Ca' Foscari University of Venice

Network Analysis Project

Network Analysis of Game of Thrones Characters

Student:

Federico Ceccato

Mat. 886734

Academic Year 2021-2022

Introduction

In this project I tried to analyse a dataset about the books of Game of Thrones. The purpose is to understand if it's possible to know the importance of the characters and their evolution based on the number of interactions that they have during the narration.

The dataset is composed by 4 columns, the "source" is the first character name that appears in a sentence and the "target" is the second name within 15 words from the first. The third column is the "weight" which tells us how many times that interaction occurs. The fourth is useless because it tells us the book number and in our analyses we'll use only the first and the second book in separate cases.

I think that a brief explanation of the Martin's world is necessary. The island on the left side is Westeros where all the main characters live and the capital is King's Landing. On the right side we have the island of Essos where in the early books we have only Daenerys Targarien and Drogo as main characters.

Figure 1: Game of Thrones Map

EDA

First Book

I started with a simple Exploratory Data Analysis.

In the first book we have 139 unique values for source and 143 for target. The distribution of interaction is right skewed with a mean of 11 and a median of 5. The majority of weights stand between 0 and 15. After that I wanted to see the characters with the most number of interactions, so I group by "source" and then by "target" and sum the number of interactions. The interesting thing I notice is that from one list to another there are some differences. I explain this thing thinking of a structure of a book, usually in a sentence the name of a character that is active in that moment is the first the appears. Instead the second name could be present in that moment and can interact with the first character or can be only mentioned. So I decide to use as label of the network draw the characters of cloumn "Source" with the highest weights and I add Tyrion Lannister because he has a very high number of interactions as a target.

Second Book

I do the same analysis on the second book to see any differences.

The dataset structure is the same as the first one. This time we have more sources and targets, 193 and 186. This means that we have more characters or more characters interactions. Also in this book the distribution of interactions is right skewed, the median is 5 but the mean is 8, less than the other book. This time is clear that the way we assume that an interaction will occur could be not correct, because we know for sure that Robert Baratheon and Eddard Stark died in the the first book and they have a lot of interactions as target in the second book. This can

be misleading because we know for sure that they were very important in the first book and also in the second they were mentioned a lot but at the end they don't interact with anyone. Unless we are in presence of flashbacks, but I think there are too much interactions to justify this result.

Analysis

First Book Network

Now we start with the network analysis. I started creating the network and drawing it, to see the complexity of it.

