Social and Technological Networks
Project 9 - Social networks of Fiction

Robin Dupont - s1579039

1 Introduction

Generally speaking, social networks analyses are made on real data set, such as Facebook fiends, email exchanges or phone calls. However, it is also possible to analyze a fictional social network, taken from a book. Most of the time, these analyses are made through a language processing analyses. Here, the aim is to realize this analysis with the minimal use of a language processing tool, but by analyzing the extracted network. Hence, the first step is to create this network and represent it as a graph. Then, determine which are the mains characters of the fiction, their relationships and determine rough communities inside this network.
Determining the main characters of a book and the graph of the general relationship from a text can be useful for artificial intelligences. Machines and robot could gain a deeper understanding of complex situations by achieving a text based analysis of their environment and obtain a way to represent relationships between human beings.

2 Related Work

There are a few publication of this topics. Here are two of them :

— - J. Seo, G-M. Park, S-H. Kim, H-G. Cho, 2013. Characteristic Analysis of Social Network Constructed from Literary Fiction.
In this paper the four researchers give hints to extract a social network and make some standard analysis and comparison with real life social networks
— J. Seo, G-M. Park, S-H. Kim, H-G. Cho, 2013. Complex System Analysis of Social Networks from Literary Fictions.
In this other paper, the authors gives another way of quantizing interactions between characters and a first approach for determining core characters, by network analysis as well as language processing.

These two papers are important because they gives a simple way to extract a social network from a literary fiction. However, the analysis is not set toward the relationship understanding. In this report we will use the network to get a better understanding of characters' relations.

3 Extracting the social network

3.1 Distance between characters

N.B. : All the analysis have been realized with Python. For details, see the code and the guideline PDF.

The first step of this analysis is to extract the social network and represent it as a graph, where each node is a character and edges are relationships between characters. First problem is to know if there should be an edge between to nodes. In order to know that, we need to find a way to quantize the relationships between the characters and determine a threshold. If the interaction is stronger than the threshold, we create an edge between the two characters. If not, we don't.

As we don't want to use a language processing unit for the analysis, we will base the relationship analysis on characters' name proximity in the text. We can make the simple hypothesis that : if two people are related to each other, their names are going appear several times in the text close to each other. For example in any Harry Potter books, we can read that kind of sentences :
"Ron and Hermione squeezed together to give Hagrid enough space to join them." or
" "Hagrid!" said Harry loudly."
These samples shows that related characters have their names close for each other. Knowing that, we can establish a way to quantize interaction between characters.
To quantize the interaction we use the method described in Characteristic Analysis of Social Network Constructed from Literary Fiction. First let consider the text as the entity T. T is composed of sentences $\langle S_i \rangle$, which can be subdivided into words $\langle w_{i,j} \rangle$. To sum up, we can write down :

$$T = \langle S_1, S_2, S_3, \ldots \rangle$$
$$S_i = \langle w_{i,1}, w_{i,2}, w_{i,3}, \ldots \rangle$$

We also define C_T which is the set of characters present in the text T. C_T is composed of characters c_i. So we can write :

$$C_T = \{c_1, c_2, c_3, \ldots\}$$

For each character c_i, we save the index of the sentence where it appears. So, let be $POS_T(c_i)$ define as :

$$POS_T(c_i) = \{j \mid c_i \in S_j\}$$

Now, we can define our interaction between characters. We use the following formula :

$$Interaction(c_i, c_j) = \sum_{l,m} \alpha^{|l-m|}$$

Where $l \in POS_T(c_i)$ and $m \in POS_T(c_j)$.

As suggested in Characteristic Analysis of Social Network Constructed from Literary Fiction, the value of α can be chosen, in so far as $\alpha < 1$ in order to guarantee that the interaction will drop with the distance. Here we chose $\alpha = 0,5$. In order to avoid too much interactions, we don't take into account l and m if $|l - m| > \beta$. Here again, beta can be chose. We chose $\beta = 10$.
Now that we have the interaction, we can define the distance between two characters by this formula :

$$dist(c_i, c_j) = -\log_2\left(\frac{Interaction(c_i, c_j)}{|POS_T(c_i)| + |POS_T(c_j)|}\right)$$

Where $|POS_T(c_k)|$ is the cardinal of $POS_T(c_j)$.

Thanks to that formula, we have a value that decreases when the interaction increases, and moreover the interaction generated by a lot of occurrences in the text is counterbalanced by the denominator.

N.B. : If the interaction is so small that the computer considers it as 0, the distance is infinite between the two characters, and therefore, we consider them as not related at all.

3.2 Creating the graph

Now that we have the distance between all the characters we can generate a graph where the nodes are the characters. To create an edge, we decide that if the distance between two characters is above a certain threshold.

$$\exists\ edge(c_i, c_j) \text{ if } dist(c_i, c_j) < \gamma$$

Unfortunately there is no universal value of γ and the value have to be adapted for each studied book. If γ is too small, the graph will miss some edges to fully represent the social network. If γ is too large, the graph will contain too much small and polluting edges.
Then we create two graphs, both non oriented. The first graph has non weighted edges, and the second has. The first one is used to determine communities as well as hub score for nodes. And the second one is used for identifying the main characters through a minimal spanning tree.

