



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

# **Network Analysis of Game of Thrones Characters**

*CSE3021*  
*Social and Information Networks*  
*Fall 2022*  
*Dr.Jani Anbarasi*

Santhosh Narayanan B - 20BCE1309

Krishna Prasad Y V S Purama - 20BCE1421

Ambati Sesa Sai Shaithya - 20BCE1605

Hrithik D - 20BCE1689

Navur Sai Charan - 20BCE1752

<b><i>S.NO</i></b>	<b><i>Topic</i></b>	<b><i>Page</i></b>
1	<i>Abstract</i>	3
2	<i>Literature Review</i>	4
3	<i>Introduction</i>	6
4	<i>Algorithm</i>	7
5	<i>Modules</i>	8
6	<i>Chosen Approach</i>	9
7	<i>Procedure</i>	12
8	<i>Network Function</i>	13
9	<i>Implementation</i>	14
10	<i>Conclusion</i>	24
11	<i>References</i>	24'

## *Abstract*

Using mathematical graph theory, network analysis is a mathematical technique that evaluates and establishes unique and important connections within a network. In this field, quantitative analysis frequently outweighs qualitative analysis. Using this network theory approach, we will analyse the characters in one of the most popular television series ever, Game of Thrones. It has more than 700 characters and supporting roles, making it one of the most cast-intensive TV dramas ever, putting network theory expertise to use. We would conduct extensive research on these characters to determine the key players and roles for each season. In this instance, the characters are seen as nodes in a network, and the number of interactions is represented by the weight of the edges.

## Literature Review

Author/Year	Title	Methodology	Observation	Parameters
Dianbo Liu and Luca Albergante, 2018	Balance of thrones: a network study on Game of Thrones	Construction of relationship network and then identifying the dynamics of the network which involves comparing nodes, network graph to show the relationships of characters.	In further analysis, the authors try to find out the relation between audience engagement and network features and represent it as a graph.	Structural balance S, unpredictability score U.
Lei Ding and Alper Yilmaz, 2010	Learning Relations among Movie Characters: A Social Network Perspective	Attempt to construct social networks, identify communities and find the leader of each community in a video sequence from a sociological perspective using computer vision and machine learning techniques.	The framework was able to successfully determine the leader of a community of characters in a movie and that it can be applied in other domains as well such as video surveillance	Gaussian process, inter character affinity, F1 measures
Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu, 2010	Movie analysis based on roles' social network	The first thing that the research does is constructing the social network and reading it, after that it identifies the leading roles in the show or movie and identifies its community along with detecting the storyline of the given movie.	The authors construct a roles' social network and identify the embedded community. The results show the effectiveness and capability to handle errors of the system	Weight matrix, centrality measures, robustness
Tong Zhao, 2019	Understanding Gender Inequality in Movie Industry using Social Network Analysis and Machine Learning	<b>Bechdel</b> test is used to identify the equality or inequality, it is a set of parameters that if qualified then the movie is not gender biased.	The paper finds that female actors generally occupy less important social positions and plays a less important role in	Degree, Closeness centrality, betweenness centrality and bechdel test.

		A dataset of movies with script is used and its social network is constructed along with the calculation of various centrality measures and finally results are calculated.	the social networks in almost all genres. This is even prevalent in Romance and Family movies. So, social network analysis can help in predicting gender disparity among movies.	
Krauss, Jonas; Nann, Stefan; Simon, Daniel; Fischbach, Kai, 2010	Predicting Movie Success and Academy Awards through Sentiment and Social Network Analysis.	Proposes a new web mining approach that combines social network analysis and sentiment analysis, the authors conducted many experiments by examining the correlation of the social network structure with external metrics such as box office revenue and Oscar Awards and then predicting the success and award getting possibility using the movie content.	Authors are able to find that discussion patterns on IMDb predict Academy Awards nominations and box office success. Two months before the Oscars were given they were able to correctly predict nine Oscar nominations using the new web mining approach (social network + sentiment analysis).	Oscar model, positive index, correlation, buzz model (these models have separate formulas to check for movie success)

## *Introduction*

### *Social Network Analysis*

A Social Network is a combination of appropriate actors or nodes, who are connected to each other and form a Network relation. Here, we assign the Characters as the nodes and observe and study the pattern and trends made. The corresponding approach is based on the intuitive idea that characters' placement in social network networks has significant effects on those other characters. We are looking to identify a variety of trends in our analysis. And they work to identify the circumstances that one of those patterns emerges and to ascertain the effects.

The social network approach is based on the intuitive idea that characters' placement in social network patterns has significant effects on those other characters. We are looking to identify a variety of trends in our analysis. Additionally, they seek to understand the causes of these patterns as well as their effects. A group can use SNA to complete the process of "understanding and grasping the networks that operate in a specific field". Thanks to this complex type of mind mapping, the group is able to highlight the communication patterns within the network as well as identify networks.

