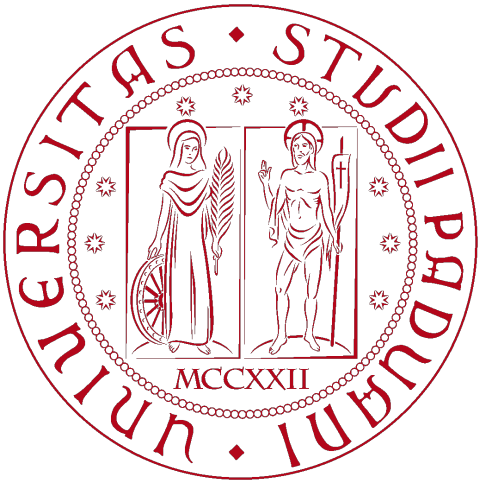

UNIVERSITÀ DEGLI STUDI DI PADOVA

Network analysis of characters and their relationships of *Harry Potter*'s books

Emma Roveroni 2058618
Fabiana Rapicavoli 2045245

September 2023

Indice

1 Introduction

1.1 Analysis of the relationships between the characters in Harry Potter

This report is about the analysis of the relationship between the characters in the Harry Potter saga's books, written by J.K. Rowling. This is a fantasy saga, in which the real world coexists with magic, that has depopulate over the years among kids, teenagers and also adults all over the world.
The characters of the books are several, they can be young students of magic, magic's professors, families, not-magic people, fantastic animals etc. Since there are seven books, some of the characters are present in them form the beginning, while others appear later on and stay till the end, others are present just in one. Of course the story of Harry Potter has people who fight against each other, some are on the good side, while others are the evil ones that need to be defeated.
Thus, for the great mole of characters and for the different type of relationships that there can be between them, it can be interesting to study a network that represents all the characters and their relationships.
Note: in the considered network, not every character and link are present.

1.2 Software

The tools used for this project are the following ones:

- `Python 3.10.12` and different libraries;
- `Gephi`.

2 Methods

2.1 Dataset

The dataset used to represent all the information described in § 1.1 is composed of two separated *csv* files:

- *characters.csv*, which contains information of single nodes, like id, name, biography, house, and side;
- *relations.csv*, which represents the edges between all the nodes, with the corresponding positive or negative weight, depending on whether the two characters are allies or enemies.

The two files are combined using `python`.

To import the graph, self loops were deleted and the attributes about the house of belonging and the side of every character were manually added.
In particular, the possible houses are:

- Gryffindor, whose symbols are bravery, daring, nerve, and chivalry;
- Slytherin, whose symbols are ambition, leadership, self-preservation, cunning and resourcefulness;
- Ravenclaw, whose symbols are intelligence, knowledge, curiosity, creativity and wit;
- Hufflepuff, whose symbols are hard work, dedication, patience and loyalty;
- None, for the non magician (or muggles, in like in the book it's indicated);
- Unkown, for minor character, when any house was specified.

While the possible side are:

- Good;
- Bad;
- Neutral.

2.2 Network properties

2.2.1 Metrics

As the first approach, the network has been explored by learning more insight about it. In order to do that, different metrics have been exploited and they will be briefly introduced in this section.

- **Density** The density of a network is computed as the ratio between the edges present in a graph and the maximum number of edges that the graph can contain. Conceptually, it provides an idea of how dense a graph is in terms of edge connectivity.
- **Small-world network** A small-world network is a type of graph where most of the nodes are not neighbors of one another but the majority of the nodes can be reached from every other node with a small number of steps. In particular, in a small-world network, the distance between two random nodes grows proportionally to the logarithm of the number of nodes present in the network. To check if a network is of the type of small-world, we need first to generate an appropriate ensemble of null-model networks, such as Erdős–Rényi random graphs, or Maslov–Sneppen random graphs, then compute the average of the mean shortest path length L_r and the clustering coefficient C_r over this ensemble of null-model networks. After that, we need to calculate the normalized shortest path $\lambda := L/L_r$ and $\gamma := C/C_r$, where L and C are the average of the mean shortest path length and the clustering coefficient of the original network. Finally, if λ and γ fulfill certain criteria (e.g., $\lambda \approx 1$ and $\gamma > 1$), we can assume that the network is of the type of small-world.
- **Connected graph** A graph is connected if, for all couples of nodes, there exists a path connecting them.
- **Diameter** The diameter of a network is the highest distance in the network. In other words, it is the shortest distance between the two most distant nodes in the network. This gives an estimation of how big is the network.
- **Shortest paths** The shortest path between any couple of nodes is the path with the minimum length.
- **Average shortest path** The average shortest path length is the sum of path lengths between all pairs of nodes u and v, normalized by $n * (n - 1)$ where n is the number of nodes in the network.
- **Degree** The degree k_i of node i in an undirected network is the number of links i has to other nodes, or the number of nodes i is linked to.

- **Average degree distribution** For an undirected graph the average degree distribution $< k >$ is computed as

$$< k >= \sum_i \frac{k_i}{N} = \frac{2L}{N}$$

where L is the number of edges and N the number of nodes.

- **Bridges** A bridge is a link between two connected components, whose removal would make the network disconnected.

2.2.2 Centrality measures

Centrality in a network is essential to understand the influence and the importance of the nodes. For this project, five centralities measures were computed:

- **Betweenness centrality**: The algorithm calculates shortest paths between all pairs of nodes in a graph. Each node receives a score, based on the number of shortest paths that pass through the node. Nodes that more frequently lie on shortest paths between other nodes will have higher betweenness centrality scores. [1] The formula is:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Figura 2.1: Betweenness centrality formula.

where sigma *st* is the total number of shortest paths from node s to node t and sigma *st(v)* is the number of those paths that pass through v (not where it's an end point);

- **Closeness centrality**: It measures its average farness (inverse distance) to all other nodes. Nodes with a high closeness score have the shortest distances to all other nodes. The formula is:

$$C(x) = \frac{N-1}{\sum_y d(y,x)}$$

Figura 2.2: Closeness centrality formula.

where *d(y,x)* is the distance (length of the shortest path) between vertices x and y and N is the total number of nodes;

- **Eigenvector centrality**: the algorithm measures the transitive influence of nodes. Relationships originating from high-scoring nodes contribute more to the score of a node than connections from low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores. The formula [2] is:
  Let $A = (a_{i,j})$ be the adjacency matrix of a graph. The eigenvector centrality x_i of node i is given by:
  $$x_i = \frac{1}{\lambda} \sum_k a_{k,i}\, x_k$$
  where the eigenvalue $\lambda \neq 0$ is a constant. In matrix form we have:
  $$\lambda x = xA$$

- **Harmonic centrality** (also known as valued centrality): it is a variant of closeness centrality, that was invented to solve the problem the original formula had when dealing with unconnected graphs. The formula is:
  $$H(v) = \sum_{u|u \neq v} \frac{1}{d(u,v)}$$
  Figura 2.3: Harmonic centrality formula.

  where *d(u,v)* is the distance between node u and v and $1/d(u,\ v) = 0$ if there is no path from u to v;

- **Degree centrality**: it measures the number of incoming or outgoing (or both) relationships from a node, depending on the orientation of a relationship projection. This graphs is undirected, so there's no distinction between in and out degree.
  Let $A = (a_{i,j})$ be the adjacency matrix of a undirected graph. The degree centrality x_i of node i is given by:
  $$x_i = \sum_k a_{k,i}$$
  or in matrix form (1 is a vector with all components equal to unity):
  $$x = 1A$$

Even though the graph has both positive and negative weights, any of this centrality measure was computed considering the weights.

2.3 Homophily analysis

Homophily [3] (from Ancient Greek *homós* 'same, common' and *philía* 'friendship, love') is a concept in sociology describing the tendency of individuals to associate and bond with similar others.
The categories on which homophily occurs include age, gender, class, and organizational role. Individuals in homophilic relationships share common characteristics (beliefs, values, education, etc.) that make communication and relationship formation easier.

In the graph, homophily was exploited to study the similarity between the nodes. In particular, the following metrics were computed:

- **Global homophily**: this metric was computed, counting for every node the frequency of the neighbors with the same attribute and dividing the sum per the total degree of the node, all these values were insert in an array. Then, we computed the mean of this array. This analysis was computed both for house and the side attribute;
- **Local homophily**: this metric was computed, exploiting the same procedure from the previous metric, but for every of the four houses and the three sides;
- **Assortativity coefficient**: assortativity in a network refers to the tendency of nodes to connect with other 'similar' nodes over 'dissimilar' nodes.
  Here we say that two nodes are 'similar' with respect to a property if they have the same value of that property. Properties can be any structural properties like the degree of a node to other properties like weight or capacity. An assortative network is when high degree nodes connect with each other avoiding low degree nodes.
  The assortativity coefficient was computed both for house and the side attributes;
- **Jaccard similarity**: this metric is a common proximity measurement used to compute the similarity between two objects.
  The Jaccard coefficient measures similarity between finite sample sets, and it is defined as the size of the intersection divided by the size of the union of the sample sets.

2.4 Analysis of the triangles

The triangle count algorithm calculates the number of triangles for each node in the graph. A triangle is a set of three nodes where each node has a relationship to the other two.

It can be used to determine the stability of a graph and it is often used as part of the computation of network indices, such as clustering coefficients or the local clustering coefficient.
About the triangles analysis, the following metrics were computed:

- **The frequency of the triangles** involving every character;
- **The total number of triangles**;
- **Analysis of the relationships**: exploiting the positive and negative weights of the edges, the frequency of the combinations of the signs for each triangle has been calculated. In other words, it was counted the frequency of the four combinations of '+ + +', '+ + -', '+ - -', '- - -'.
  What it's expected to be found is that there are more of '+ + +' relations because trios of friends are very common in a social network and '+ - -' because of the "enemy of my enemy is my friend rule";
- **The average clustering coefficient**, where the clustering coefficient of a single node is the fraction of existing triangles through that nodes, and it follows the next formula:

$$C = \frac{1}{n} \sum_{v \in G} c_v$$

Figura 2.4: Average clustering coefficient formula.

  where c_v is the clustering coefficient of the node v, for every node of the network.