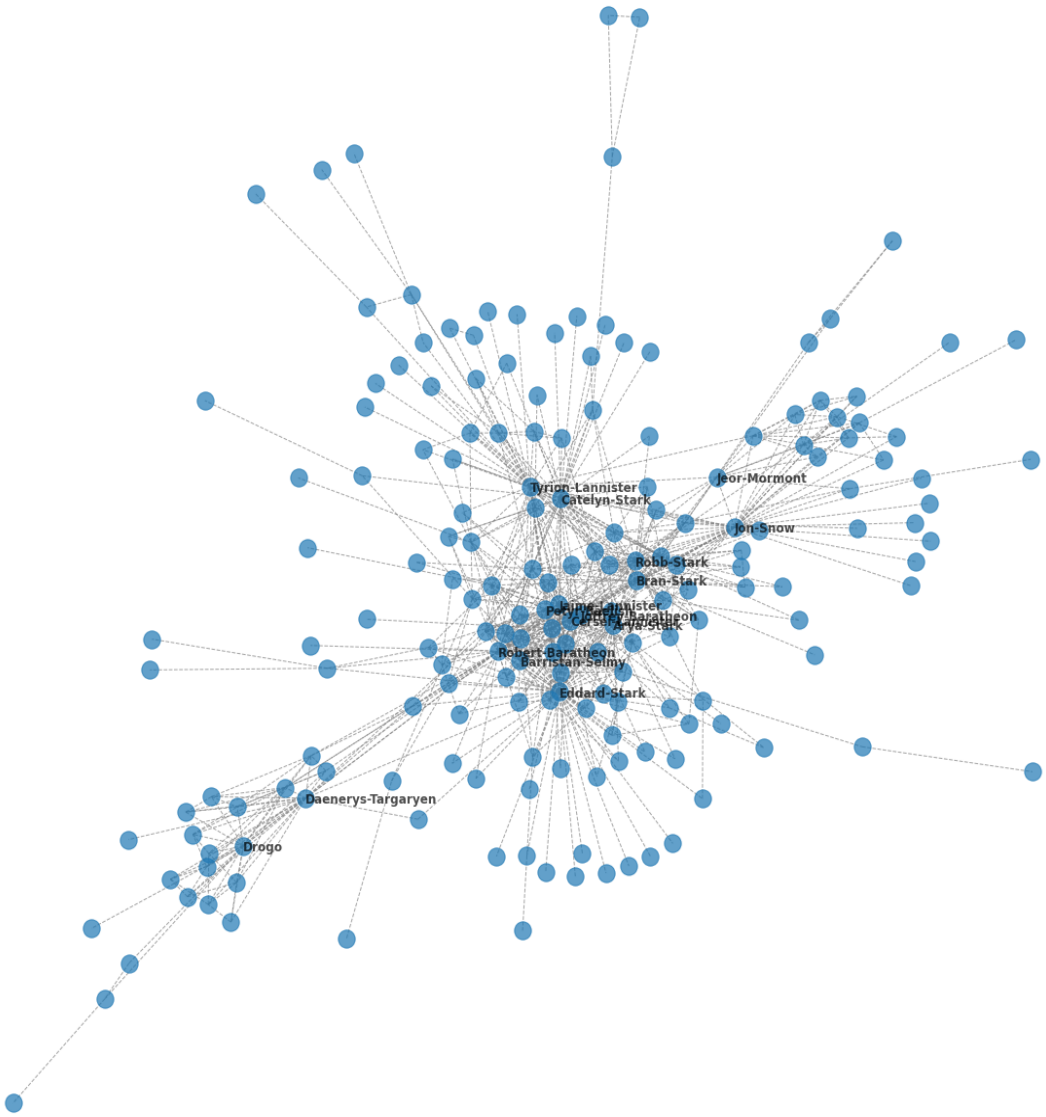

Figure 2: Network Plot of Book 1

The main features of the network are:

- Number of nodes: 187
- Number of edges: 684
- Average degree: 7.3155
- Network density: 0.03933

We also see a dense central part where we have the nodes with the highest interactions. But we can also see how the geographic position of each characters influence the interactions.

In fact on the bottom left we have Drogo and Daenerys that are in an isoleted region, the same for Catelyn Stark, Tyrion Lannister and for Jon Snow also in another region. Instead in the center of the graph we have the carachters in the main region and the ones with the majority of interactions. From this picture we can say that the plot of the network can correspond to the reality.

Second Book Network

I did the same thing for the second book and see the differences.

