>>>
```

## 2.5. Inferring Character Relationships

The first step we implemented is to create a network representation for the characters. Thanks to the function in lib, we can directly apply create\_network on char:

```
>>> network = create_network(tagged_texts, chars, N=15)
```

we then save the the representation of network as a gml file, with networkx for further visualization and network analysis:

```
>>> import networkx as nx
>>> os.makedirs('networks')
>>> nx.write_gml(network, os.path.join('networks', 'les-mis.gml'))
```

the gml was like:

```
graph [  
  node [  
    id 0  
    label "Valjean"  
  ]  
  node [  
    id 1  
    label "Fantine"  
  ]  
  ...
```

## 2.6. Transfer the gml file back to local

Finally, we use the command:

```
scp username@ipadress:/home/username/Projects/book-project/networks/les-mis.gml  
/Users/jianwenl
```

in local terminal to transfer the gml file back.

## 3. Network visualization and analysis

In this section, we use gephi2 for network visualization and analysis.

### 3.1. Data exploration

We firstly check in the data laboratory:

Id	Label	Interval	Eigenvector Centrality	Modularity Class
169	Anacephorus		0.000854	9
170	Madame Everybody		0.004463	7
171	Bibles		0.00598	11
172	Fabre		0.103802	12
173	Monsieur Pontmercy		0.184201	7
174	Le Baron Marius Pontmercy		0.026588	7
175	Chelles		0.101272	7
176	Th&#233;odule		0.09831	7
177	Bernis		0.005272	11
178	Pierre de Bruys		0.001467	6
179	Frederick		0.021531	1
180	Chanverie		0.228213	12
181	Keksek&#231;a		0.039428	4
182	Tarsus		0.007756	9
183	Jacques		0.000796	2
184	Charras		0.011468	14
185	Ballets		0.116471	12
186	Has Monsieur		0.007199	1
187	Thouars		0.013439	0
188	Donzelot		0.013432	14
189	Colbert		0.003073	14

we can find out the nltk actually made many mistakes in the find\_people process. 1) the “Madame Everybody” is actually a general inference; 2) the “Bibles” is the name of the book; 3) the “Th&#233;odule” is some name but we can only get recognized part of the name; 4) “Has Monsieur” is a combination of words but not name.

### 3.2. Data cleansing

Since NLTK used to recognize variation of a name as different person, we firstly check the replication of names.

Id	Label	Modularity Class
250	Gribeauval	
510	Valjean	
536	Monsieur Valjean	
571	Chevalier	
604	Jean Valjean	
664	Bougival	
1204	Valsin	
1207	Valjean's	

As is shown above, there are many “Valjeans” (‘Valjeans’, ‘Valjeans’, ‘jean Valjeans’ and ‘monsieur Valjeans’) in the the network, however, they are the same person. Thus, we use the “Merge Nodes” option to merge these nodes within the network.