- **Triadic closure**, also known as global clustering coefficient: it computes the fraction of all possible triangles present in the network G. In other words, triadic closure measures the probability that the adjacent vertices of a vertex are connected, so the tendency of edges in a graph to form triangles.
  Possible triangles are identified by the number of "triads" (two edges with a shared vertex), according to the following formula:

$$T = 3 \frac{\#triangles}{\#triads}$$

Figura 2.5: Triadic closure formula.

2.5 Most loved and hated characters

Exploiting the positive and negative weighs, it was possible to detect the most loved and the most hated characters, so the top five nodes with more positive or negative weighs than the others, respectively.

The analysis was made both considering the absolute values and the normalized ones, by the total degree of that character.

The procedure involves counting the frequency of positive and negative edges for each character and extracting the first five nodes with the highest values. Then, the same procedure was replicated, but considering only nodes with a number of edges at least equal to the mode of the degree, and the counts were normalized by the total number of edges of that character.

2.6 Community detection

Granovetter's theory suggests that networks are composed of tightly connected sets of nodes (i.e., communities), loosely connected between them. These techniques of community detection are able to automatically find such densely connected group of nodes.
The methods used in this project are the following:

- **Infomap**: The algorithm is divided into two phases, which are repeated until the map equation cannot be maximized further. In the 1st phase, each object is considered as a separate cluster. For each object p (p = 1, ..., N), its neighbors q (q = 1, ..., N) are checked for whether the map equation increases if p is removed from its cluster and into the cluster of an object q is assigned. The object p is then assigned to the cluster, which maximizes the increase in map equation. However, this only applies in the case of a positive increase. If no positive increase, it can be realized by shifting, the object p remains in its previous cluster. The process described above will be repeated and sequentially performed for all objects until no improvement in map equation can be achieved.
  The 2nd phase is useful to obtain weights for the connections between the clusters. The sum of the weights of the connections between the objects of two corresponding clusters is used. If such, a new network was formed with "metacluster", the steps of the 1st phase will be applied to the new network next, and the map equation will be further optimized. A complete run of both phases is called a pass. Such passes are repeatedly carried out, until there is no more change in the cluster, and a maximum of map equation is achieved. The map equation exploits the duality between finding cluster structures in networks and minimizing the description length of the motion of a so-called random walk. This random walker randomly moves from object to object in the network. The more the connection of an object is weighted, the more likely the random walker will use that connection to reach the next object. The goal is to form clusters in which the random walker stays as long as possible [4];

- **Girvan Newman**: Girvan-Newman will remove the edges with the largest edge betweenness in every iteration and it computes the modularity Q of the communities split. It will repeat these steps, if Q does not satisfy the modularity threshold [5];

- **k-clique percolation**: k-clique is a fully connected sub-graph on k vertices. Two k-cliques are said to be adjacent if they have exactly k-1 common vertices, i.e., if they differ only in a single vertex. A sub graph, which is the union of a sequence of pairwise adjacent k-cliques, is called k-clique chain and two k-cliques are k-clique-connected, if there exists at least one k-clique chain containing these two k-cliques. Finally, k-clique percolation cluster is defined as a maximal k-clique-connected sub graph, i.e. it is the union of all k-cliques that are k-clique-connected to a particular one k-clique [6];

- **Label propagation**: The intuition behind the algorithm is that a single label can quickly become dominant in a densely connected group of nodes, but will have trouble crossing a sparsely connected region. Labels will get trapped inside a densely connected group of nodes and those nodes that end up with the same label when the algorithms finish can be considered part of the same community.
  The algorithm works as follows: Every node is initialized with a unique community label (an identifier). These labels propagate through the network.
  At every iteration of propagation, each node updates its label to the one that the maximum numbers of its neighbours belongs to. Ties are broken arbitrarily but deterministically.
  LPA reaches convergence when each node has the majority label of its neighbours and it stops if either convergence, or the user-defined maximum number of iterations is achieved.

- **Greedy modularity maximization**: This algorithm uses a greedy optimization process in which, starting with each node being the sole member of a community of one, it repeatedly joins together the two communities whose amalgamation produces the largest increase in the modularity. For a network of n nodes, after n - 1 such joins we are left with a single community and the algorithm stops [7].

- **Walktrap**: This approach is based on random walks, especially on the fact that in a graph random walks tend to get "trapped" into densely connected parts corresponding to communities. By using some properties of random walk, it's possible to define a measurement of the structural similarity between vertices and between communities, thus defining a distance [8].

- **Louvain**: The Louvain algorithm works with the same procedure than infomap, but computing the modularity, instead of the map equation. [8].
  This community detection was computed in the `Gephi` software.

2.7 Robustness

Robustness is another important aspect of network analysis. The main aim is to assess the capacity of the network to maintain functionality (or connectivity) after undergoing failures

and perturbations, more specifically, after node removal.
To remove nodes from the network, different strategies were experimented:

- **Random node removal**: nodes are removed randomly, to check the network's ability to survive random failures;
- **Homophily-based node removal**: nodes are removed starting from the ones with the highest homophily score.
- **Centrality-based node removal**: nodes are removed starting from the ones with the highest betweenness centrality score.

After the node removal, the most important metrics were computed, in order to compare the last results with the previous ones, in the original graph.

2.8 Link prediction between the existing nodes in the graph

Link prediction consists in inferring which new interactions among its members are likely to occur in the near future, given a snapshot of a social network, based on measures for analyzing the "proximity" of nodes in a network.
The methods used in this project are:

- **Preferential attachment**: Preferential Attachment is a measure used to compute the closeness of nodes, based on their shared neighbors.
  Preferential attachment means that the more connected a node is, the more likely it is to receive new links.
- **Adamic Adar index**: Adamic Adar is a measure used to compute the closeness of nodes based on their shared neighbors.
- **Jaccard similarity**: The same procedure already explained in the homophily subsection, but used in the node where there are not a relationship yet, instead.

All the method were computed just between the couple of nodes without an edge yet.

2.9 Link prediction between the graph and new nodes

The procedure is very similar to the previous one, with the only difference the link prediction is computed between the original graph and one or more new nodes.
The algorithm used in this project is Barabási–Albert preferential attachment one. A graph

of nodes is grown by attaching new nodes each with edges that are preferentially attached to existing nodes with high degree.

In the graph, it has been added:

- One node with just one edge;
- One node with ten edges;
- Ten nodes with x edges, where x is the average number of the edges per node.

For every addiction, the preferential attachment score of the new link was computed through the preferential attachment algorithm, between the nodes involved in the procedure.

3 Results and discussion

In this chapter, the characteristics of the network will be take into consideration.

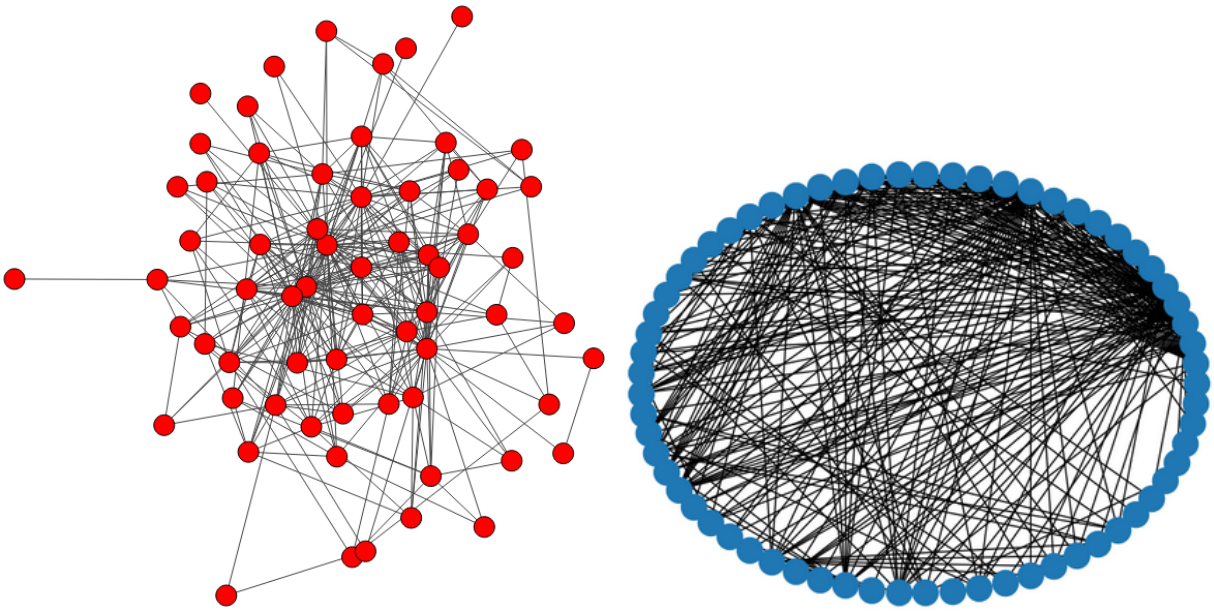

Figura 3.1: Graph illustration.

The graph is composed by a list of 65 characters and the 330 relationships between them, so the related graph has the same number of nodes and edges, respectively, of course.
As it is possible to notice from the picture, the graph is connetcted.

3.1 Exploratory results

Before starting to analyse the graph, an exploratory analysis has been computed, in order to better understand the main characteristics about the node and the links, considering the general entity and frequency about the attributes and the weights.

The frequency of the house and the side attributes in the graph is highlighted in Figure 3.2. In this graph, of course there are not all the characters present in the books, but this statistics resume the same proportion.