3.2.1 Non weighted graph

As mentioned before, if the distance between two characters, is smaller than γ, an edge between the two characters is created. And therefore, the two nodes created if they do not exist.

Once the graph is created, we can compute the communities thanks to a python library that uses the Louvain Modularity. This algorithm identify strongly connected components of a graph as a community. Detecting communities in our case, allows us to see which characters is friend with which. And therefore, identify friends groups.

Then we can calculate the hub score of each character. To do so, the algorithm gives to each nodes a weight of $1/n$ where n is the number of nodes. Then the weight is equally shared to connected neighbors. In so far as this algorithm converges, we finally obtain a (approximated) hub score for each node.
This score represent the social integration of the node in the network. The higher the score is, the better his integration is. Therefore, he is supposed to be an important character in the book, in so far as the writer might have build the social network around him.

3.2.2 Weighted graph

To build the weighted graph, we use the same principle as the non weighted one. Except that for each created edge, we give it the distance between the two characters as weight.
With this graph we can determine the minimal spanning tree (MST) thanks to the Kurskal algorithm. This greedy algorithm takes all the edges of the weighted graph and sort them from the lowest weight to the highest. Then it picks edges to create an edge set that connects all the nodes, avoiding cycles.
The MST is interesting because all the characters close to the root are supposed to be the most important one. To identify the core characters, we take the nodes which has the highest hub score and we look at the nodes that are directly connected to him in the MST.

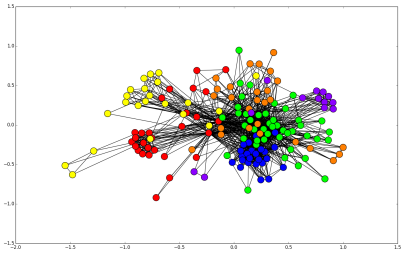
4 Applying the theory

in this section we will apply the theory on several books which are :
— *Les Miserables* (LM)
— *Harry Potter and the Philosopher's Stone* (HP)
— *Bel Ami* (BA)
— *Don Juan* (DJ)

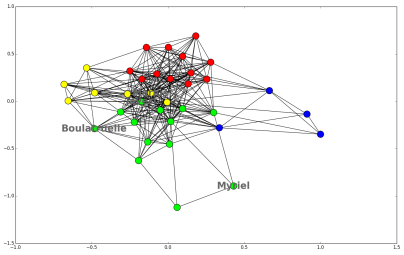
4.1 The Graph - Figure 1

The first thing to do is to generate the graphs in order to have a first insight of the social network. Here are the social networks of the four books :

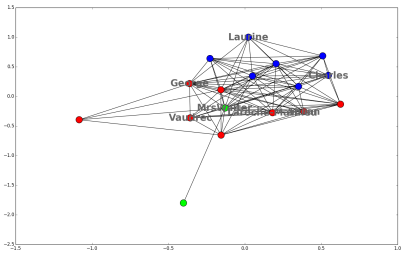
We can see the different colored communities communities. In some books such as DJ, the communities are quite good because they separate the peoples accordingly to their status in the book (people in yellow, who disagree with Don Juan. In green who are close to him. In red, who want to kill him. In blue, some peasants that he tried to seduce).
In HP, the communities are sometimes good (because the villains such as Draco, Crabbe and Goyle are in the same). But they also makes mistakes. For example, Harry, Ron and Hermione are not in the same communities.
For the two other books, and in general (except for DJ) communities are too hazardous to base the full analysis on it, though some results are quite good.



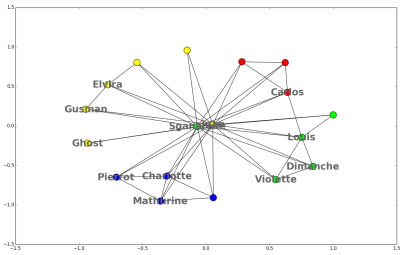
(a) HP graph



(b) LM graph



(c) BA graph



(d) DJ graph

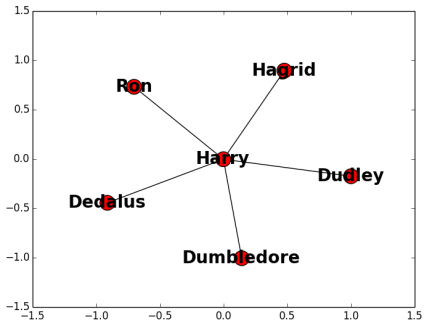
FIGURE 1 – Graphs

4.2 The core MST - Figure 2

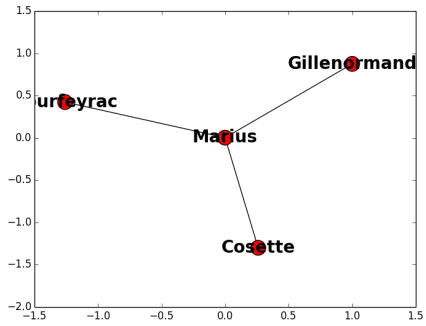
Here, we give only the core of the MST (which center is the node with the highest hub score as explained before)

For each book, exect BA, the center node is the main character. The surrounding nodes which are those directly connected in the full MST, are most of the time the main characters. But sometimes (such as Dedalus for HP) some very anecdotal characters happens to be directly linked to the hero.