We also need to clean some node which are obviously not names:

Gephi 0.9.2 - Project 1

Overview Data Laboratory Preview

Workspace 1

Data Table

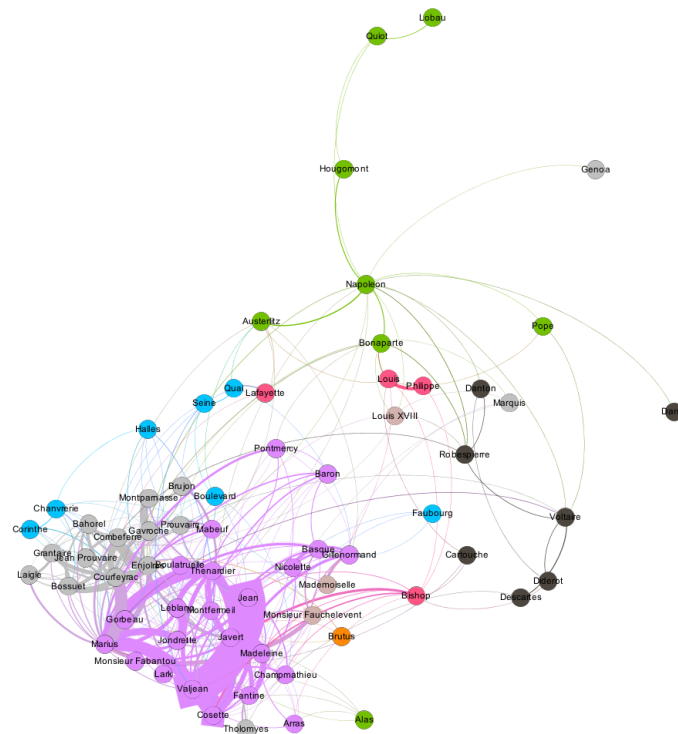
Nodes Edges Configuration Add node Add edge Search/Replace Import Spreadsheet Export table More actions Filter: mis Label

Id	Label	Interval
1	Misery	
66	Mister	
358	Mother Mis&#233;ricorde	
872	Les Mis&#233;rables	
933	Miss	
1008	Mamselle Miss	
1046	Miserables	

Add column Merge column Delete column Clear column Copy data to other column Fill column with a value Duplicate column Create a boolean column from regex match Create column with list of regex matching groups

On the figure shown above, we can find out that mister and miss are obviously not some particular people. In addition, “Misery” is just an adj but not names. We need to delete these kinds of nodes.

Finally, we get the graph for the network in “les miserable”:



we can find out that Valjean, Mabeuf etc are the main characters in the book. The relations among Valjean, Javert and Thenardier is very strong. It turns out my graph is little different to the showcase in project. The may results from the degree range and weight range filter difference. My graph shows some back ground characters like Napoleon and Pope.

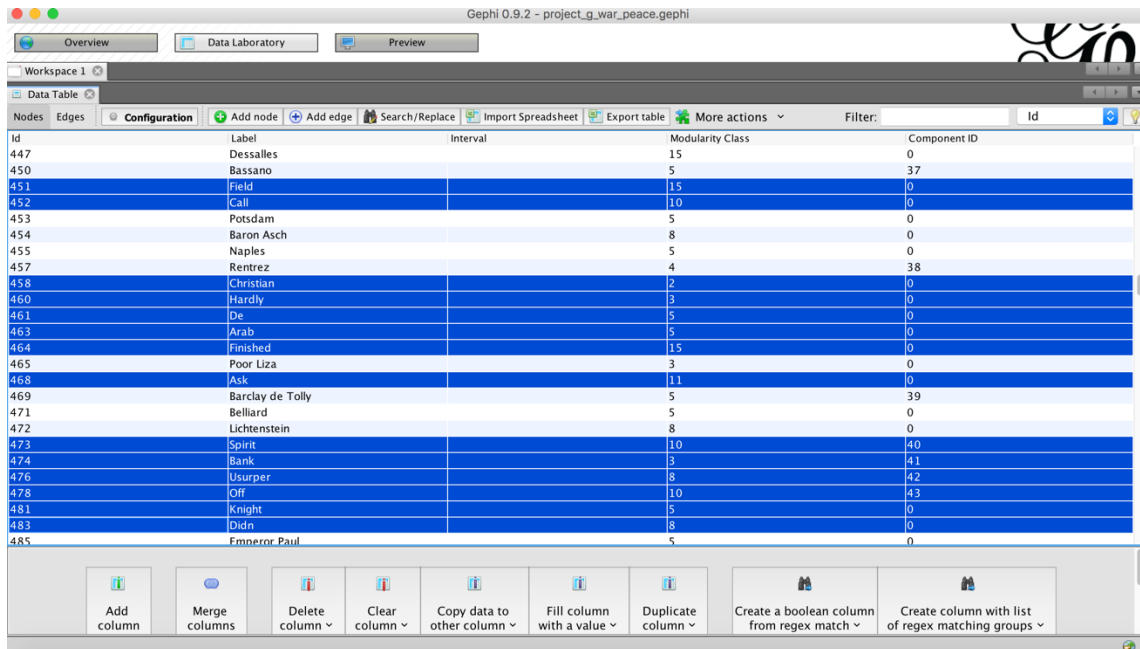
#### 4. Network Visualization and Analysis on “War and Peace”

Now we tried to do network visualization and analysis on other texts. Here, I decided to take Leo Tolstoy’s famous book “War and Peace” as corpus. The original text is also from the Project Gutenberg. We used wget approach to download it into the instance and then do the analysis steps on it for the character network.