The base structure of the plot is formed by the relationships between the characters. As a result, plot information such as major roles and corresponding cliques can be determined through social network analysis among characters. Most books progress through characters, and the author narrates the story and relationships between characters through relationships through them. As a result, the thread connecting the characters which weave through and from the whole is more effective than co-appearance in establishing character social networks.

## *Algorithm*

- To combat the problem, we would employ a step-by-step process. We would use some data science techniques in our project because it is a topic that requires a lot of data.
- The first step would be to load the data set into the proper environment form. We would employ some common data-cleansing methods after loading the data to enhance the accuracy of our dataset. The normal strategy for cleaning our data would be to remove outliers, fill in incomplete data, and remove redundant data using data and quantitative techniques.
- We would perform some visual analytical review after getting the data organised and ready utilizing our environment graph charting capabilities. After gaining a solid understanding of what the dataset is trying to tell us, we will use network analysis approaches to address the core issue.
- As this was a really well-liked TV programme, many data lovers all around the world have examined it. After finishing our analysis, we'll try to draw some previously undiscovered conclusions from our data.

Python would be used in the creation of our project. Google Co-laboratory would be the IDE of choice because it is free software and offers a Linux environment. Additionally, it enables the usage of GPU as a hardware booster since executing codes on a large dataset necessitates GPU for quicker and more effective outcomes that are not possible with standard personal computers.

## ***Modules***

1. Pandas:

A Python package that provides several data structures and methods for handling arithmetical data and longitudinal data is an accessible library that is developed over the NumPy library. Our dataset, which is in CSV format, is loaded into pandas in the structure of a data frame, enabling analysis in Python really simple. For data cleansing and visualisation, we would use a number of pandas methods, such as `pd.read_csv()`, `pd.describe()`, and `pd.fillna()`.

2. NumPy:

We will use NumPy array to support pandas as the majority of its actions are array-based.

3. Matplotlib.pyplot:

To obtain a strong visual exploratory study, we would use this package for visual analytics.

4. Sea Born:

This is a good framework for creating graphs that go well and creating some maps utilizing our data.

5. NetworkX API:

We would be able to incorporate network analysis paradigms with the aid of a very crucial tool in our project. In addition, we may make use of its potent graph-creation features.



## ***Chosen approach***

### **Centrality measures:**

Centrality measures are essential for comprehending networks, which are commonly referred to as graphs. These methods perform use graphs to identify the importance of any particular node in a network. They all work in different ways, but they all show parts of the network that need attention by sifting through noisy data.

### **Degree Centrality:**

The degree centrality of a node is the number of links that lead to it. In-degree and out-degree are two different measurements of centrality measures that are defined if the network is directed. The number of edges a node has determined its degree of centrality, which is the easiest to compute. The higher the degree, the more centrally positioned the node is.

---

$$C_D(v_i) = d_i = \sum_j A_{ij}$$

### **Betweenness Centrality:**

Betweenness centrality quantifies how dependent a vertex is on links between other vertices. Because they have more power over how information moves between nodes, vertices with high betweenness may have a massive impact (importance) inside a network.

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

**Closeness Centrality:**

The number of edges or links connecting two nodes (a and b) on the shortest path (i.e., path with the fewest edges) between them is known as the geodesic distance, or d. Geodesic length can be defined mathematically as follows:

$$C_c(p_i) = \frac{N-1}{\sum_{k=1}^N d(p_i, p_k)}$$

**Page Rank:**

The Page Rank score for a given page/node is based on the links made to that said page/node from other pages/nodes. The links to a given page/node are called the backlinks/in-degrees for that page/node. The web/social network thus becomes a democracy where pages/nodes vote for the importance of other pages by linking to them. It is a variant of the Eigenvector value, but because it uses backlinks/in-degrees it is used in directed networks. Directed networks are networks that allow handles (the node or webpage) to follow another without that page or node following back.

**Average shortest path:**

The average number of steps along the shortest paths for all potential pairs of network nodes is known as average shortest-path length, which is a notion in network topology. It is a way to gauge how well people can move large amounts of data through a network.

**Diameter:**

Diameter is the length of the longest path (in a number of edges) between two nodes.

**Transitivity:**

A network's transitivity or clustering coefficient serves as a gauge for the nodes' propensity to group together. A network with high transitivity has internal node communities or clusters that are closely connected to one another.

**Density:**

Density refers to the "connections" between participants. Density is defined as the number of connections a participant has divided by the total possible connections a participant could have

$$\text{Network Density:} \\ \frac{\text{Actual Connections}}{\text{Potential Connections}}$$

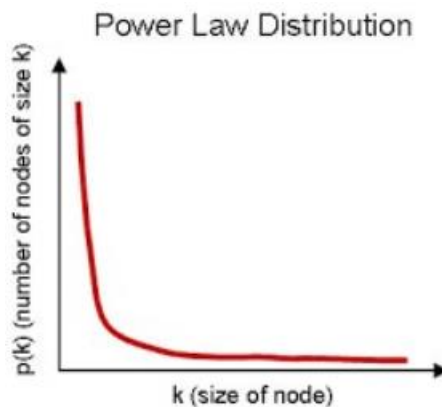
**Clustering Coefficient and Average Clustering Coefficient:**

A graph's local clustering coefficient measures how near a vertex's neighbours are to forming a clique (complete graph).

$$C_i = \frac{2 n_i}{k_i (k_i - 1)}$$
$$\langle C \rangle = \frac{1}{N} \sum_i C_i$$

**Power Law:**

In statistics, a power law is a functional relationship between two quantities, where a relative change in one quantity results in a proportional relative change in the other quantity, independent of the initial size of those quantities: one quantity varies as a power of another.