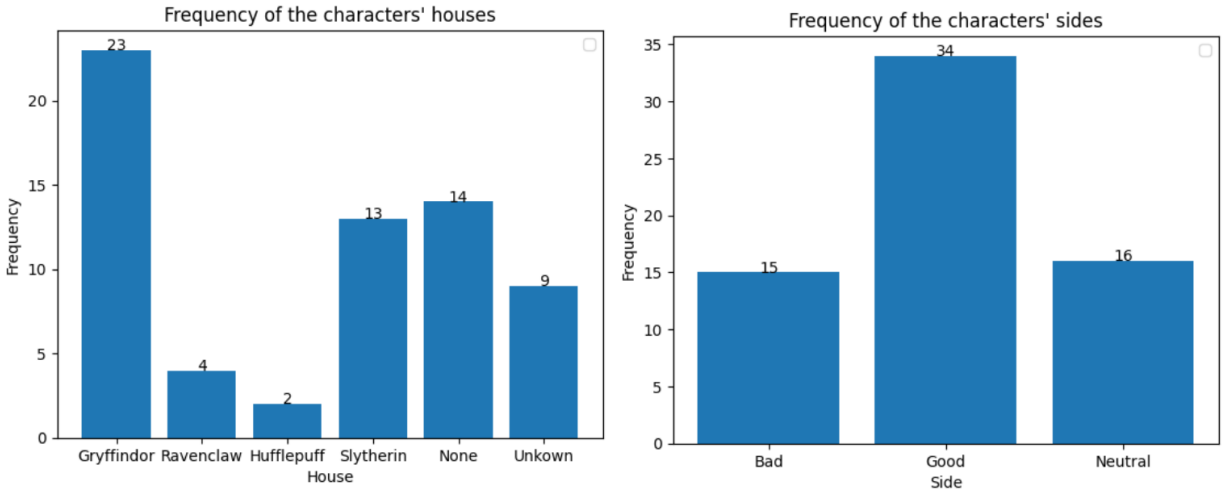

Figura 3.2: The frequency of the house and of the side attributes in the graph.

In Figure 3.3, it's possible to have a better visualization of how the characters are divided by houses (first graph) and the side (second graph).

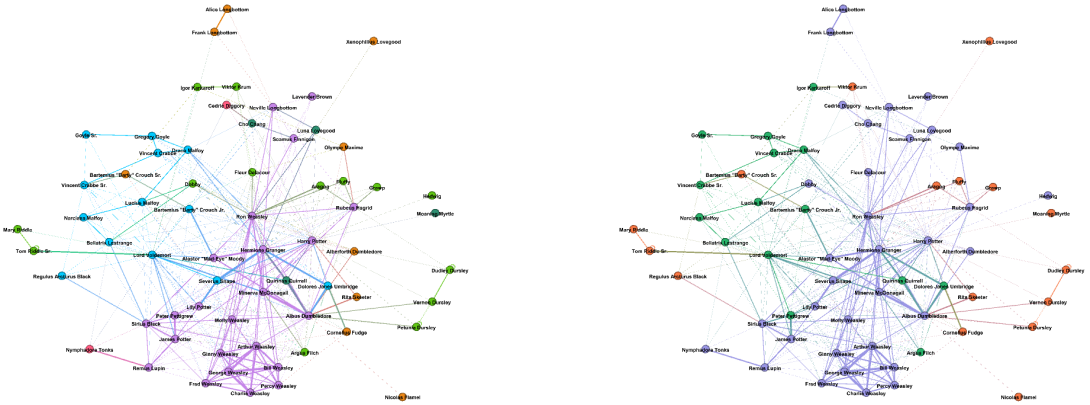

Figura 3.3: Graph illustration divided in the first plot by their houses and in the second one by their side. The graph is made with `Gephi`.

The frequency of the positive and negative weights in the graph is shown in Figure 3.4. It is possible to notice good edges are the double of the negative one.

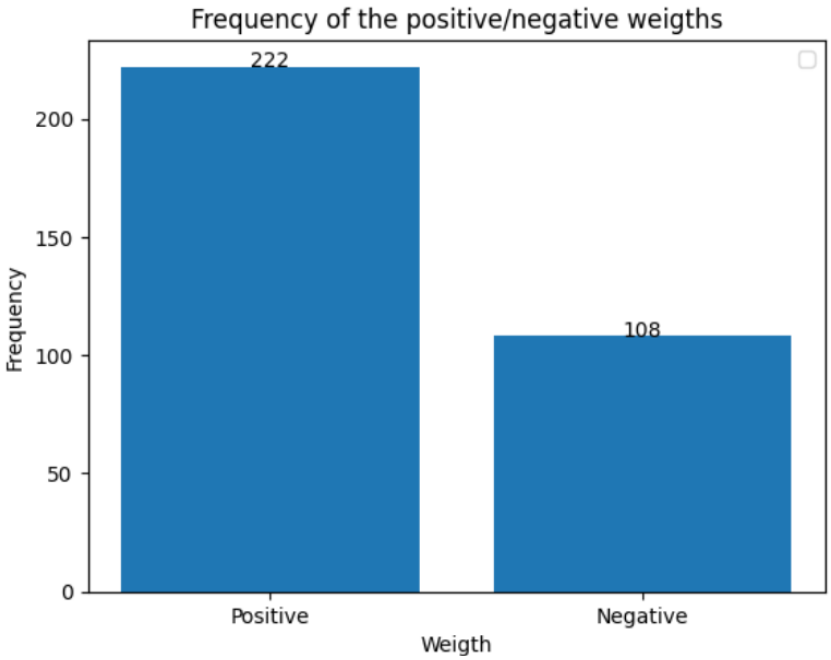

Figura 3.4: Differences between positive and negative weights.

3.2 Network properties

In this section, the major metrics will be analysed, in order to better understand the basic characteristics of the graph.

3.2.1 Metrics

3.2.1.1 Density

The density of the Harry Potter characters' graph is 0.159, since 330 edges are present and the number of the possible edges is 2.080 in total. This value highlights our network is not particularly dense, probably due to the fact some nodes have many edges and some others have just one/two edges, but we'll analyse deeply this phenomena in the degree subsection.
Another thing to take into consideration is in this network there are characters present in different books, for example Quirinus Quirrell is in the first book and Alastor Moody in in the fourth book, so it is impossible the two characters do not meet, or in the network there are also the magic pets, so it is normal these nodes have edges just with the respective owners.

3.2.1.2 Small world network

The Harry Potter's network resulted to be of the family of small world network. This is supported by the values found for $\lambda = 1.03$ and $\gamma = 3.31$, which both agreed with the requirements explained in § 2.2.1.

3.2.1.3 Diameter and shortest path

The diameter in this network is 4. This is a low value. This suggests the nodes are very connected to each other, in particular in some region of the network and it's easy to create connections between the characters.

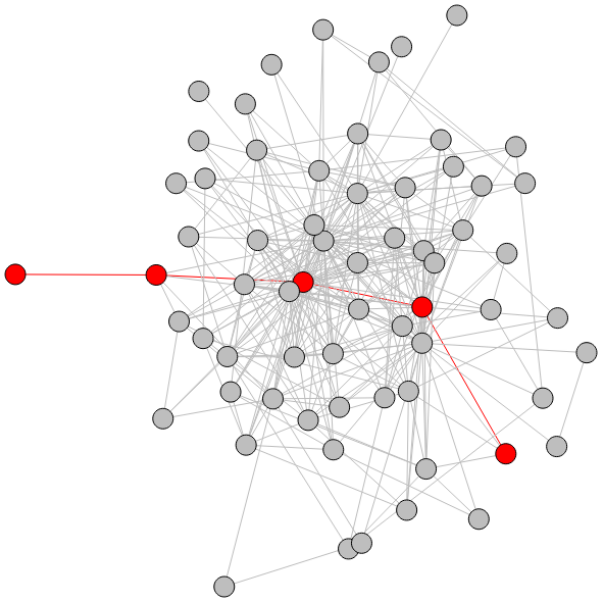

Figura 3.5: Diameter.

The diameter highlighted in the Figure 3.5 (just one of the 730 possible) involves the following edges: [['Regulus Arcturus Black', 'Sirius Black'], ['Sirius Black', 'Hermione Granger'], ['Hermione Granger', 'Luna Lovegood'], ['Luna Lovegood', 'Xenophilius Lovegood']], where Regulus Arcturus Black and Sirius Black are brothers, Sirius Black and Hermione Granger are allies, Hermione Granger and Luna Lovegood are friends and Luna Lovegood and Xenophilius Lovegood are daughter and father.

To justify the low diameter there is also the average shortest path, which is 2.028 and it's a good result because this is the half of the diameter and because this means that on average a node can connect with whether other node through a link from one of its neighbors. Thus, this is the confirmation that the graph is well connected, even though the density is low.

3.2.1.4 Degree

The average degree in this network is 10.154, while the minimum degree is 1 (belonging to the ancient Nicolas Flamel, the owl Hedwig, the ghost Moaning Myrtle and one the secondary character's father Xenophilius Lovegood) and the maximum is 48 (belonging to the main character Harry Potter).
The mean is index that there are more characters which have a low degree. In fact, the mode of the degree, so the most frequent degree, is 4 with 13 nodes. In Figure 3.6, it is also possible to see the degree distribution.
Considering this last observation, the causes that lowers the diameter and the average shortest path is the fact many nodes are connected with Harry Potter or with one of his neighbor. In fact, Harry Potter has the 73.85% of the nodes linked with him.

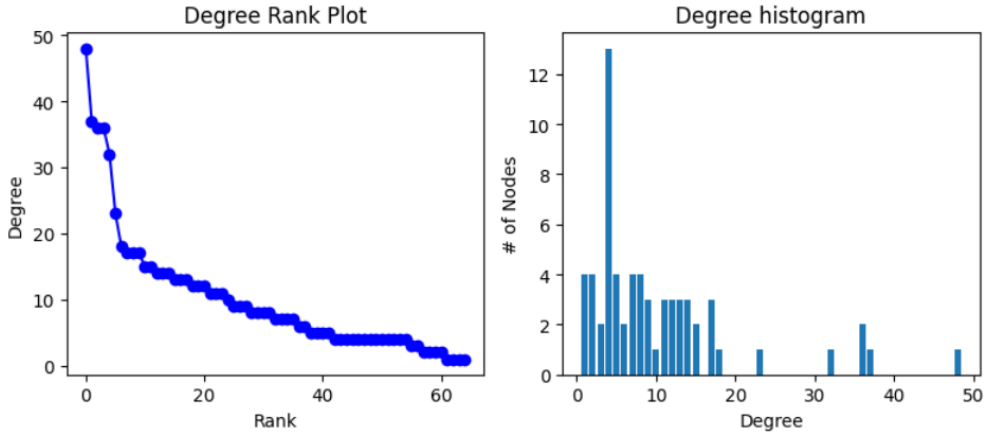

Figura 3.6: Degree distribution.

