I590: Data Visualization
Game of Thrones - Network Visualization
Mid-term Report

Ganesh Nagarajan | Parmeet Singh Ahluwalia | Sanjana Pukalay

Introduction

Game of Thrones (GoT) is arguably the most popular television series. It is adopted from a fiction book with the same title. The book contains a rich fictional world and ensemble of characters giving an immersive experience to the reader. However, due to diverse characters, places and belief systems, the story creates a complex network interaction between these elements of the story. Although, there were attempts to recreate the network analysis before, they haven't included much other information other than just character names. In this analysis, we propose to crawl the data from http://gameofthrones.wikia.com/wiki/Game_of_Thrones_Wiki and create networks from character information boxes to understand character interactions.

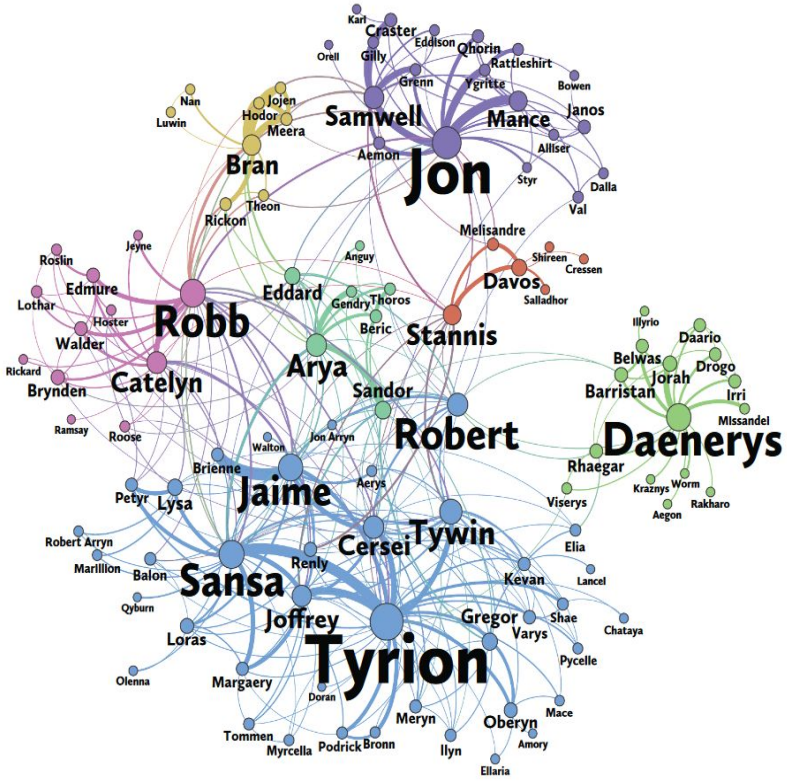
Although Game of Thrones is a work of fiction, it is inspired from real-world events. Hence, any analysis done on this dataset is directly generalizable to actual real-world problems.

Background and Existing work

This work is based on the paper published in Mathematical Association of America by Andrew Beveridge and Jie Shan, titled "Network of Thrones". The paper focusses on creating a network based on interactions between various characters. The data was extracted from the third book in the Series *A Storm of Swords*.

The network consisted of 107 vertices representing the various ladies and lords, guards and mercenaries, councilman and all the other characters. The vertices are joined by 353 integer-weighted edges where higher weights represent stronger relationships between characters. The ebook was parsed incrementing the edge weight between two characters if their names were within 15 words of one another. This created a Social Network of all the characters that interacted with each other directly or indirectly. The network contains multiple dense subnetworks, held together by a sparser global web of edges. Second, it is organized around a subset of highly influential people, both locally and globally. To Quantify this, various methods of Network Science were used. This includes Community Detection and Centrality measures. Community detection shows the distinctive communities. These communities were formed using only network structure and hence it is coherent i.e. many edges within

communities and fewer edges between communities. The global metric *modularity* was used in order to detect modularity. Centrality measures are used in order to find the most important vertices that are present. *Pagerank* was one of the variations to find them.