## ***Procedure***

### **Step 1:**

Load the dataset into a pandas data frame using `pd.read_csv()`.

### **Step 2:**

Once we've got the data loaded as a pandas knowledge frame, it is time to make a network. we'll use NetworkX, a network associateanalysis library, and make a graph object for the primary book.

### **Step 3:**

Measure the importance of a node in a very network by gazing at the number of neighbours it has, that is, the number of nodes it's connected to. For example, a potent account on Twitter, wherever the follower-following relationship forms the network, is an account that incorporates a high number of followers. This life of importance is named degree spatial relation.

### **Step 4:**

Various different measures like betweenness centrality and PageRank to seek out vital characters in our Game of Thrones character co-occurrence network and see if we are able to uncover a lot of attention-grabbing facts regarding this network. We'll plot the evolution of the betweenness centrality of this network over the five books. Here we tend to take main characters from one to 5 books and think about them as nodes.


The number of links occurring upon a node, or degree centrality (i.e., the number of ties that a node has). The degree is often understood in terms of the fast danger a node faces from catching no matter what is passing across the network (such as a virus, or some information). we frequently construct 2 distinct lives of degree spatial relation, specifically in-degree and out-degree, for a directed network (where ties have direction). First, we tend to measure the importance of a node in a very network by gazing at the number of neighbours it has, that is, the number of nodes it's connected to. For instance, a Twitter account with several followers is taken account vital since the interaction between followers and followers creates the network. Degree centrality is the name given to the present metric of significance.

### ***Network Functions used:***

A two-dimensional, area, potentially heterogeneous column data format with marked axes is called a "Pandas Data Frame" (rows and columns). Data is arranged in rows and columns in a data frame, which is a two-dimensional data structure. The data, rows, and columns are the three main parts of a Pandas Data Frame.

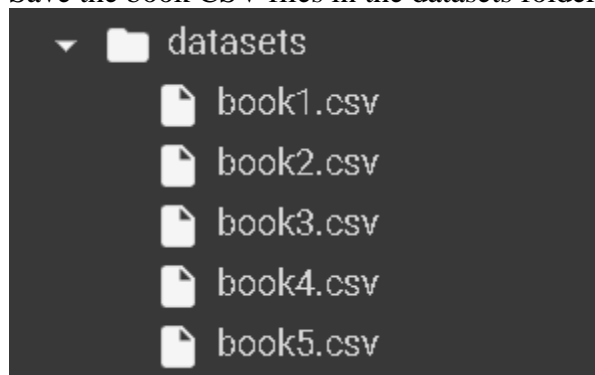
To mark out locations on a diagram, use the plot () function. The plot () function by default creates a line connecting two points. The function accepts parameters to indicate diagram points. The x-axis points are contained in an array that makes up parameter 1. The points on the y-axis are contained in an array in parameter 2.

A graph is a visual representation of a collection of things where some object pairs are linked together. Vertices are the points used to depict interconnected items, while edges are the connections between them. In this course, we go into great detail on the many words and functions related to graphs.

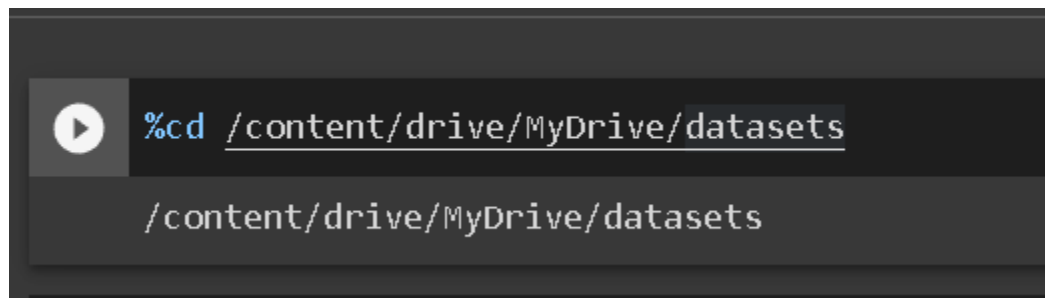
1 to 10 of 684 entries <span>Filter</span> 				
Source	Target	Type	weight	book
Addam-Marbrand	Jaime-Lannister	Undirected	3	1
Addam-Marbrand	Tywin-Lannister	Undirected	6	1
Aegon-I-Targaryen	Daenerys-Targaryen	Undirected	5	1
Aegon-I-Targaryen	Eddard-Stark	Undirected	4	1
Aemon-Targaryen-(Maester-Aemon)	Alliser-Thorne	Undirected	4	1
Aemon-Targaryen-(Maester-Aemon)	Bowen-Marsh	Undirected	4	1
Aemon-Targaryen-(Maester-Aemon)	Chett	Undirected	9	1
Aemon-Targaryen-(Maester-Aemon)	Clydas	Undirected	5	1
Aemon-Targaryen-(Maester-Aemon)	Jeor-Mormont	Undirected	13	1
Aemon-Targaryen-(Maester-Aemon)	Jon-Snow	Undirected	34	1
Show <span>10</span> per page <span>1</span> 2 10 60 69				