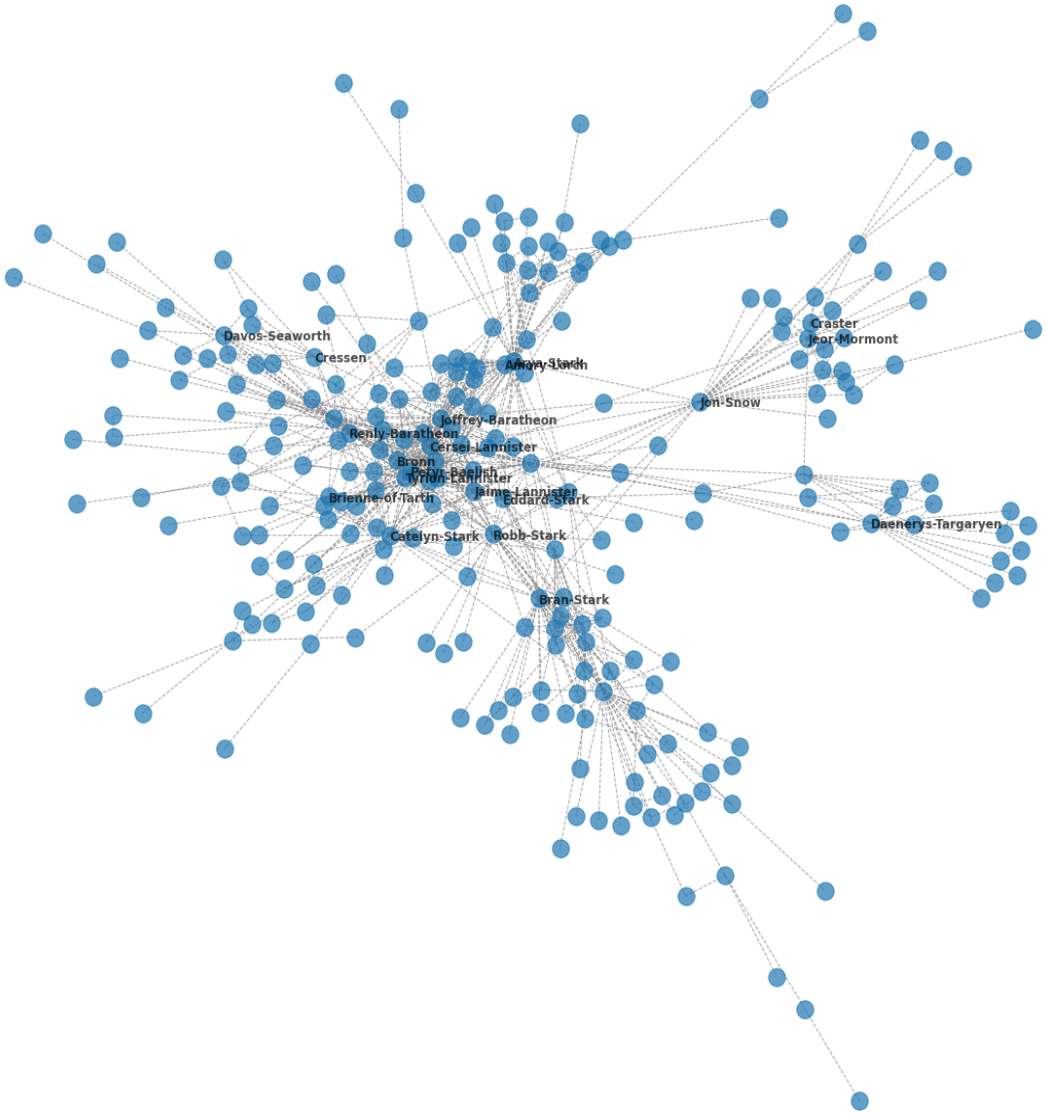

Figure 3: Network Plot of Book 2

The main features of the second network are:

- Number of nodes: 259
- Number of edges: 775
- Average degree: 5.9846
- Network density: 0.02319

The network has a larger number of nodes and edges than the first one but the density is lower. As in the first one we have different zones, we can say that there are some characters that are in a sort of close environment, but we'll investigate it later.

Metrics

After this first part I decided to use some metrics to understand the importance of the characters.

The metrics are:

- Degree Centrality: which is defined as the number of links incident upon a node.
- Betweennes Centrality: is the average length of the shortest path between the node and all other nodes in the graph.
- Eigenvector Centrality: is a measure of the influence of a node in a network.
- Closeness Centrality: quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.

First Book

The results for the first book are very similar for each measure.

Degree Centrality:

- 'Eddard Stark', 66
- 'Robert Baratheon', 50
- 'Tyrion Lannister', 46

Betweenness Centrality:

- 'Eddard Stark', 0.26960
- 'Robert Baratheon', 0.21403
- 'Tyrion Lannister', 0.19021

Eigenvector centrality:

- 'Eddard Stark', 0.29640
- 'Robert Baratheon', 0.26948
- 'Sansa Stark', 0.23155

Closeness centrality:

- 'Eddard Stark', 0.56363
- 'Robert Baratheon', 0.54545
- 'Tyrion Lannister', 0.51098

As we can see the most important characters for each measure are Eddard Stark, Robert Baratheon and Tyrion Lannister. This is in line with the storyline of the book.

Second Book

The results for the second book are not so similar for each measure. We can explain this fact with the fragmentation of the story for many characters that begin an own story in a separate geographic region.