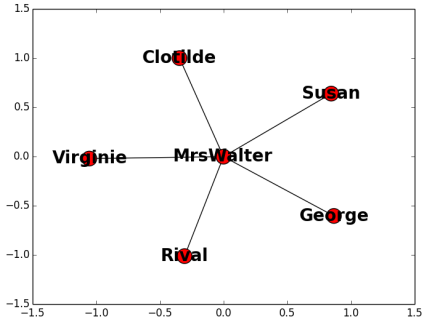
In the case of LM, Valjean does not appear in the main characters, which is surprising. But we can explain it because He is strongly bound to Cosette, as Marius. So the distance between Marius and Valjean is bigger than the sum of the distance Marius - Cosette and Cosette - Valjean. That is why the MST considers Valjean as a secondary characters.



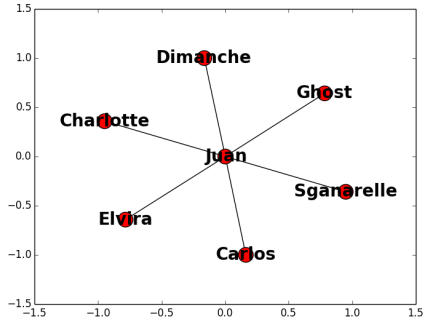
(a) HP MST



(b) LM MST



(c) BA MST



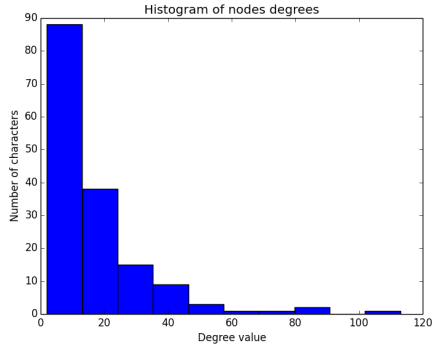
(d) DJ MST

FIGURE 2 – MST core

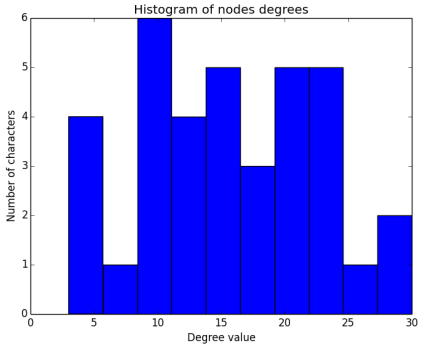
4.3 Degree Distribution - Figure 3

In this section we will have a look at the degree distribution.

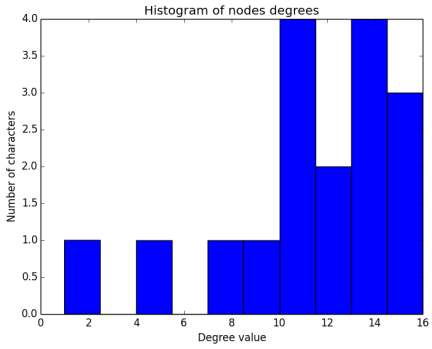
Human social networks would normally follow a power law distribution. It's to say that the number of nodes having a degree d is proportional to $\frac{1}{d^\alpha}$.
In that case, The degree distribution seems to vary a lot according to the book. Only the HP book seems to have a power law -like distribution. This can be explained by the fact that books are small ecosystems, where everybody roughly knows everybody. HP is the book with the more characters, which tends to recreate a more realistic word. More over there are a lot of secondary characters, which helps to create the peak toward the origin.



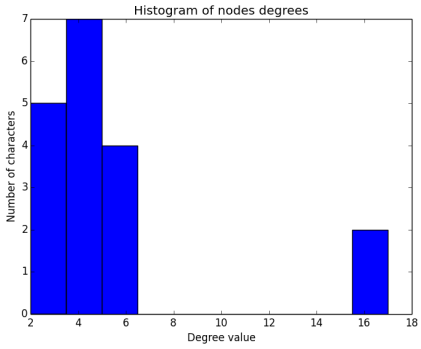
(a) HP Histogram



(b) LM Histogram



(c) BA Histogram



(d) DJ Histogram

FIGURE 3 – Histograms

5 Conclusion and possible improvements

The extraction of a social network based on the name proximity rather than the language processing, give surprisingly good results for a such simplistic approach. However, these results are not perfect. But I think most of the imperfection is due to two factors.

The first one is the name recognition. Ah characters can be designated by several ways, this is hard to catch all the apparitions of a character. Or sometimes, characters change their name (such as in BA or LM). So to create a more precise network, a way is to lay the stress on a more precise and accurate name recognition.

The second one is friendship or enmity. When to names are close, there is a strong chance that they are related, but we do not know if they like each other. To do so we should analyze words like emotions in between, to give a sharpest weight to edges.

With all this improvements, the graph should be more accurate, and there should be less gross mistakes.