## *Implementation*

Save the book CSV files in the datasets folder in the drive.



Go to the location of the datasets



Importing required libraries for the project

```
[ ] %load_ext autoreload
    %matplotlib inline
    %config InlineBackend.figure_format = 'retina'

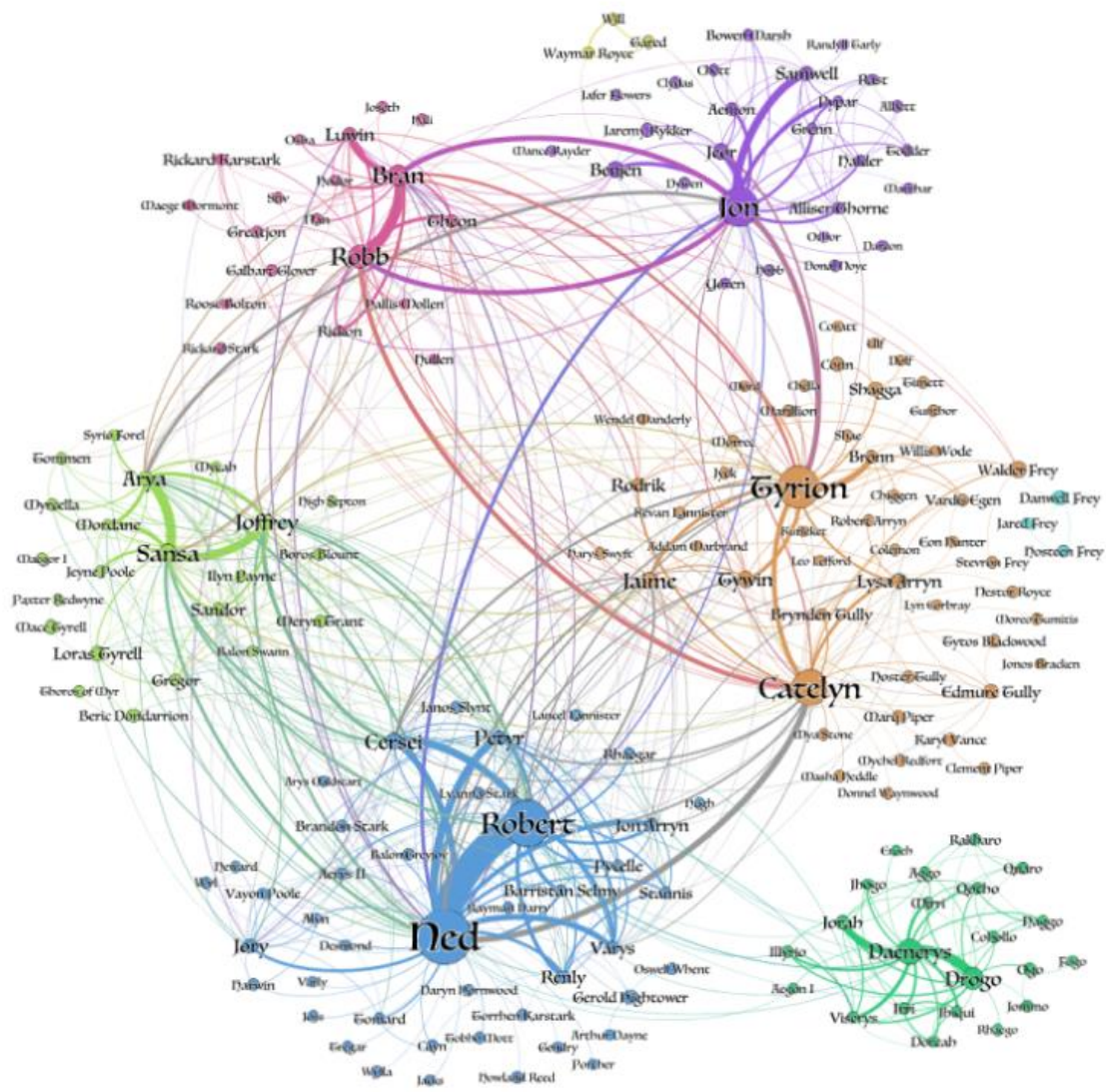
    from itertools import combinations
    from collections import defaultdict
    import networkx as nx
    import pandas as pd
    import matplotlib.pyplot as plt
    import community
    import warnings
    warnings.filterwarnings('ignore')
```