Given above is the *Network of Thrones* visualization which depicts the Social Network generated from a Storm of Swords novel. This is the third novel in the series and the main reason for choosing this over the other novels is because the main narrative had matured and all the characters were scattered geographically and enmeshed in their own social circles. The color of a vertex indicates a community. The size of a vertex corresponds to its Pagerank value, and the size of its label depicts its betweenness centrality. An edge's thickness represents its weight i.e. which two characters are strongly related to one another.

Research Question

The Network of Thrones visualization and paper did a good job of showing the various interactions between the various characters. It used various network statistics like in-degree, out-degree and centrality measures to represent the various aspects of the network. We have expanded on this by not only focussing on the interactions between characters but also putting

a focus on what house a character belongs to, the allegiances they have and also their belief systems[The old gods and the new].

The main Questions that are asked here are :

- Who are the top 5 most popular Houses in terms of connections?
- How are parameters like House Allegiances, Belief systems and Religion related?
- What kind of inferences can be drawn when the data is visualised graphically?

The main difference between our network and *Network of Thrones* is the place from which data is extracted. We are focussing on data that is obtained from the Game of Thrones wikia page.

Methodology

The methodology adopted for this project is very similar to the typical lifecycle of a data product. It consists of the following steps, particularly in the same order:

- Data Collection
- Data Extraction
- Data Cleaning
- Data Processing
- Data Visualization

Data Collection

The data has been obtained from the Game of Thrones wikia page. It contains all the information regarding the show including episodes, characters, culture, society and also the general opinions of people regarding the popular events in the show. Particularly, we were interested to extract information regarding individual characters since that would be the key to creating the graph or network visualization. Like every other information on wikipedia, the information on this website can also be exported and downloaded in XML format.

```
<siteinfo>
  <sitename>Game of Thrones Wiki</sitename>
  <base>http:///slave.db-b.service.consul/wiki/Game_of_Thrones_Wiki</base>
  <generator>MediaWiki 1.19.24</generator>
  <case>first-letter</case>
  <namespaces>
    <namespace key="-2" case="first-letter">Media</namespace>
    <namespace key="-1" case="first-letter">Special</namespace>
    <namespace key="0" case="first-letter" />
    <namespace key="1" case="first-letter">Talk</namespace>
    <namespace key="2" case="first-letter">User</namespace>
    <namespace key="3" case="first-letter">User talk</namespace>
    <namespace key="4" case="first-letter">Game of Thrones Wiki</namespace>
    <namespace key="5" case="first-letter">Game of Thrones Wiki talk</namespace>
    <namespace key="6" case="first-letter">File</namespace>
    <namespace key="7" case="first-letter">File talk</namespace>
    <namespace key="8" case="first-letter">MediaWiki</namespace>
    <namespace key="9" case="first-letter">MediaWiki talk</namespace>
    <namespace key="10" case="first-letter">Template</namespace>
    <namespace key="11" case="first-letter">Template talk</namespace>
    <namespace key="12" case="first-letter">Help</namespace>
```

Data Extraction

Data associated with the entities like houses, seasons, people, places etc. needed to be extracted from the huge dataset most of which contained other textual information that was not relevant for generating the graphical network. Every character in the show has a separate wiki page containing all the information related to him/her and also all the possible ways they are related to other characters and the show in general.