The steps are similar with what we did for les misérable:

1. Insert your text(s) in MongoDB.
2. Write a find query to retrieve all the texts from MongoDB.
3. Pass the results of the query to the tag\_text and find\_people functions to find any people mentioned in the text.
4. Clean up the list of people using the remove command
5. Use the create\_network function to create a network representation of the relationships between people in the texts.
6. Use Gephi to find important characters and groups of similar characters in the network.

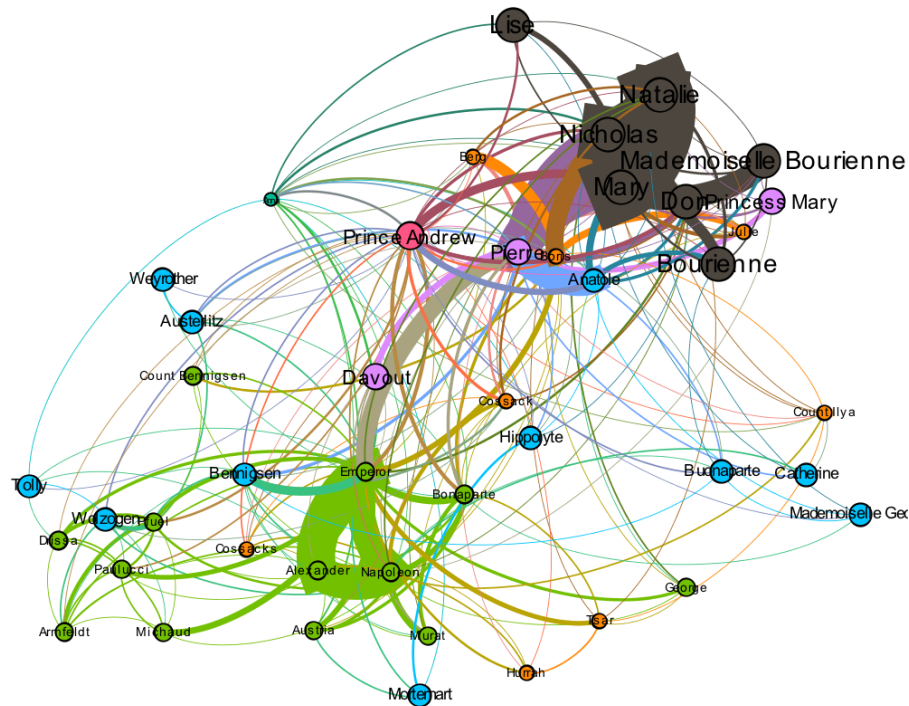
After open the gml file in gephi, we can found out similar situation for the data cleansing:



Id	Label	Interval	Modularity Class	Component ID
447	Dessalles		15	0
450	Bassano		5	37
451	Field		15	0
452	Call		10	0
453	Potsdam		5	0
454	Baron Asch		8	0
455	Naples		5	0
457	Rentrez		4	38
458	Christian		2	0
460	Hardly		3	0
461	De		5	0
462	Arab		5	0
464	Finished		15	0
465	Poor Liza		3	0
468	Ask		11	0
469	Barclay de Tolly		5	39
471	Bellard		5	0
472	Lichtenstein		8	0
473	Spirit		10	40
474	Bank		3	41
476	Usurper		6	42
478	Off		10	43
481	Knight		5	0
483	Didn		5	0
485	Emperor Paul		5	0

It is obviously that many characters, like “Spirit” and “Hardly”, cannot be the name of people in the book. Thus, we need to remove them.