Storing the books in the all\_books array

```
[ ] recommended = defaultdict(int)

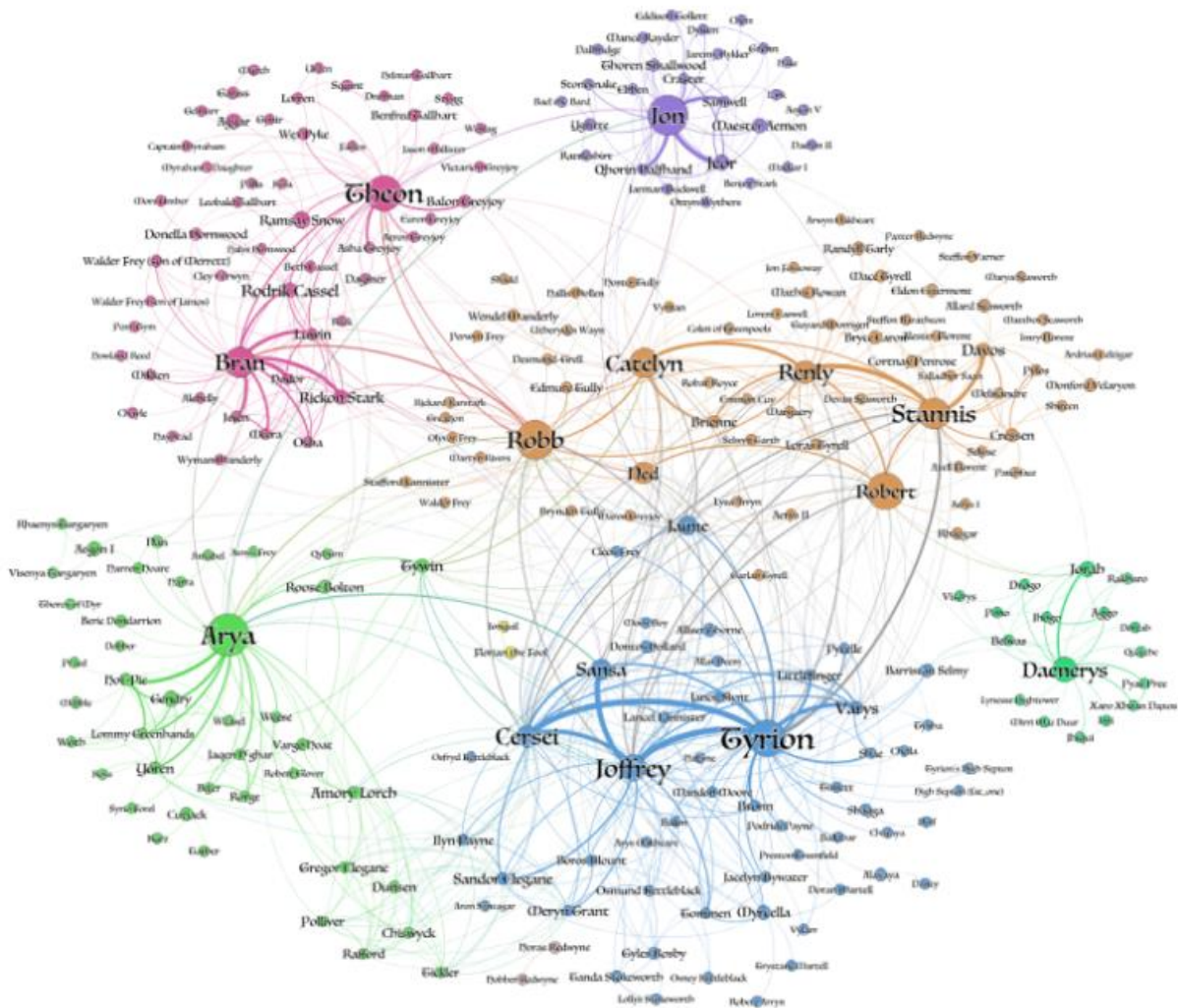
all_books = ["book1.csv", "book2.csv", "book3.csv", "book4.csv", "book5.csv"]
```