Every page contains what is called an ‘Info Box’ associated with it which contains the key information required to construct the graph. An example Infobox of the character ‘Daenerys Targaryen’ looks like the following:

Season(s)	1, 2, 3, 4, 5, 6, 7
First seen	"Winter is Coming"
Appeared in	49 episodes (see below)
Titles	Queen of the Andals, the Rhoynar, and the First Men (claimant) Lady Regnant of the Seven Kingdoms (claimant) Protector of the Realm (claimant) Khaleesi of the Great Grass Sea Breaker of Chains Mother of Dragons Queen of Meereen (formerly)
Also known as	Daenerys of the House Targaryen, the First of Her Name Daenerys Stormborn Dany The Unburnt *Mhysa* The Queen Across the Sea The Silver Queen The Beggar Queen The Dragon Queen The Prince That Was Promised
Status	Alive
Age	Approx. 22 in Season 7
Origin	Dragonstone (birthplace) Free Cities (raised)
Allegiance	House Targaryen
Culture	Valyrian
Father	{Aerys II Targaryen}
Mother	{Rhaella Targaryen}

All the data present in the Infobox is extracted using a parsing program written in Python and stored into a CSV file.

Data Cleaning

As observed in most of the datasets available to public, the data obtained is not clean and needs data cleaning to some extend. The advantage of using XML data is that it is semi structured and extensive pre cleaning is not required.

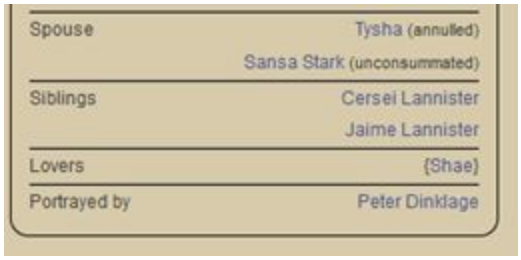
However, there were some issues with the data. An extensive part of the data is not required to create the visualization. A part of the data from the Infobox was extracted and stored in a CSV format. The CSV just contained the important components like Persons, Relations and Entities.

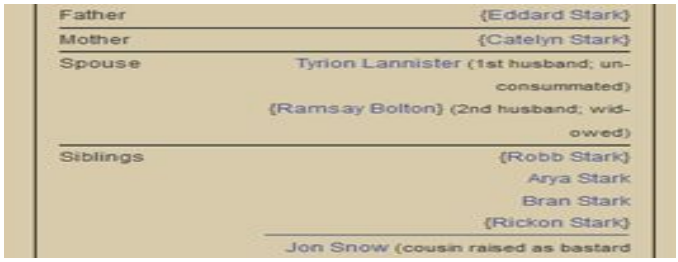
1	Eddard Stark	Actor	Sean Bean
2	Eddard Stark	Actor	Robert Aramayo
3	Eddard Stark	Actor	Sebastian Croft
4	Tyrion Lannister	Actor	Peter Dinklage
5	Tywin Lannister	Actor	Charles Dance
6	Cersei I Lannister	Actor	Lena Headey
7	Cersei I Lannister	Actor	Nell Williams
8	Joffrey I Baratheon	Actor	Jack Gleeson
9	Catelyn Stark	Actor	Michelle Fairley
10	Robert I Baratheon	Actor	Mark Addy
11	Jon Snow	Actor	Kit Harington
12	Robb Stark	Actor	Richard Madden

In the above CSV, the first column are the characters, second is the relations and third can be any entity in the show.

This reduced data had a lot of duplicate values in the sense there were multiple rows with the same values. These redundant rows were eliminated and a filtered version of the CSV was created.

There was another major issue with the data. A lot of rows were associated with each other. For example, if Person A holds a relation ‘Family’ with another Person B, then there was a corresponding row which said Person B hold a relation ‘Family’ with B. This adds a lot of noise to the data. Unique relations are required in order to obtain a clean graph. Therefore, all such relations had to be eliminated. Following is an example of this kind of noise:





Data Processing

After obtaining the clean data, it needs to be processed and converted to a format such that it can be readily used in the front end. As mentioned, all the data was in the form of a csv which now needed to be converted to JSON format to be able to connect with the front end Javascript.