The five characters with the highest degree are In Table 3.1.

Character	Degree
Harry Potter	48
Ron Weasley	37
Lord Voldemort	36
Hermione Granger	36
Albus Dumbledore	32

Tabella 3.1: First five nodes with the highest degree.

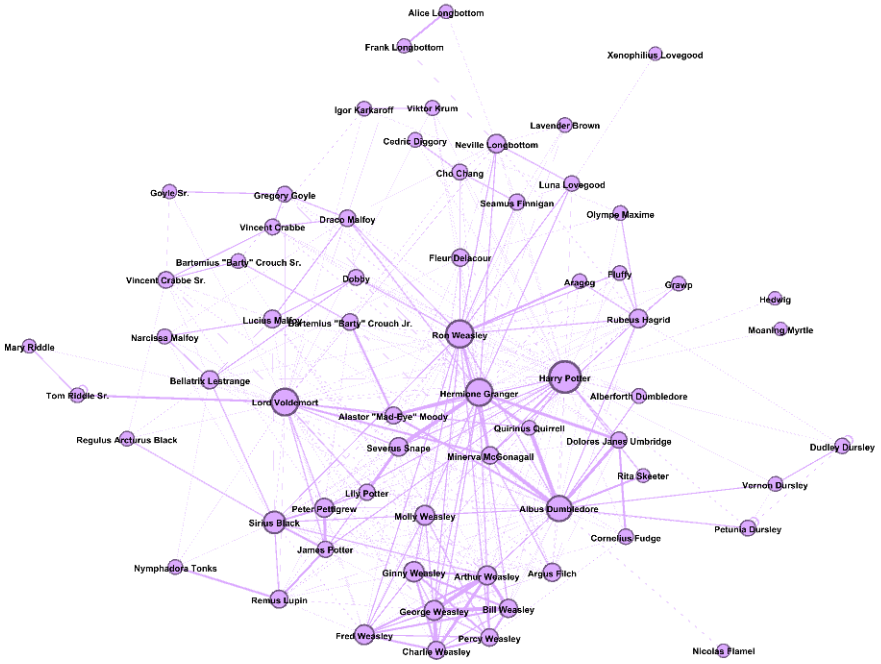

Figura 3.7: Network with nodes proportionally as large as their degree.

This last graph, Figure 3.7, finally confirms the previously mentioned intuition: there are more nodes with few edges, than the contrary.

3.2.1.5 Bridges

In this network, there are four bridges, and, of course, they involved the four nodes with the lowest degree, as already mentioned before.
The bridges, which can be seen in Figure 3.8, are between Albus Dumbledore and his old friend Nicolas Flamel, Harry Potter and his owl Hedwig, Harry Potter and his friend Moaning Myrtle, and between Luna Lovegood and his father Xenophilius Lovegood.

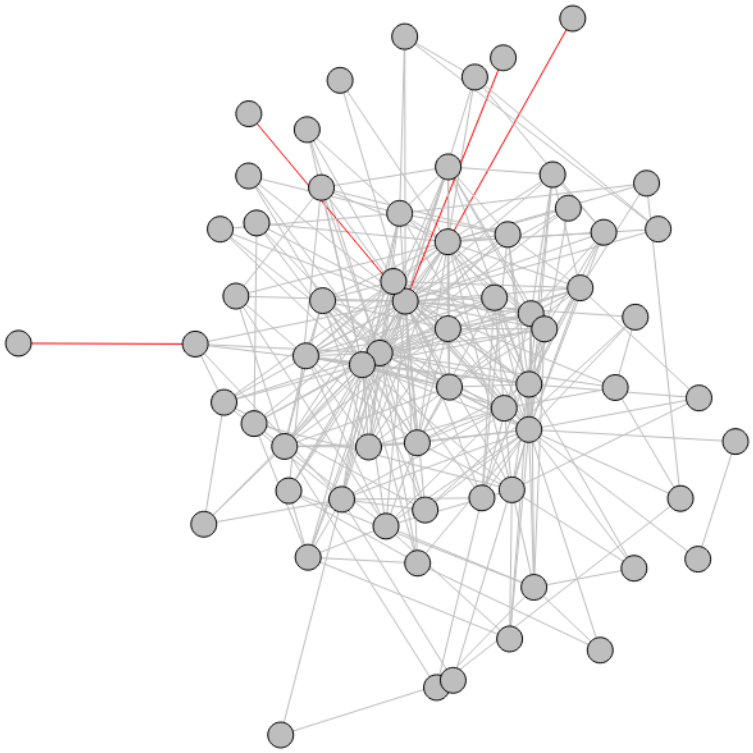

Figura 3.8: Bridges present in the network. They are highlighted in red.

3.2.2 Centrality measures

The following tables report the results obtained by the different centrality measures. In Table 3.2 it's possible to see the first five nodes/characters of the network with the highest centrality scores, for each type of centrality. As it is possible to notice, the first five nodes with the highest centrality measures are always the same. In particular, the top one is the node of Harry Potter for every type of centrality. This clearly makes sense, since Harry Potter is the main character of the books and he is the one that brings together all the characters. The other personages still are the main ones: Hermione Granger and Ron Weasley are Harry Potter's best friends, thus they have similar relationships to Harry Potter's ones. On the other hand, Lord Voldemort is the main antagonist in the books so again having him in the nodes with the highest centrality scores is coherent, and Albus Dumbledore, which is the principal of the magic school of Hogwarts and he helps the trio to beat the enemy.
The five most central nodes are even the node with the highest degree.
In the second table number 3.3, it has been reported the centrality means.

Measure	**Characters with highest centrality**	**Score of highest centrality**
Betweenness	Harry Potter	0.281940
	Lord Voldemort	0.203448
	Albus Dumbledore	0.100495
	Ron Weasley	0.098503
	Hermione Granger	0.093993
Closeness	Harry Potter	0.800000
	Ron Weasley	0.703297
	Hermione Granger	0.695652
	Lord Voldemort	0.688172
	Albus Dumbledore	0.653061
Eigenvector	Harry Potter	0.320910
	Ron Weasley	0.292882
	Hermione Granger	0.277872
	Albus Dumbledore	0.259822
	Lord Voldemort	0.258514
Harmonic	Harry Potter	0.017857
	Ron Weasley	0.019802
	Hermione Granger	0.020000
	Lord Voldemort	0.020067
	Albus Dumbledore	0.020979
Degree	Harry Potter	0.750000
	Ron Weasley	0.578125
	Lord Voldemort	0.562500
	Hermione Granger	0.562500
	Albus Dumbledore	0.500000

Tabella 3.2: The table sums up the top five nodes with the highest centrality

Centrality	**Mean**
Betweenness centrality	0.016
Closeness centrality	0.505
Eigenvector centrality	0.098
Harmonic centrality	0.028
Degree centrality	0.159

Tabella 3.3: Centrality means.

From the two tables, it is possible to notice the scores about the top five characters of the closeness, eigenvector and degree centrality are highly bigger than the respective means. On the contrary, the scores for the top five characters of the harmonic and betweenness centrality are very near to the mean.

To have a look at a better visualization of the ranking based on the results given by the centrality measures, we can use Figure 3.9.

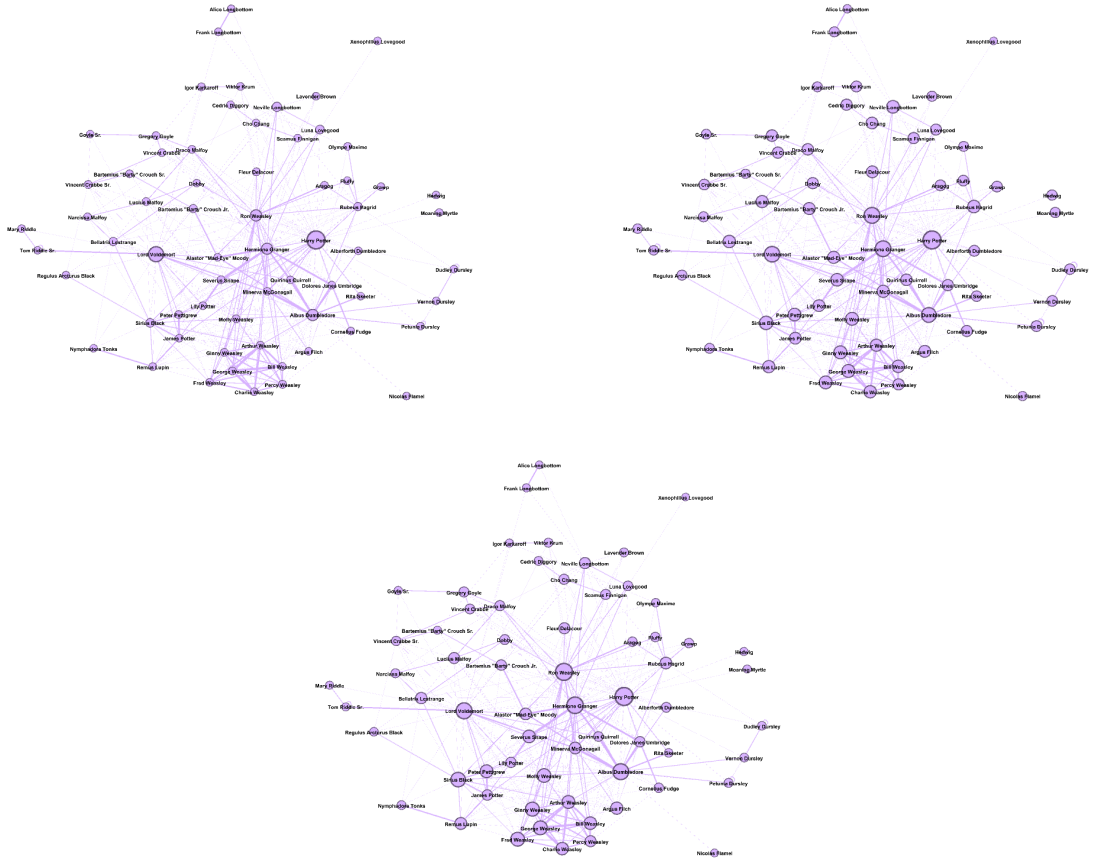

Figura 3.9: Graph illustration of ranking based on centrality scores. First plot by betweeness centrality; second plot by closeness centrality; third plot by eigenvector centrality. The graph is made with `Gephi`.