Finally, we get the network graph for the characters in “War and Peace” as follows:



we can find out in the graph that characters are separated into different clusters. The Main characters like “Natalie”, “Prince Andrew” and “Lise” are the top characters I the graph. They have many relations to others.

#### 4. Answers to the Project Questions

Minimal:

1. Is a window of size 15 a good window size for the characters that you think are related? Why and why not?

Choosing 15 as the window size for the characters is actually a good choice. The reason that we choose 15 in this project is that even though the book itself is incredibly long (365 chapters and 1900 pages), the average length of sentence is very close to 15 words. In fact, if we check other books, the length is very similar. For instance, “Game of throne” has a length of 17. We choose 15 as a window size to capture the network relationship. This is a hyper-parameter in the model.

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  $N=15$



The strength of larger window size is that we can capture more relations in the book which names do not appear near. On the other hand, the weakness for the larger window size is that we actually will build a graph with much more edges. Considering the nodes (represent the people are the same), a bigger window size will capture more relations between nodes. The problem for that is we may consider some in fact not close people into their relation.

An example of a relationship that was missed because of a window size of  $N=15$ , is the case in long conversation. As we know, the window can only seize the relations within 15 chars, however, this will miss the information when some character has a long conversation. For instance, when A spoke three or four sentences in a time, which is a usual case, the  $N=15$  window will lose the information to whom he spoke in the conversation. This contradicts to the fact that people in the dialogue should be much closer.

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

We presented the network graph as above mentioned.

Minimal plus:

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

An insight I can glean from the network analysis is that we find out the importance of "napoleon" and "empire" in the book of "War and peace". As we know, war and peace is novel describing a story in Russia from 1805 to 1820. This story mainly focuses on four families in Russia: the Bezukhovs, the Bolkonskys, the Rostovs and the Kuragins. By a simple read-through, we can easily find out main characters like Natasha and Andrei just like described on the graph.

However, it is only through the graph that we can know history figures like Napoleon and Tsar have a so important role in this book. It is not only as a background of the story, but even surpass some families of the fours, which we used to think is the most important role. In fact, if you are familiar with War and Peace, you will figure out the transition of "War" and "Peace" in the story, this may indicate that the story have more focus on the macro history than we thought.

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

We presented the network graph as above mentioned.

3. An archive containing the text(s) you chose to analyze (ZIP).

We download the “War and Peace” by Leo Tolstoy texts file from the Project Gutenberg. The link is as follows:

<http://www.gutenberg.org/files/2600/2600-0.txt>

Extra Credit Option:

1. 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?

The result differs when we fine tuning the hyper parameters. I gave some main process and fine-tuning as follows and the results graph is in question 2.

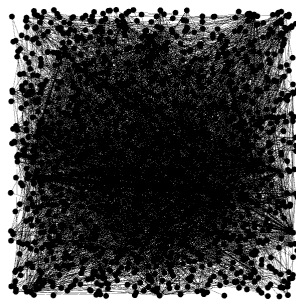
Firstly, we check the network data in the data laboratory in gephi2. We can find out that the data in graph is too complicated and can not be analyzed. This is because there are so many nodes (1026) and edges (4705) in the original network.

We used the Giant component to filter those component with most nodes and applied the ForceAtlas 2 algorithm to get a good layout. To get the cluster, we run modularity class on the graph to get a colored figure. We have tuned the resolution of modularity classes. It turns out that higher resolution gave us less cluster.