1	{data:{id: '0'	lbl :'Titles'	source :'EddardStark'	target: 'LordofWinterfell'}	classes : 'Titles'}
2	{data:{id: '1'	lbl :'Titles'	source :'EddardStark'	target: 'LordParamountoftheNo	classes : 'Titles'}
3	{data:{id: '2'	lbl :'Titles'	source :'EddardStark'	target: 'WardenoftheNorth'}	classes : 'Titles'}
4	{data:{id: '3'	lbl :'Titles'	source :'EddardStark'	target: 'HandoftheKing'}	classes : 'Titles'}
5	{data:{id: '4'	lbl :'Titles'	source :'EddardStark'	target: 'LordRegent'}	classes : 'Titles'}
6	{data:{id: '5'	lbl :'Titles'	source :'EddardStark'	target: 'ProtectoroftheRealm'}	classes : 'Titles'}
7	{data:{id: '6'	lbl :'Actor'	source :'EddardStark'	target: 'SeanBean'}	classes : 'Actor'}
8	{data:{id: '7'	lbl :'Actor'	source :'EddardStark'	target: 'RobertAramayo'}	classes : 'Actor'}
9	{data:{id: '8'	lbl :'Actor'	source :'EddardStark'	target: 'SebastianCroft'}	classes : 'Actor'}
10	{data:{id: '9'	lbl :'Status'	source :'EddardStark'	target: 'Deceased'}	classes : 'Status'}
11	{data:{id: '10'	lbl :'Culture'	source :'EddardStark'	target: 'Northmen'}	classes : 'Culture'}
12	{data:{id: '11'	lbl :'Allegiance'	source :'EddardStark'	target: 'HouseStark'}	classes : 'Allegiance'}

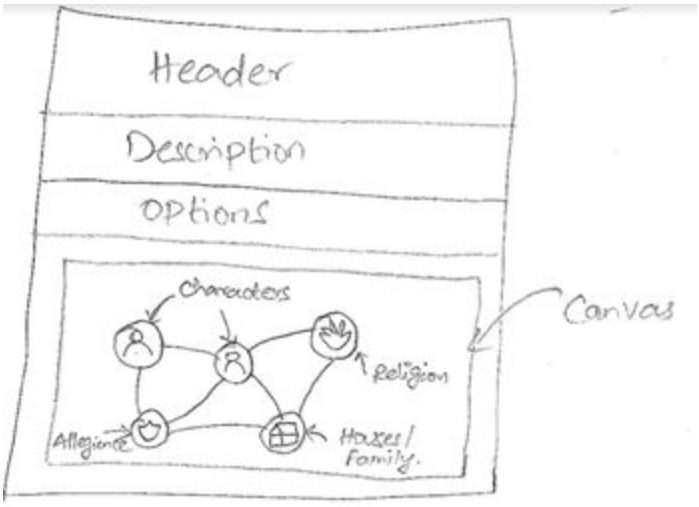
Utmost care needs to be taken while handling data pertaining to the nodes and edges since they names exactly need to match. Otherwise, the edge is not created between the nodes and the graph remains incomplete.

Data Visualization

The cleaned and processed data is visualized using a JavaScript framework called Cytoscape. It is an open source software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data. Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization.

The nodes of the graph are the Persons or Entities and the edges between them depict the relationship between the nodes. About 7602 Relations were encountered between 2236 Nodes in the graph.

The initial sketch of the visualization looks like the following:



The layout of the page is done using HTML, CSS, JavaScript and Bootstrap. At the top, we have the header showing important information about the project. Below the header is the general description explaining the visualization. Just below the description, a set of options are provided to the user to understand what their preferences are and show the visualization accordingly. The user can choose a layout of their choice and filter the graph based on various relations like Family, Religion etc. This is particularly interesting to gain insights about the data. Finally, there is an option to zoom in or out.