3.3 Homophily

The homophily of the network was explored based on node attributes: house and side. What we are expecting is that similar nodes may be more likely to attach to each other than dissimilar ones. In this specific case, we forecast to see that characters who belong to the same house or are on the same side, are more likely to interact with each other.

3.3.1 Global Homophily

With regard to global homophily the mean of the homophily level for the house attribute is equal to 0.412. This result is quite low, but it can be understandable since in the book students of different houses interact with each other. What's more, a lot of connections in the network are between heroes and antagonists, which typically belong to different houses: characters on the good side usually belong to the Gryffindor house, while characters on the evil side usually belong to the Slytherin house.
Instead, the mean of the homophily level for the side attribute is equal to 0.539, which again is not that high, but, for the same reasons as before, it can be justified by the fact that there are a lot of interactions between good and evil characters.

3.3.2 Local homophily

In Table 3.4 it's possible to observe the results obtained by the computation of homophily score related to each one of the four houses and of good, bad, and neutral sides.

House/Side	**Homophily score**
Gryffindor	0.707
Slytherin	0.502
Ravenclaw	0.0
Hufflepuff	0.0
Good Side	0.77
Bad Side	0.43
Neutral	0.127

Tabella 3.4: Local homophily scores.

It is possible to see from the table, homophily scores of Gryffindor and good side are high, meaning that nodes of Gryffindor students and of characters that are on the good side tend to interact more. The extremely low scores of the Ravenclaw and Hufflepuff homophily could be a reason why the house global homophily of the house attribute is so low: as a matter of fact, there are just two Hufflepuff and four Ravenclaw students in the network.

For the same reason, it is possible to notice neutral local homophily decreases the side global homophily too.

3.3.3 Assortativity

The assortativity coefficient for the house attribute corresponds to 0.127, for the side attribute is equal to 0.145, while the total assortativity is -0.29, regardless the specific attributes.
Even though these values are low, they make sense. For example in the case of the house attribute, students interact in the story despite their different houses: they attend the same classes, do sports, and still live in the same schools in separate dormitories, so it's normal, like in real schools, that they make friends with each other. What's more, the network it's not just a simple social network: here the links represent not only friendships but mostly alliances and enemies since there is a war between good and bad.
For these reason, there is not a linear correlation between the attributes.

3.3.4 Jaccard similarity

In Table 3.5 the first five Jaccard similarity scores between every couple of nodes are reported. These results are coherent with the story of the characters in the books. In fact, it is possible to observe that the most similar nodes belong all to the Weasley family, more specifically they are all brothers and sisters. For example, the pair of nodes with the highest Jaccard similarity is the one of Fred and George Weasley, which are the Weasley twins and they have the same friends, same enemies, they belong to the same house (Gryffindor) and they are on the same side (the good one). So this explains why they are detected as the most similar elements in the network.
The only exception in this list is Hermione Granger, who does not belong to the Weasley family. Her character and the one of Ron Weasley are in the top five of the most similar nodes. In the books, these two characters are best friends, together with Harry Potter, and later on in the story they are also romantically involved. So, as for the Weasley twins, they hang out with the same people and belong to the same house (Gryffindor) and the same side (Good).

First Node	**Second node**	**Jaccard similarity score**
Fred Weasley	George Weasley	0.88
Bill Weasley	Charlie Weasley	0.78
Hermione Granger	Ron Weasley	0.78
Fred Weasley	Ginny Weasley	0.75
George Weasley	Ginny Weasley	0.75

Tabella 3.5: First top five highest Jaccard similarity scores.

3.4 Analysis of the triangles

The total number of unique triangles in this network is 820 and the first five characters which are involved in the major number of triangles are exposed in Table 3.6.

Character	Number of triangles, which they are involved
Harry Potter	227
Ron Weasley	199
Hermione Granger	179
Albus Dumbledore	162
Lord Voldemort	156

Tabella 3.6: Top five characters involved in the most number of triangles.

The sum of the five values is 923, which is higher the total number of triangles, mentioned before, just because the values in the table admit repetitions, so, for example, the triangle between the golden trio (Harry Potter and his two dearest friends Ron Weasley and Hermione Granger) is repeated in all nodes in the table, but the 830 number of triangles counts just the unique ones.
Considering this last fact, we can divide 923 by 3, which is 307,67 and we can notice that the proportion of the number of single triangles in the first central nodes is about the 36,99% of the total.

Furthermore, exploiting the different positive and negative signs of the edges, the entity of the relations has been studied. It has been counted the frequency of the four combinations of '+ + +', '+ + -', '+ - -', '- - -'. The resulting values in this network are in Table 3.7.

Relation type	Count
+ + +	445
+ + -	49
+ - -	314
- - -	12

Tabella 3.7: Frequencies of the different types.

The results are positive because, in a social network, it could be expected to find more of '+ + +' relation because trios of friends are very common in a social network and '+ - -' because the enemy of my enemy is my friend rule, while the relations '- - -' and '+ + -' are less common because it is unlikely that in a social network there are many trios of enemy between each other and if A is friend with B and B is friend with C, it also unlikely that A and C are enemy.

Finally, the clustering coefficient and the triadic closure have been computed and the respectively values are 0.612 and 0.413.
The formation of triangles is typically measured by the clustering coefficient, in which the focal node is the centre-node in an open triad. In contrast, the recently proposed closure coefficient measures triangle formation from an end-node perspective and has been proven to be a useful feature in network analysis [9].
Considering both the values conjunctively, these results suggest the network have many triangles and the node are very connected from each other, even though every vertex isn't connected with all the possibilities it could exploit.

3.5 Most loved and hated characters

Exploiting the positive and the negative weights, it was possible to identify the most loved and the most hated characters, so the top five nodes with more positive or negative weights than the others, respectively.

The analysis was computed both considering the absolute values and the normalized ones, by the total degree of that character.

3.5.1 Most loved and hated characters - absolute values

In the tables 3.8 and 3.9, it is possible to notice the top five most loved and hated characters, considering the absolute values.

Character	**Number of positive edges**	**Total degree**	**Percentage degree**
Harry Potter	30	48	62%
Ron Weasley	27	37	72%
Albus Dumbledore	24	32	75%
Hermione Granger	24	36	66%
Ginny Weasley	16	18	88%

Tabella 3.8: Most loved characters.

Character	Number of negative edges	Total degree	Percentage degree
Lord Voldemort	28	36	77%
Harry Potter	18	48	37%
Peter Pettigrew	12	15	80%
Hermione Granger	12	36	33%
Ron Weasley	10	37	27%

Tabella 3.9: Most hated characters.

It is very curious to notice the most central nodes are present between the most loved and the most hated character, which was predictable, but is even more curious notice the golden trio, the main characters in the adventures Harry Potter, Hermione Granger and Ron Weasly, are the most loved and hated characters at the same time. This is probably due to the fact they are the main combatants in the war, so it is obvious they have many positive edges, as well as many negative ones, but they always have more positive links.
Just to give an overview an explain who the other characters in the tables are: Ginny Weasley is one of Ron Weasley's sister and Peter Pettigrew is one of the allies of Lord Voldermort and who betrayed Harry Potter's parents.

3.5.2 Most loved and hated characters - normalized values

In the tables 3.10 and 3.11, it is possible to notice the top five most loved and most hated characters with the number of positive and negative edges, normalized by the number of edges per character.

Character	Number of positive edges	Total degree	Percentage degree
Percy Weasley	11	11	100%
Luna Lovegood	6	6	100%
Viktor Krum	5	5	100%
Alberforth Dumbledore	5	5	100%
Grawp	4	4	100%

Tabella 3.10: Most loved characters with normalized values

Character	Number of negative edges	Total degree	Percentage degree
Peter Pettigrew	12	15	80%
Lord Voldemort	28	36	77%
Quirinus Quirrell	3	4	75%
Regulus Arcturus Black	3	4	75%
Fluffy	3	4	75%

Tabella 3.11: Most hated characters with normalized values.

It is curious to see that in hated characters table, there still is Lord Voldemort and Peter Pettigrew, just in a different order, while, in the most loved characters table, all the characters changed, it is objective that two characters are hated, but the old loved characters with the absolute values were loved also because their many edges.
Just to give an overview an explain who the other characters in the tables are: Percy Weasley is one of Ron Weasley's brother and the chief of the Gryffindor house in the first books, Luna Lovegood is a friend and allies of the golden trio, Viktor Krum participated in the three wizards tournament with Harry Potter in the fourth book, Alberfoth Dumbledore is Albus Dumbledore's brother (which is the school's principal), Grawp is the giant brother of Rubeus Hagrid (which is the school's guardian), Quirinus Quirell and Regulus Arctutus Black are minor Lord Voldemort's allies and Fluffy is the Rubeus Hadrid's three heads dog.

3.6 Community detection

For each algorithm applied for community detection we are going to show the results obtained.

3.6.1 Infomap

The community detected from the *Infomap* algorithm are two, and the way the graph is divided can be seen in Figure 3.10.