Degree Centrality:

- 'Tyrion-Lannister', 53
- 'Joffrey-Baratheon', 47
- 'Cersei-Lannister', 43

Betweenness Centrality:

- 'Arya Stark', 0.18811
- 'Jon Snow', 0.17443
- 'Robb Stark', 0.16494

Eigenvector centrality:

- 'Joffrey Baratheon', 0.30736
- 'Cersei Lannister', 0.29520
- 'Tyrion Lannister', 0.28264

Closeness centrality:

- 'Robb Stark', 0.47777
- 'Eddard Stark', 0.45744
- 'Robert Baratheon', 0.44869

I think that the most reliable measures are the degree and eigenvector centrality because they capture the evolution of these three characters that in book 2 have high importance.

Community Detection

Thinking on the map of the Martin's world I taught it could be interesting implementing a community detection to find hidden relations among the nodes in the network. To do this I decided to apply different methods to find the best partition.

Louvain

Louvain is an unsupervised algorithm divided in two phases: Modularity Optimization and Community Aggregation. It's very useful because we don't have to specify the number of communities a priori. Below we see the two communities detection for each book.

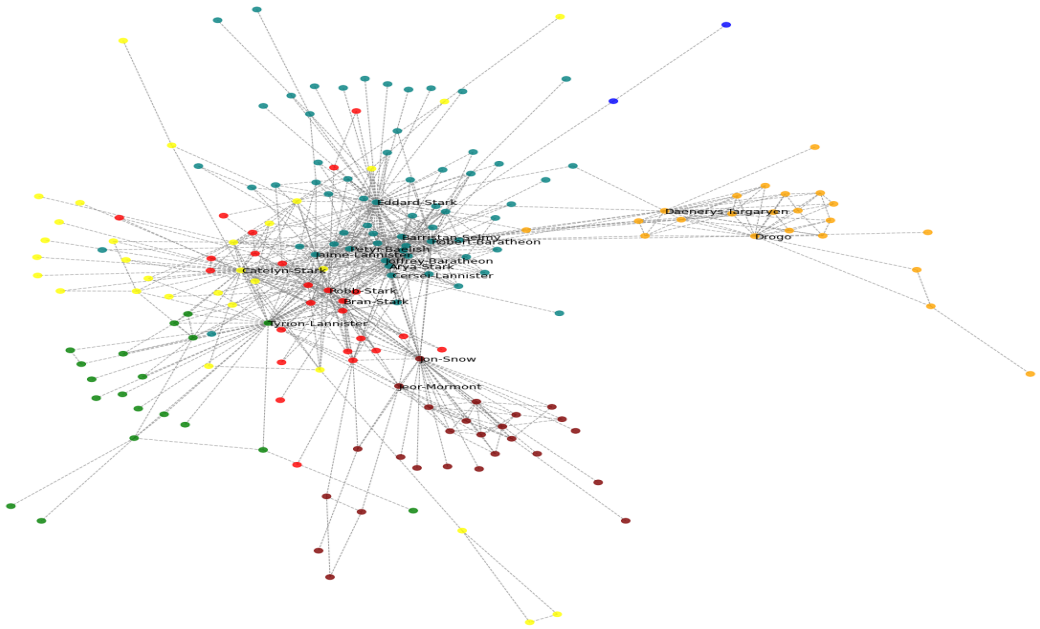

Figure 4: Louvain Method for Book 1

In the first book we have 7 partitions which correspond to the main geographic areas where the characters are set. This partition really correspond to the geographic division of the characters between each other.