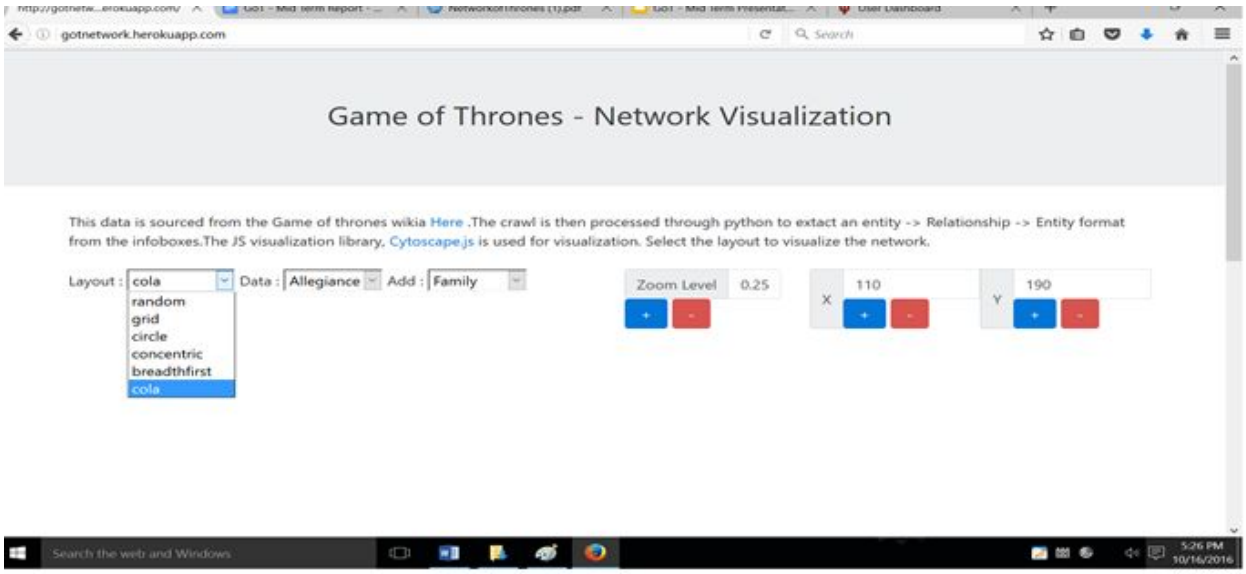
Following Visualization Layouts are available as a part of the application,

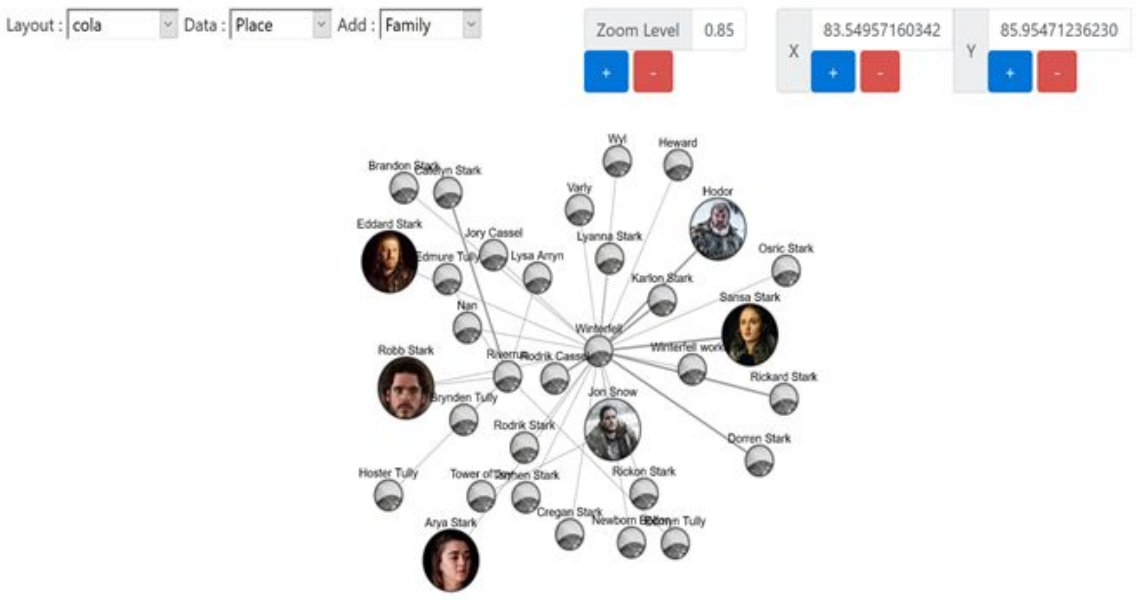
1. Random
2. Grid
3. Circle
4. Concentric
5. Breadth First Layout
6. Cola (A Force Directed Layout framework)