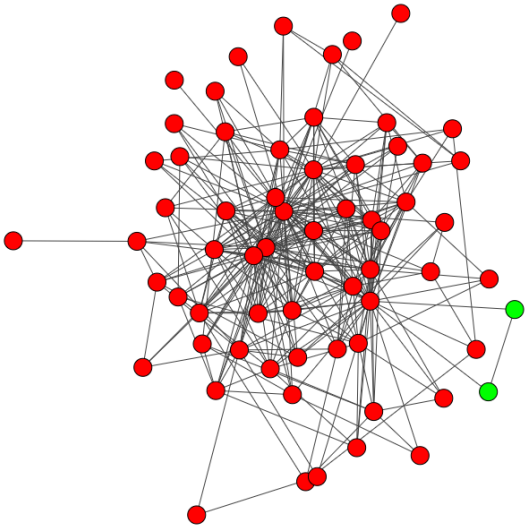

Figura 3.10: Infomap community detection.

The small community that the algorithm detects is composed of just two nodes (the green ones in Figure 3.10) is the one of Tom Riddle Sr. and Mary Riddle, which are husband and wife and parents of Lord Voldemort. It makes sense that they are separated communities since in the story told in the Harry Potter saga they are already dead and do not have any interactions with living present characters, except for their son. The modularity score for the *Infomap* method is 0.006, which is very low.

3.6.2 Girvan Newman

In Figure 3.11 it is possible to observe the community detected by the *Girvan Newman* algorithm.

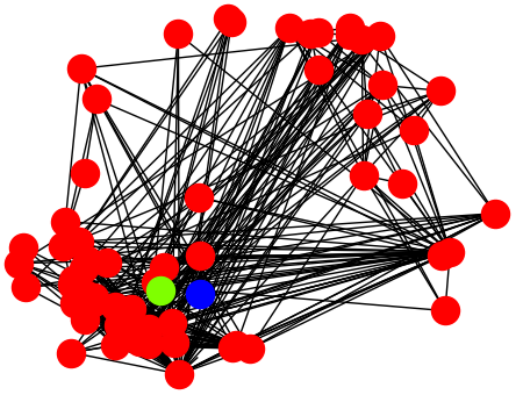

Figura 3.11: Girvan Newman community detection.

In this case, the community identified by the algorithm are three, but two of these communities include just one node each: one is the character of Nicholas Flamel and the other one is Xenophilius Lovegood. This community detection algorithm doesn't seem to fit the network due to these results.
This is also supported by the modularity score for the *Girvan Newman* method, which is 0.006, again a very low result.

3.6.3 k-clique percolation

When defining the *k-clique percolation* algorithm, we have also to define a number (k) as the size of the smallest clique. For this project, it has been chosen $k = 6$. The communities identified with this algorithm are two, more populated than the previous two algorithms, and they can be seen in Figure 3.12, in which green and red nodes represent two different communities, while the grey ones are nodes that are not classified.

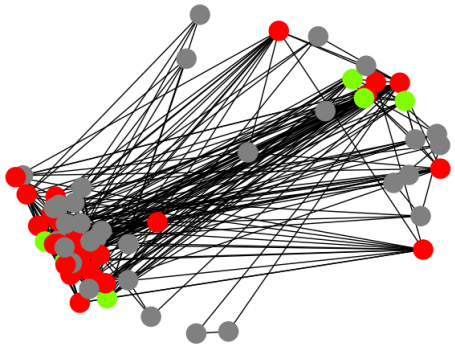

Figura 3.12: k-clique percolation community detection.

Since *k-clique percolation* is an algortihm that allows overlaps in communities, all the characters in the "green community" are also present in the red one. The distinction still makes sense, since in the green community the three protagonist (Harry Potter, Hermione Granger and Ron Weasley) are present together with three of the antagonists, which are Slytherin students (Draco Malfoy, Gregory Goyle and Vincent Crabbe).
The modularity for the clique percolation method is 0.039, still quite low.

3.6.4 Label propagation

The results, given by *Label propagation*, algorithm are very similar to the one obtained with *Infomap* algorithm. In Figure 3.13, it is possible to see that the communities identified are just two: one (the red nodes) is the biggest one that includes almost all the nodes of the network, and one (the green nodes) includes just two nodes.

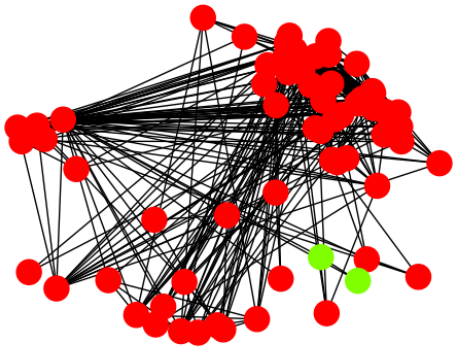

Figura 3.13: Label propagation communities detection.

As for *Infomap*, the two nodes in the small community are the one of Mary Riddle and Tom Riddle Sr., the parents of Lord Voldemort.
The modularity for the *Propagation clustering* method is 0.006, so it can observed that even the modularity is the same of the *Infomap* algorithm.

3.6.5 Greedy modularity maximization

Using *Greedy modularity maximization* algorithm has given better results with respect of the previous algorithms. In fact, as it's possible to see in Figure 3.14, the algorithm identifies four communities. What's interesting in these communities division is that one of the community includes the majority of characters that are evil in the books, another one, instead, includes all characters that are mostly (or only) presents in the fourth book of the saga. The other two communities are quite similar, meaning that they both include people on the good side: the bigger one contains the characters that are adults, like professors, and the smaller one, contains the students on the good side.

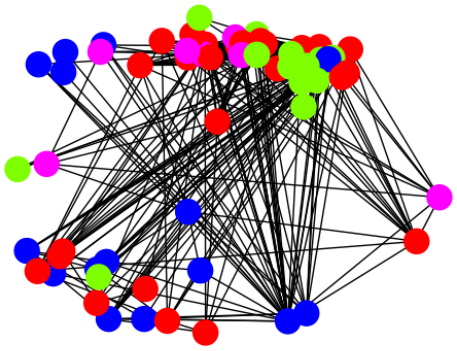

Figura 3.14: Greedy modularity maximization community detection.

With this algorithm the modularity increases a lot with respect of the previous ones. In fact, the modularity for the *Greedy modularity communities* is 0.233.

3.6.6 Walktrap

The number of communities detected by the *Walktrap* algorithm is higher with respect to the other algorithms. Here we have thirteen communities, but most of them are made of very few nodes, in some cases even just one node, as it's possible to see in Figure 3.15.

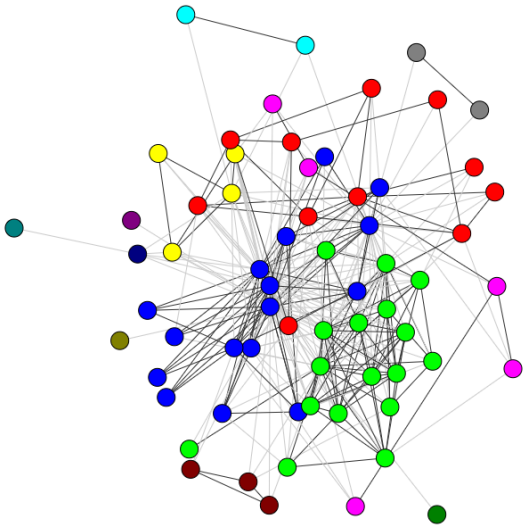

Figura 3.15: Walktrap community detection.

Differently from before, with this type of division, it's quite difficult to find a justification based on the books' story. Despite that, the modularity score is anyway higher than some of the previous algorithm, in fact it is equal to 0.182.

3.6.7 Louvain

Louvain algorithm identified five communities, as it's possible to observe in Figure 3.16. This algorithm was performed using directly `Gephi`.

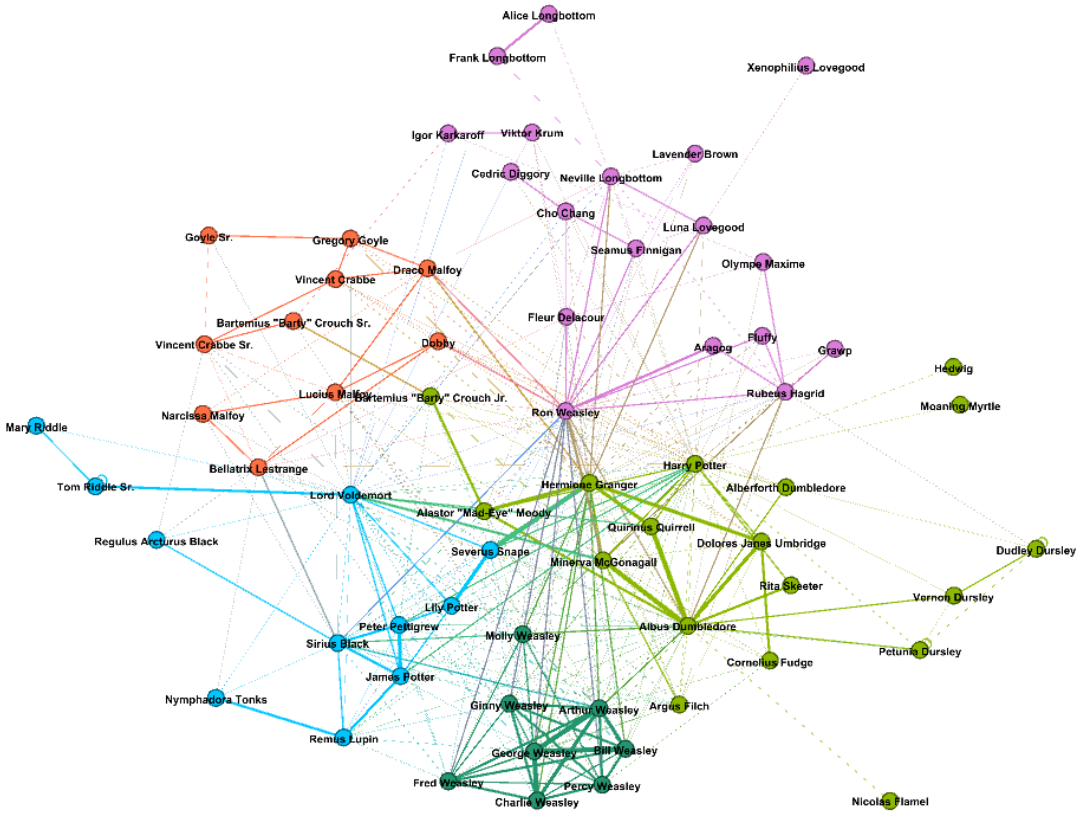

Figura 3.16: Louvain community detection graph subdivision, made with `Gephi`.