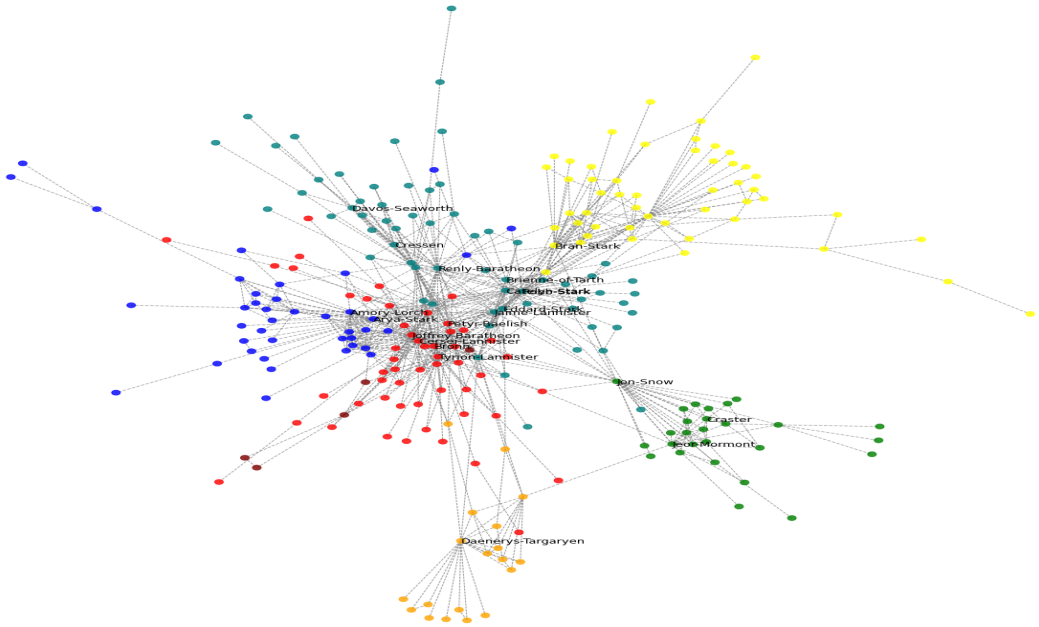

Figure 5: Louvain Method for Book 2

We see the same thing also in the second book but this time the main characters for each partition are different. In fact we now see that Arya Stark is in a different partition and the same thing for Renly Baratheon and Brann Stark. So we see a change in the narration and evolution of the characters.

Label Propagation

Label Propagation Algorithm (LPA) is an iterative algorithm where we assign labels to unlabelled points by propagating labels through the dataset. The Label Propagation cretes 7 partitions for book 1 and 10 partitions for book 2. These partitions are not so clear, in fact they could be a good division but if we look better it's difficult to find the ratio of them due to the creation of mini division made of 1 or 2 nodes.