## Graphs of Books 1

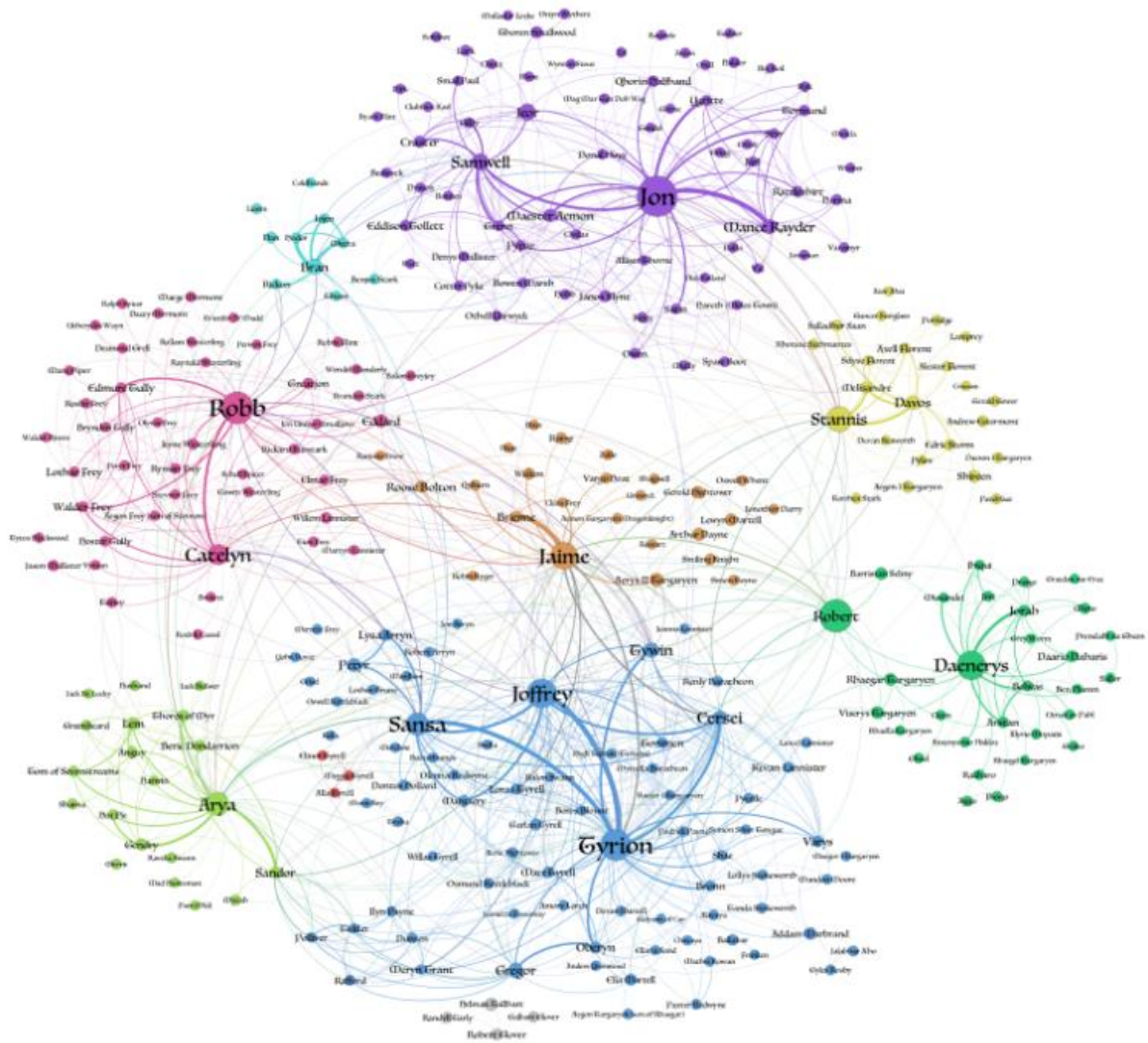




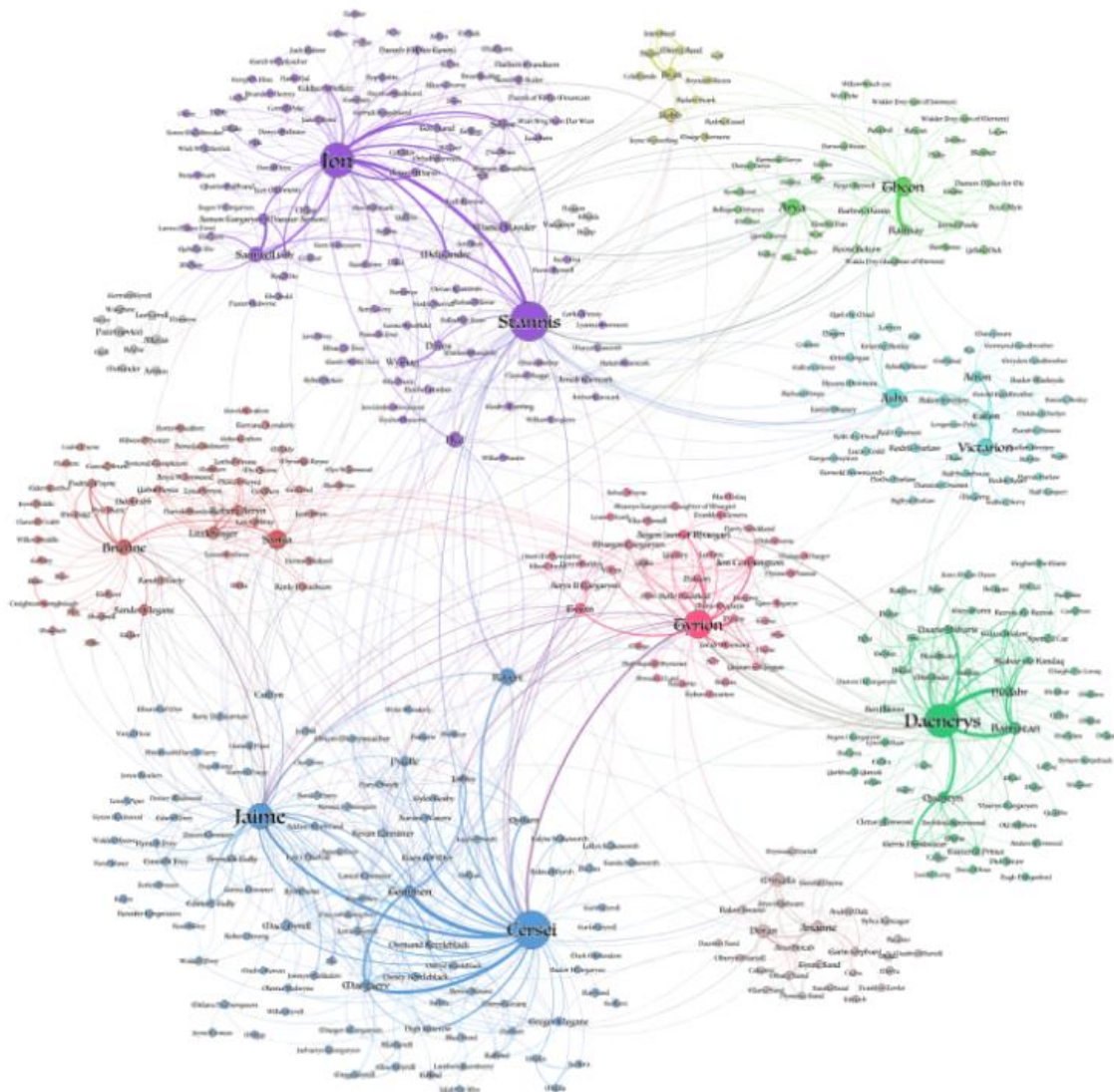
Graph of Book 2



Graph of Book 3



## Graph of Book 4/5



Creating a list containing only the source and target using pandas.

```
li = []
for f in all_books:
    tmp = pd.read_csv(f)
    li.append(tmp)

[ ] books = pd.concat(li,axis=0,ignore_index=True)
books = books[['Source','Target']]
```

Converting the list to graph G with networkx library.

```
[ ] books.drop_duplicates(subset=['Source','Target'],inplace=True)
G = nx.from_pandas_edgelist(books, source='Source',target='Target')

[ ] print("\nAll of the relations")
print(books)
```

All of the relations

	Source	Target
0	Addam-Marbrand	Jaime-Lannister
1	Addam-Marbrand	Tywin-Lannister
2	Aegon-I-Targaryen	Daenerys-Targaryen
3	Aegon-I-Targaryen	Eddard-Stark
4	Aemon-Targaryen-(Maester-Aemon)	Alliser-Thorne
...	...	...
3903	Tyrion-Lannister	Yezzan-zo-Qaggaz
3904	Tyrion-Lannister	Ysilla
3905	Tywin-Lannister	Wylis-Manderly
3906	Victarion-Greyjoy	Wulfe
3908	Yandry	Ysilla

[2823 rows x 2 columns]

Total Characters, interactions, average shortest path, diameter, and density of the network.

```
[ ] print("\nTotal number of characters in the books: ",len(G.nodes()))  
/nTotal number of characters in the books: 796  
  
[ ] print("\nTotal number of interactions in all the books are: ",len(G.edges()))  
Total number of interactions in all the books are: 2823  
  
[ ] print("\nAverage of the shortest path is ",nx.average_shortest_path_length(G))  
Average of the shortest path is 3.416225783003066  
  
[ ] print("\nDiameter of entire network: ",nx.diameter(G))  
Diameter of entire network: 9  
  
[ ] print("\nDensity of the network: ",nx.density(G))  
Density of the network: 0.008921968332227173
```

Average Clustering and transitivity of the network.

```
[ ] print("\nAverage clustering of the network: ",nx.average_clustering(G))  
Average clustering of the network: 0.4858622073350485  
  
[ ] print("Transitivity of entire network: ",nx.transitivity(G))  
Transitivity of entire network: 0.2090366938564282  
  
[ ] between centrality = nx.betweenness centrality(G)  
  
[ ] degree centrality = nx.degree centrality(G)  
  
[ ] page_rank = nx.pagerank(G)  
  
[ ] closeness centrality = nx.closeness centrality(G)
```