With this algorithm, the obtained communities, under some aspects, make sense, by considering the story of the books. In fact, for example, one of these communities (dark green one) is composed of the members of the Weasley family, while another one (light purple one) is composed mostly of Hogwarts's students that are friends with the three main characters. Even the community of orange nodes is coherent with the books, since it gathers the three families, both parents and children (Malfoy, Crabbe and Goyle) that are in contrast with the good characters and that are allied together along Long Voldemort's side.
Here the modularity score is the highest one among all the community detection methods since it's equal to 0.342.

3.6.8 Summary of community detection results

To sum up, the results obtained by the different community detection algorithms can be observed better in the following Figure 3.17.

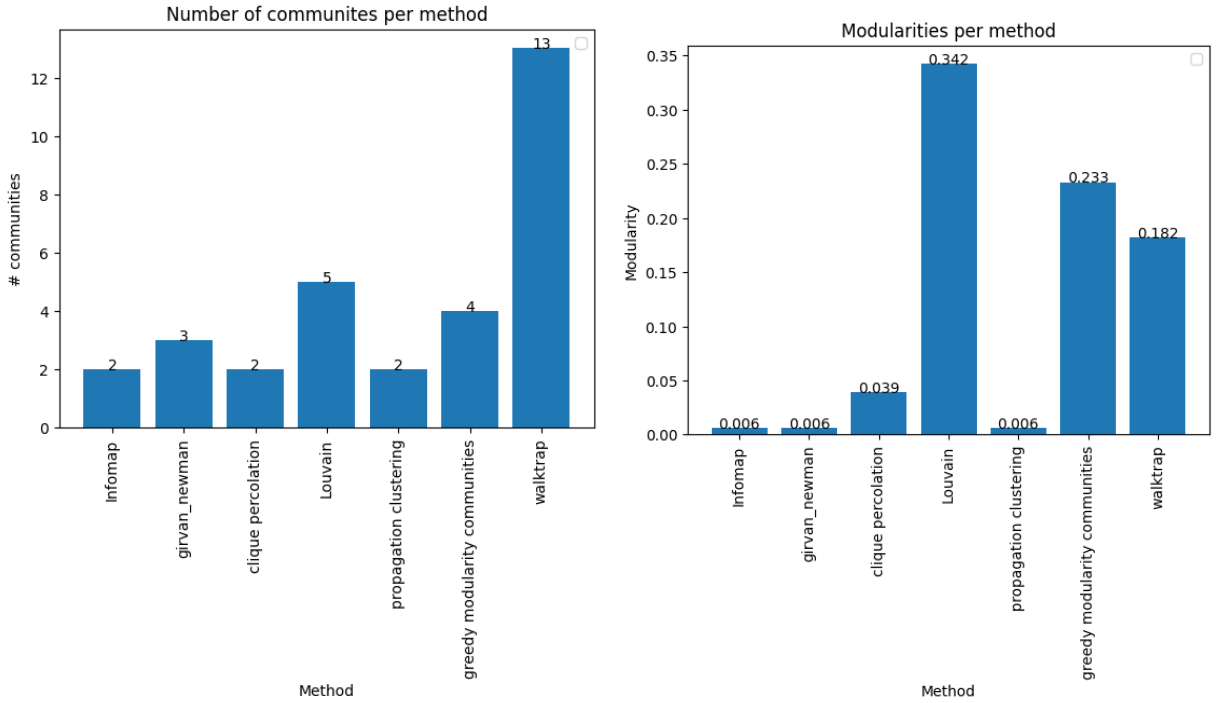

Figura 3.17: In the first histogram there are the number of detected communities for each algorithm; the second histogram represents the modularity score for each algorithm.

In Figure 3.17, it is possible to notice that the majority of the algorithms identify very few communities, like two or three. These algorithms are also the ones that have the lowest modularity, with scores that are really close to 0. These algorithms are *Infomap*, *Girvan Newman*, *k-clique percolation* and *Label propagation*.
The other three algorithms have higher modularity scores, as we can see in Figure 3.17 and detect more communities. The one with the highest number of communities is the *Walktrap*, while the one with the highest modularity score is *Louvain*. As mentioned in the previous discussion, having more communities does not mean that the algorithm is the best one, since, as in the case of *Walktrap*, most of these communities are very little, also made by just one node, and are not coherent with the plot of the books. Thus, for this network, the best community detection algorithms are *Louvain* and *Greedy modularity communities*, since they have a quite good modularity score and the communities they identify, are consistent with what happens in the books of Harry Potter's saga.

3.7 Robustness

In the following section, robustness will be analyzed in the three methods already explained: Random node removal, homophily-based node removal, and centrality-based node removal.

3.7.1 Random node removal

After removing the nodes in a random way, the basic metrics have been computed and the differences between the methods can be observed in Table 3.12 and 3.13.

Removed nodes	Components	Diameter	Average shortest path
5	2	5	2.12
10	1	4	2.070
15	4	4	2.258
20	1	4	2.071
25	4	5	2.189
30	1	4	1.877
35	4	3	1.788
Original	1	4	2.028

Tabella 3.12: Network metrics computed after random nodes removal.

Removed nodes	Mean betweenness centrality	Mean closeness centrality	Mean eigenvector centrality	Mean harmonic centrality	Mean degree centrality
5	0.456	0.018	0.099	0.018	0.099
10	0.495	0.020	0.105	0.020	0.105
15	0.403	0.023	0.105	0.023	0.105
20	0.495	0.025	0.116	0.025	0.116
25	0.403	0.027	0.120	0.027	0.120
30	0.547	0.027	0.142	0.027	0.142
35	0.433	0.021	0.145	0.021	0.145
Original	0.504	0.016	0.098	0.028	0.159

Tabella 3.13: Network centrality computed after random nodes removal.

As it can be noticed from the tables, the metrics change, of course, but slightly, so it is possible to assume that the network is robust by considering the giant component, but this method is probably not the best and most robust one because in three cases out of seven, when 15, 25 and 35 nodes are removed, there are four components and there are always three isolated nodes, which is not the best scenario.

3.7.2 Homophily-based node removal

After removing the nodes based on the homophily, so after removing the 10% and the 30% of Gryffindor and Slytherin characters, the basic metrics have been computed and the differences

between the methods can be seen in Table 3.14 and 3.15.

House	Removed nodes	Components	Diameter	Average shortest path
Slytherin	10%	1	4	2.010
Slytherin	30%	1	4	1.99
Gryffindor	10%	1	4	2.031
Gryffindor	30%	1	4	2.067
Original		1	4	2.028

Tabella 3.14: Network metrics computed after homophily-based nodes removal.

House	Removed nodes	Mean betwennes centrality	Mean closeness centrality	Mean eigenvector centrality	Mean harmonic centrality	Mean degree centrality
Slytherin	10%	0.509	0.017	0.100	0.017	0.100
Slytherin	30%	0.514	0.017	0.103	0.017	0.102
Gryffindor	10%	0.504	0.017	0.101	0.017	0.101
Gryffindor	30%	0.495	0.019	0.106	0.019	0.106
Original		0.504	0.016	0.098	0.028	0.159

Tabella 3.15: Network centrality computed after homophily-based nodes removal.

As it can be noticed from the tables, the metrics slightly change and the giant component doesn't divide, so it is possible to affirm that the network is robust.

3.7.3 Centrality-based node removal

After removing the nodes, based on the most central nodes, according to the betweenness centrality, the basic metrics have been computed and the differences between the methods are summed up in Table 3.16 and 3.17.

Removed nodes	Components	Diameter	Average shortest path
1	3	4	2.112
3	6	5	2.307
5	6	5	2.307
Original	1	4	2.028

Tabella 3.16: Network metrics computed after centrality-based nodes removal.

Removed nodes	Mean betwennes centrality	Mean closeness centrality	Mean eigenvector centrality	Mean harmonic centrality	Mean degree centrality
1	0.455	0.017	0.095	0.017	0.095
3	0.365	0.018	0.093	0.018	0.093
5	0.365	0.017	0.093	0.018	0.093
Original	0.504	0.016	0.098	0.028	0.159

Tabella 3.17: Network centrality computed after centrality-based nodes removal.

As it can be noticed from the tables, the metrics remain almost the same in the giant component, but the network always breaks from three to six components because the most central nodes are linked also with node with just that link, so, when the node is removed, all that nodes became isolated.
This method is not robust.

3.8 Link prediction between the existing nodes in the graph

3.8.1 Preferential attachment

In Table 3.18 are illustrated the results given by the *Preferential attachment* method, more specifically are reported the top five connections between two nodes that are the ones with the highest and the lowest scores.

First node (highest)	Second Node (highest)	Score (highest)	First node (lowest)	Second Node (lowest)	Score (lowest)
Lord Voldemort	Arthur Weasley	540	Moaning Myrtle	Xenophilius Lovegood	1
Percy Weasley	Harry Potter	528	Xenophilius Lovegood	Nicolas Flamel	1
Argus Filch	Lord Voldemort	468	Hedwig	Nicolas Flamel	1
Bellatrix Lestrange	Hermione Granger	468	Moaning Myrtle	Hedwig	1
Albus Dumbledore	Neville Longbottom	448	Moaning Myrtle	Nicolas Flamel	1

Tabella 3.18: Preferential attachment results: both the top five connections with the highest and the lowest score are reported.