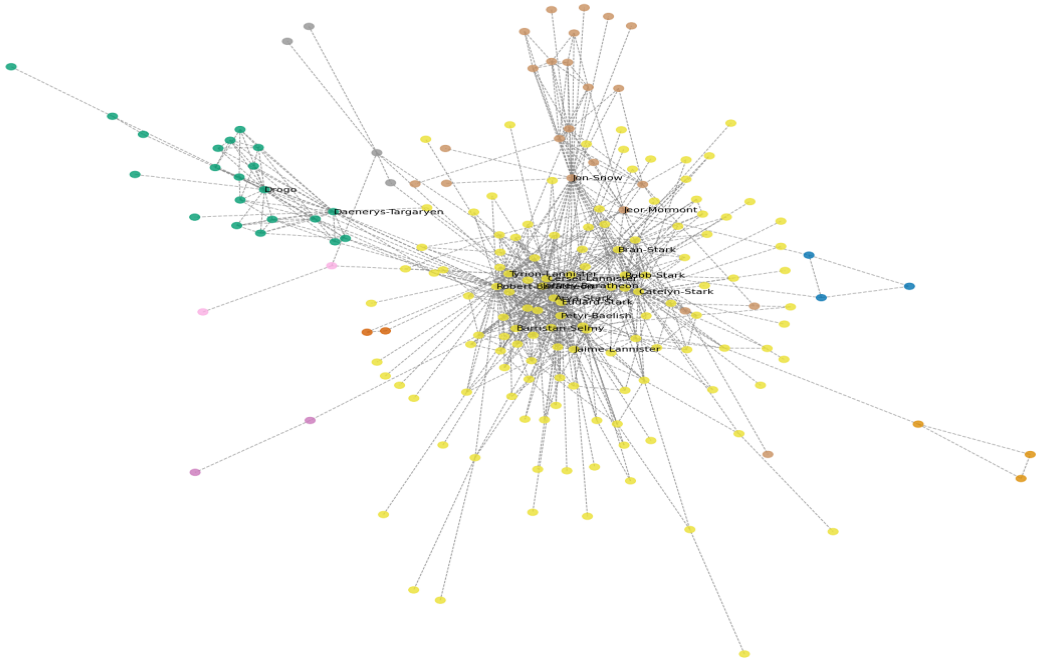

Figure 6: Label Propagation Method for Book 1

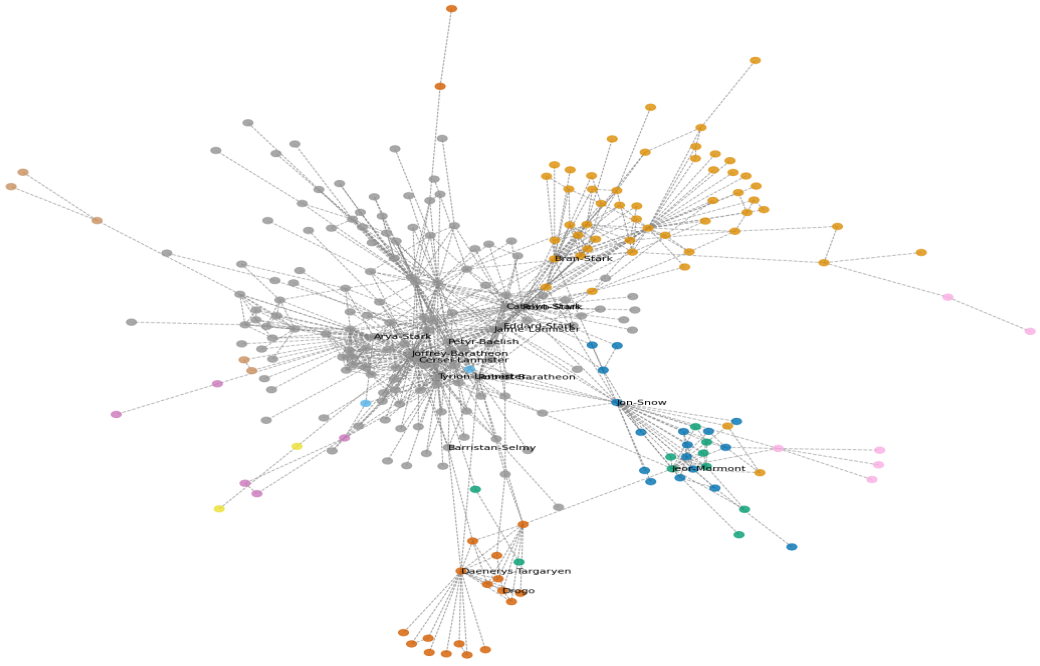

Figure 7: Label Propagation Method for Book 2

Girvan Newman and K-Clique

I also used two other algortihms. The Girvan Newman algorithm which detects communities by progressively removing edges from the original network, focusing on the betweennes. He performs in the right way but only detecting two large communities that are the biggest one that correspond to the characters in Westeros and the one of Daenerys Targarien and Drogo who are in Essos region. We see this division in boht books.

Then I applied the K-Clique algorithm where a clique is a subset of vertices of an indirect graph such that every two distinct vertices in the clique are adjacent. After some tentative I think that 5 cliques were the right parameter to obtain an acceptable result. But also with this algorithm the partition is a bit strange and not so reliable. For this reason I won't show it.

Conclusions

In the end we can conclude that the method of counting as interaction when two characters names are within 15 words is quite reliable even if sometimes death people are mentioned by other characters and this creates an interaction which not exist. We see this more often in the second book.

But as we can see from our network analyses, it's possible to understand the importance of the characters by their interactions and with the use of some metrics like degree centrality and eigenvector centrality that are the most reliable instead of betweenness and closeness.

Also with the communities detection algorithms we were able to understand the evolution of the main characters, in particular we can say that each communities belong to a specific geographic area were the characters act. Maybe it would have been different if the area wasn't so large and the characters was concentrated in the same zone.

The best communities detection algorithms were the Louvain and then the Label Propagation. They are able to catch some hidden relations between characters and put them together. Also in the second book they are able to keep the differences of some characters that change their importance in the narration.

At end we can say that the way we calculate the interactions works, even if I think we could implement it adding the count of column and quotation marks so we can isolate an open speech with the first character that is present for sure and the second can have a more probability to be there with the first one.