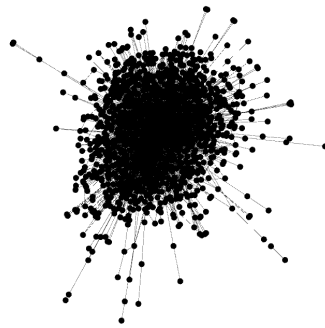
Most importantly utilize Degree range to get those nodes within a range of degree. We also tried degree range 4 and 9, respectively, to test the layout. The degree range 9 gave us less nodes and edges, which

In general, we need to do a lot of fine-tuning to produce a visualization that was useful and easy to understand. One way of automating this fine-tuning I can think of is to give a targeting ratio for shown nodes and edges. By tuning parameters like range degree and weight, we are actually filtering the nodes to some particular extent. If we can direct give a ratio, this may help. Another way is to auto removing some particular characters, like location and “Monsieur”, if we can use other model to filter the nodes name and recognize those real names, that will help us for data cleansing.

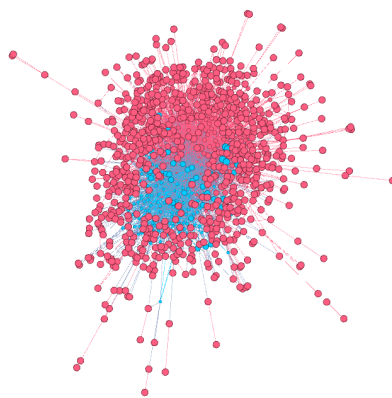
2. Include a copy of the graph (or portion of it) that you generated for the characters in content you chose that went through the cleaning that you carried out (PDF)



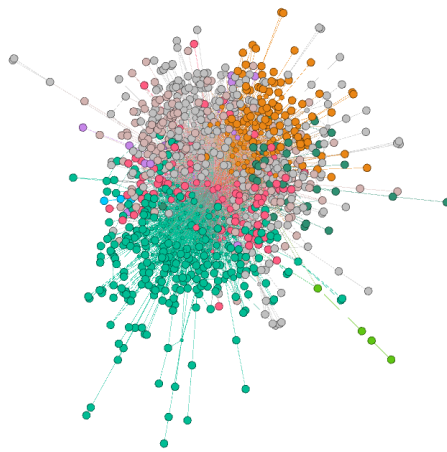
Original graph



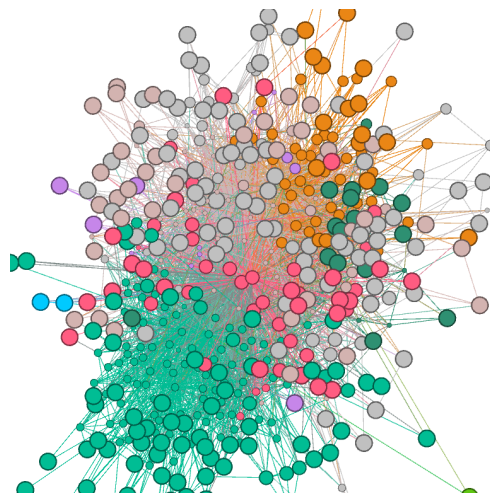
After ForceAtlas 2



Modularity class with resolution 2



Modularity class with resolution 1



Degree range 4



Degree range 9

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

We still used the “War and Peace” by Leo Tolstoy texts file from the Project Gutenberg. The link is as follows:

<http://www.gutenberg.org/files/2600/2600-0.txt>

## 5.Conclusion

By building the network and the corresponding analysis on the book, we find out that network analysis on characters is a much better way to understand the relations of characters in text. By applied pymongo, we can easily store and retrieve the data in MongoDB. We can also use NLTK to extract the name of people in text and build a network representation on it. Finally, by utilizing Gephi, we can visualize and analysis the network relation easily. In addition, the data cleansing is an very important step for network analysis, it directly affects the visualization and analysis result.