The link predictions with the highest scores, knowing the story of the books, seem plausible. For example, Lord Voldemort and Arthur Weasley are enemies, the first is the main villain, while the second one is the father of the Weasley family and he's part of the alliance that fights against the former. What's more, Lord Voldemort ordered once to attack directly Arthur Weasley. So even though it's not one of the most important relationships in the books, their connection is not out of context. The same goes for the link prediction between Percy Weasley and Harry Potter since the former is not only the brother of Harry's best friend, Ron, but he is also the chief of the Gryffindor house in the first books and thus it is he that introduced Harry to all the important things about Hogwarts and their house.
For what regards link predictions with a low score, the results make even more sense. In fact, all the characters that are taken into consideration are characters that in the books do not know each other, and so the possibility to have a link between them is kind of impossible.

3.8.2 Adamic Adar index

The results, given, instead, by the *Adamic Adar index* algorithm, can be seen in Table 3.19. As for the *Preferential attachment* results, here are reported the top five best and worst prediction scores.

First node (highest)	**Second Node (highest)**	**Score (highest)**	**First node (lowest)**	**Second Node (lowest)**	**Score (lowest)**
Peter Pettigrew	Severus Snape	4.533	Lavender Brown	Mary Riddle	0.0
Lord Voldemort	Arthur Weasley	4.45	Fluffy	Mary Riddle	0.0
Argus Filch	Lord Voldemort	4.15	Lavender Brown	Nicolas Flamel	0.0
Percy Weasley	Harry Potter	3.69	Lavender Brown	Cornelius Fudge	0.0
Albus Dumbledore	Fleur Delacour	3.44	Moaning Myrtle	Bartemius "Barty" Crouch Sr.	0.00

Tabella 3.19: Adamic Adar index results: both the top five connections with the highest and the lowest score are reported.

The results are quite similar to the one of *Preferential attachment* algorithm: some of the predictions in the ones with the highest scores are even the same, like the one between Lord Voldemort and Arthur Weasley, or the one between Percy Weasley and Harry Potter and also between Argus Flitch and Lord Voldemort. As mentioned before, these results are possible, considering the interactions present in the books. The predictions with the lowest scores are

different from the one predicted by *Preferential attachment* method, but most of the characters that are involved are the same. A low score makes sense, because these characters do not even know each other.

3.8.3 Jaccard similarity

Exploiting the Jaccard similarity to predict the new edges, the results are visible in Table 3.20:

First node (highest)	**Second Node (highest)**	**Score (highest)**	**First node (lowest)**	**Second Node (lowest)**	**Score (lowest)**
Aragog	Fluffy	1	Moaning Myrtle	Bartemius "Barty" Crouch Sr.	0.0
Aragog	Grawp	1	Bartemius "Barty" Crouch Sr.	Cho Chang	0.0
Grawp	Fluffy	1	Dobby	Tom Riddle Sr.	0.0
Moaning Myrtle	Hedwig	1	Dobby	Nicolas Flamel	0.0
Peter Pettigrew	Severus Snape	0.813	Dobby	Xenophilius Lovegood	0.00

Tabella 3.20: Jaccard similarity results: both the top five connections with the highest and the lowest score are reported.

This last table is very different from the previous two and the only connection in common is Peter Pettigrew with Severus Snape from the highest score, Moaning Myrtle with Bartemius "Barty" Crouch Sr. from the lowest score and the couple Moaning Myrtle and Hedwig which went from the lowest scores to the highest.
Having read the books, it is possible to evince the previous two tables generate better results because in this network there are presents all the possible edges and the other tables could predict better some links that actually occur or not occur in the books, while this table predicts better the lowest score. Considering the method didn't read the book, of course, and it computes the links agnostically, the result is still good because the characters involved are actually quite similar and have something in common: for example Aragog, Fluffy and Grawp are all characters related to Rubeus Hagrid, the first two are his pets and the last one is his giant brother.

3.9 Link prediction between the graph and new nodes

3.9.1 Adding one node with one edge

Exploiting the Barabasi Albert method to add one single node with just one edge, the node connects with Tom Riddle Sr. and this is a strange result because Tom Riddle Sr.'s degree is just 2 and it's a minor character, so the connection is unexpected, but if we compute the link score with the preferential attachment method, it is possible to notice the score is 3, which is very low.

3.9.2 Adding one node with ten edges

Trying to add just one single node with ten edges and the result is in the table 3.21:

Link	**Score**
Harry Potter	490
Hermione Granger	370
Albus Dumbledore	330
Sirius Black	240
Severus Snape	150
Argus Filch	140
Bellatrix Lestrange	140
Vincent Crabbe Sr.	100
Vernon Dursley	50
Tom Riddle Sr.	30

Tabella 3.21: Adding one node with ten edges.

In this case, the result seems better because the node linked with more central nodes, like Harry Potter and Hermione Granger. In fact, the scores are higher too.

3.9.3 Adding ten nodes with x edges, where x is the average number of the edges per node

In this last case, ten nodes with the number of edges equal to the average degree per node, which is 10, have been added. Altogether, the top ten most linked characters are in Figure 3.18:

Once again, like the second case, it is possible to notice the nodes linked with the most central nodes and, in particular, with Harry Potter, which is the most central node in general, taking into consideration all the centrality.

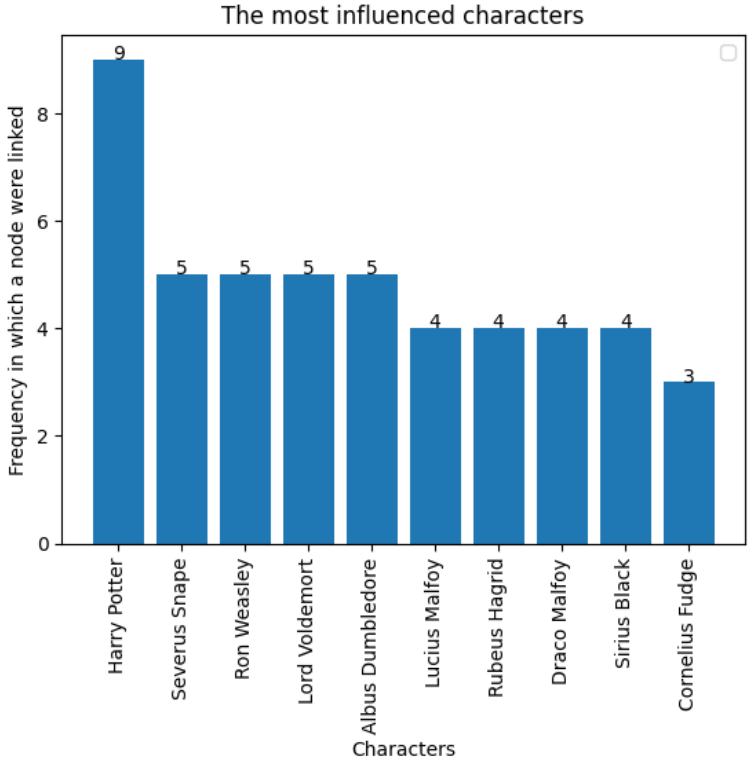

Figura 3.18: Top 10 most influenced nodes, adding 10 nodes.

4 Conclusions

In this project, the basic characteristic, metrics, properties and structure of this network have been analyzed.
We can conclude our small world network is not particularly dense, but, nevertheless, the nodes are strongly connected, especially in some area of the network, since every of the most central nodes is connected between each other and this causes many triangles, an high clustering coefficient and a low average shortest path.
The top five central nodes are the same for all the five centrality methods analyzed, even though not always with the same order, and these nodes are: Harry Potter, Hermione Granger, Ron Weasley, which are the main character of all the adventures, Lord Voldemort, which is the major antagonist, and Albus Dumbledore, which is the principal of the magic school of Hogwarts and he helps the trio to beat the enemy.
It is possible to affirm the network has a strong homophily regarding the Gryffindor house and good side attributes, good homophily for Slytherin house and bad side attributes and very low level of homophily for the other attributes. Overall, the level is medium, higher for the side attribute, than the house, while, considering the Pearson correlation, the value is very close to zero, both considering the attributes separated or together.
The most similar nodes link are between the Weasley family member.
An overview about the triangles, relationship type and the most loved/hated characters have been made.
The majority of the community detection techniques didn't find many communities, but this is justified by the fact the network seems an unique big cluster, considering the nodes are very connected from each other, so it's difficult to divide the graph into many clusters. Nevertheless, the community detection which found a good number of communities with the higher modularities, are *Luovain* and *Greedy modularity communities* methods, even though the scores are still low.
The network, thanks to the high connectivity between the nodes, is pretty robust and the most robust method is the one based on the homophily.
Regarding link prediction, the results highlighted it is possible to add new links, both between the pre-existed nodes or new nodes, efficiently and easily, but all methods used turned out to be valid, without big distinctions.

Bibliografia

[1] Centrality explanation. https://neo4j.com/. Accessed: 2023-08-13.

[2] Centrality formulas. https://www.sci.unich.it/~francesc/teaching/network/. Accessed: 2023-08-13.

[3] Homophily. https://en.wikipedia.org/wiki/Main_Page. Accessed: 2023-08-13.

[4] Infomap. https://www.statworx.com/en/content-hub/blog/community-detection-with-louvain-and-infomap/. Accessed: 2023-08-14.

[5] Girvan Newman. https://medium.com/analytics-vidhya/girvan-newman-the-clustering-technique-in-network-analysis-27fe6d665c92. Accessed: 2023-08-14.

[6] V. Tiselko, O. Dogonasheva, A. Myshkin, N. Khoroshavkina, and O. Valba. K-clique percolation in human structural connectome. *Journal Name*, X(Y):Z, 2022.

[7] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

[8] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[9] Mingshan Jia, Bogdan Gabrys, and Katarzyna Musial. Directed closure coefficient and its patterns. *Plos one*, 16(6):e0253822, 2021.