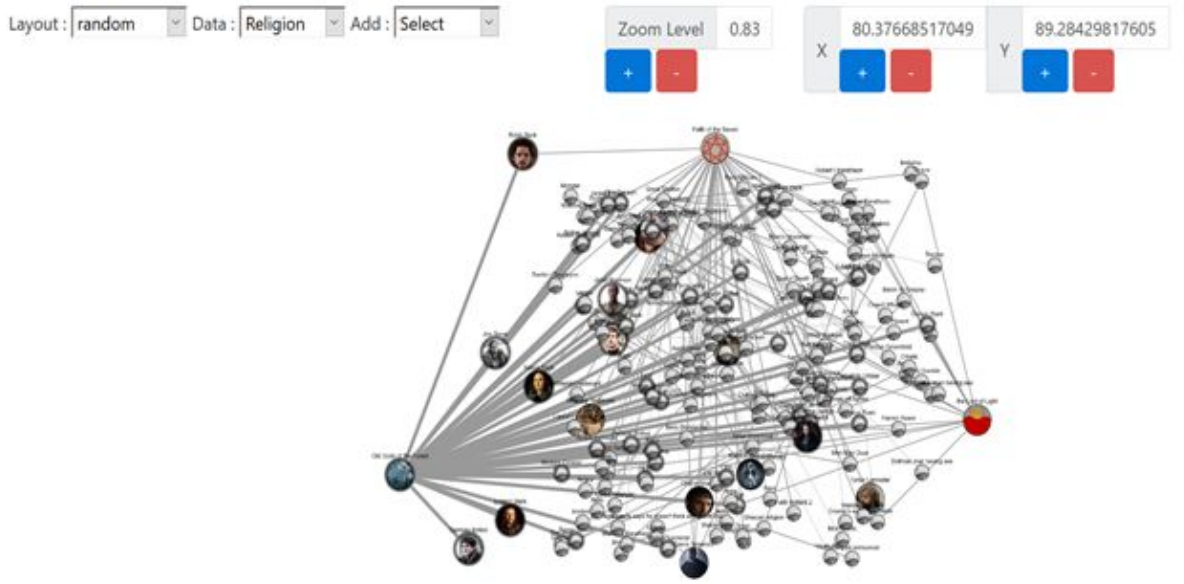
Out of these visualization, force directed framework helps to detect “communities” rather than other layouts, which emphasizes more on connections.
The first dropdown data, adds the primary data to the canvas and the Add dropdown adds any additional parameter selected so that the factor interactions can be studied. This was hosted at http://gotnetwork.herokuapp.com/ The application takes a while to load once the data option is selected because both data filtering and data manipulation happend in the cluent side rather than the server side due to the nature of javascript library design.

Results

The final results look as shown in the following screenshots:





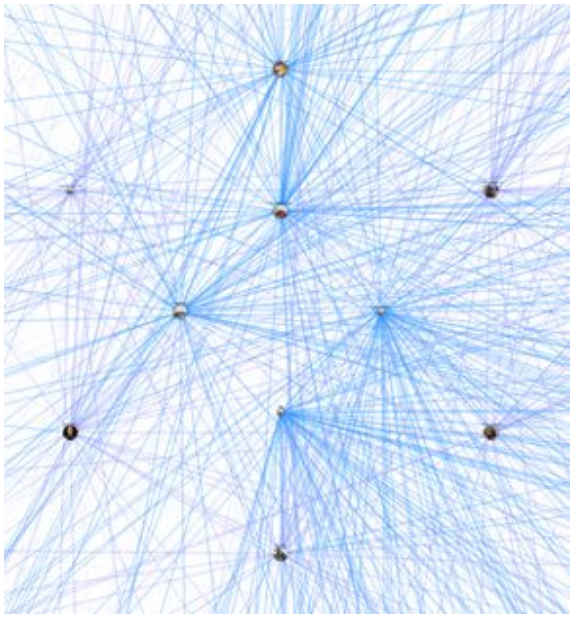


When the pointer is over a node, all its connected nodes and edges have increased focus. This helps in identifying the nodes and for easy interface.

Insights

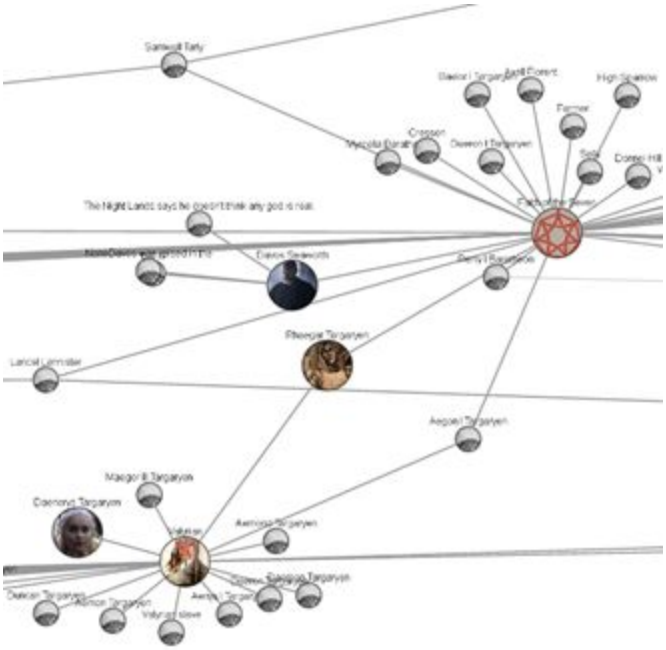
A lot of insights can be gained by looking at the visualization. After visualizing the network and seeing the various ways in which every character is related, it was startling to see some allegiances which were not apparent on the show. This was because of the data that was extracted from the wikipedia page. Difference in certain beliefs. Eg: Robb Stark was connected to the Faith of the Seven in the previous visualization. By making it a network between all the characters, we could easily show which characters are strongly connected and which are not. Certain important characters could actually be recognized using this method. Eg: Jon Snow.

Result 1.0 : Most Connected Characters



The above visualization lists most connected houses. While, Starks and Targaryens turn out to be the most connected houses as expected, the third entry “Night’s Watch” is an interesting find. This is because, in the storyline of the GoT, the Night’s Watch isn’t given a lot of importance. However, it turns out to be one of the most connected houses.

Result 2.0 : Bridge Points / Influencers



In network analysis, bridge points or connectors are always points of interest. In Marketing, these are the possible opportunities for cross selling, while in other interdisciplinary areas, cross-over points are always of some interest. In the above fig, it can be seen that the node center “Faith of the Seven” and the “Valyrian” is connected by just two nodes. From the storyline, these were kings who cross over the seas and established a new kingdom, However it is interesting because this change in the religion could affect how future characters would interact.

Also, we could find special characters like Davos, Who was brought up as religious, then turns out to be agnostic. These might be small information, which could increase the engagement of the audience and in real world scenarios create opportunities for marketing and cross-selling.

Future Work

Once the preliminary goals are accomplished, this project can be extended to perform clustering based on themes and community detection of characters.

References

[1] https://www.maa.org/sites/default/files/pdf/Mathhorizons/NetworkofThrones%20(1).pdf
[2] http://gameofthrones.wikia.com/wiki/Game_of_Thrones_Wiki