Top 10 Values in-betweenness, degree, closeness centrality and page rank.

```
08 print(sorted(between centrality.items(),key=lambda x:x[1],reverse=True)[0:10])
[('Jon-Snow', 0.19211961968354493), ('Tyrion-Lannister', 0.16219109611159815), ('Daenerys-Targaryen', 0.11841801916269228),
]

08 [22] print(sorted(degree centrality.items(),key=lambda x:x[1],reverse=True)[0:10])
[('Tyrion-Lannister', 0.15345911949685534), ('Jon-Snow', 0.14339622641509434), ('Jaime-Lannister', 0.1270440251572327), ('
]

08 [23] print(sorted(page_rank.items(),key=lambda x:x[1],reverse=True)[0:10])
[('Jon-Snow', 0.018999569248566855), ('Tyrion-Lannister', 0.01834123261931105), ('Jaime-Lannister', 0.015437447356269758),
]

08 [24] print(sorted(closeness centrality.items(),key=lambda x:x[1],reverse=True)[0:10])
[('Tyrion-Lannister', 0.4763331336129419), ('Robert-Baratheon', 0.4592720970537262), ('Eddard-Stark', 0.455848623853211),
]

[25] G_books = []
for book_name in all_books:
    book = pd.read_csv(book_name)
    G_book = nx.Graph()
    for _,edge in book.iterrows():
        G_book.add_edge(edge['Source'],edge['Target'],weight=edge['weight'])
    G_books.append(G_book)

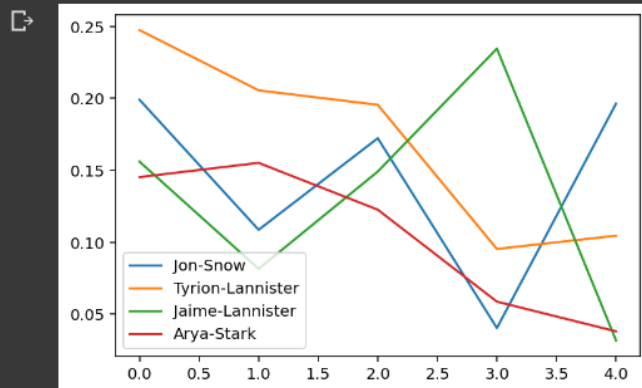
[ ] evol = [nx.degree centrality(book) for book in G_books]

[ ] degree_evol_df= pd.DataFrame.from_records(evol)
```



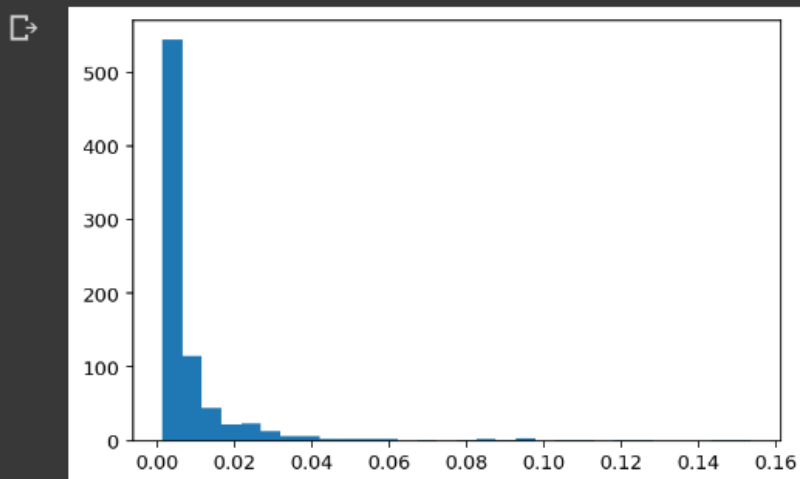
Degree centrality evolution.

```
degree_evol_df[['Jon-Snow', 'Tyrion-Lannister', 'Jaime-Lannister', 'Arya-Stark']].plot()  
plt.show()
```



The graph proves the power law.

```
plt.hist(degree centrality.values(), bins=30)  
plt.show()
```



## ***Conclusion***

We have ventured to Game of Thrones to explore the fundamental network science methods. Using an empirical approach, we discovered communities and prominent individuals in our network. Our Network analysis supported certain hypotheses and revealed fresh information about this vividly envisioned narrative. To present an alluring glimpse of network science's potential, we have developed a fantastical application. There are many more important uses, and network science promises to be crucial for comprehending our contemporary networked lives. Wider quantitative approaches to other fields of literary research, such as drama, tv, movie, rhythm, category, canonicity, literature, history, and fantasy, may be inspired by the finding of trends of plausibility, cognition, and unpredictability through computational methods.

## ***References***

<https://networkx.org/documentation/stable/index.html>

<https://networkx.org/documentation/stable/reference/index.html>

<https://www.youtube.com/watch?v=0P7QnIQDBJY>

[https://www.researchgate.net/publication/355424225\\_Network\\_Science\\_Predicts\\_Who\\_Dies\\_Next\\_in\\_Game\\_of\\_Thrones](https://www.researchgate.net/publication/355424225_Network_Science_Predicts_Who_Dies_Next_in_Game_of_Thrones)

<https://github.com/mathbeveridge/asoiaf>