Social Network Analysis
Project Report
(CSE4008)

Network Analysis On Relationships Between Characters In Book

By

Vesalapu Likhit (190C2030068)

vesalapu.likhit.19cse@bmu.edu.in

Sura Chandra Sekhar Reddy (190C2030063)

sura.reddy.19cse@bmu.edu.in

Annamareddy Nikhil Chowdary (190C2030003)

annamareddy.chowdary.19cse@bmu.edu.in

Submitted to

Dr. Kiran Sharma
Assistant Professor

Department of Computer Science and Engineering
School of Engineering and Technology

December, 2022

TABLE OF CONTENTS

1. Background :

One of the most challenging tasks in Natural Language Processing(NLP) is to extract information and relationships from text and that's what we'll be doing in this paper. We're analyzing a fantasy story book called The Witcher.

There are several methods for extracting relationships from text from simple rule-based methods to advanced deep learning models. However we found that the deep learning models that are currently available often trained on a specific dataset for example research papers on drug discoveries. So they are not really suitable for fantasy novels, so we have decided to go for a simple baseline model that is based on Named Entity Recognition together with some custom rules as follows.

2. Problem Statement :

As all know that a book contains a number of characters and a number of relations between them. In this paper we are considering The Witcher book to find the relationship between characters using Network Analysis. Additionally, finding the communities and importance of a character over time.

3. Aim & Objectives :

- In this paper, we'll be reading The Witcher books and creating a Social Network of characters with the help of NLP and Network Analytics.
- We're going to create a network graph of all the characters and be able to answer several questions like
  (I) What are the most important characters?
  (II) How did their relative importance develop throughout the series?
  (III) What are the main communities of characters that form this Social Network?

4. Literature Review :

Elson, Dames & McKeown (2010), Ardanuy & Sporleder (2014), Bamman, Underwood & Smith (2014), and Vala et al. (2015) all use the Stanford NER tagger (Finkel, Grenager & Manning, 2005) to recognise personalities in literature. These experiments conduct Named entity recognition on a selection of Sherlock Holmes literature, with F1-scores ranging from 45 to 54. According to Vala et al. (2015), the collection's biggest fault is its abundance of minor characters, a problem that we anticipate will also exist in our collections of classic and contemporary books.

Named Entity Recognition(NER) :

Identifying the characters is the hardest part of building a social network of literary characters. NER is frequently used to find passages in texts that name objects. Additionally, identified passages are frequently divided into different categories like person, place, and organization. This method is frequently used to locate other numerical mentions, such as dates, times, money amounts, and percentages.

5. Dataset Description :

As our project is to find relations between characters in a book, we have downloaded the eight books of “The Witcher” series from the website called Witcher Wiki. So, our dataset contains text files of eight books from The Witcher series.

We have done Web Scraping to get the data of characters from all the books. After scraping the data from the witcher wiki’s website. We have saved the data as a csv file. So, in our dataset we have a characters csv file.

Eight books are used to determine the communities in the series and importance of characters over time. For example, one character may have more importance in 1st, 2nd and 5th books than another character may have his/her importance in only one book.

6. Proposed Methodology :

There are several methods for extracting relationships from text from simple rule-based methods to advanced deep learning models. However we found that the deep learning models that are currently available often trained on a specific dataset for example research papers on drug discoveries. So they are not really suitable for fantasy novels, so we have decided to go for a simple baseline model that is based on Named Entity Recognition together with some custom rules as follows.

First, we tokenized every book into a list of sentences and labeled its sentences by the name of the characters appearing in the sentence. Second, we defined a window size of how far two sentences are apart from each other and assume that if two characters are mentioned in two sentences within this window then there is a relationship between them with named energy recognition of course we will recognize all kinds of energies from text from person's name to organization's name to location and so on. And the good thing is we already know the names of all the characters from the books because we have scraped them witcher wiki website so we can easily filter out all the entities that are not characters such as names of a monster or a kingdom but if you like you can include them for extra analysis as well.

7.1 Data Preprocessing :

In the data preprocessing part, we did read the list of characters from characters.csv file which we have web scraped from the witcher wiki. We have found out the named entity in each sentence and then we added a function to filter out non-character entities in the sentences. After using this function we still had some sentences with two characters identified. So to avoid two names in one character name we have taken only the first names of the characters.

7.2 Extracting Relationships :

After processing the data, we have created a window size of 5 so that we can get a relationship between the characters in five sentences. For example, Geralt had a relationship in the first 5 sentences with Niedamir. One of the most important things we have added while creating the relationships is not to take duplicates in the given window size and if only one character is in the given window size with no relationship, it is also avoided. For example, Geralt cannot have a relationship with Geralt himself. Then all the relations are stored in a list. The list is processed so that we get the number of times one character is related to another. Suppose Geralt is related to Niemadir 10 times and Niedamir is related to Geralt 2 times, the numbers are stored in separate columns. Finally we have created the relationship dataframe(Fig 1) having “source”(column which stores the source characters), “target”(column which stores the target characters) and “value”(column which stores number of times the same source and target are related to each other) columns.

We know every edge in the network will have two components: the source node and the target node. Here we just pass our relationship dataframe and specify the source and target column and we also want to create the edge attribute. The edge attribute has different kinds of properties of the edge so here we have one property that is the “value”. So we will input the value column here for the edge attributes. Then we used Kamada Kawai layout(Fig 2) which is a type of network graph to visualize our relationships. For interactive network visualization, we have used the Pyvis library as shown in Fig 3.

7.3 Centrality Measures :

Calculated all the centrality measures like Degree Centrality, Betweenness Centrality and Closeness Centrality. Degree centrality(Fig 4) is calculated to find importance scores based simply on the number of links held by each node. Betweenness centrality(Fig 5) measures the number of times a node lies on the shortest path between other nodes. Closeness centrality(Fig 6) scores each node based on their ‘closeness’ to all other nodes in the network.

7.4 Community Detection :

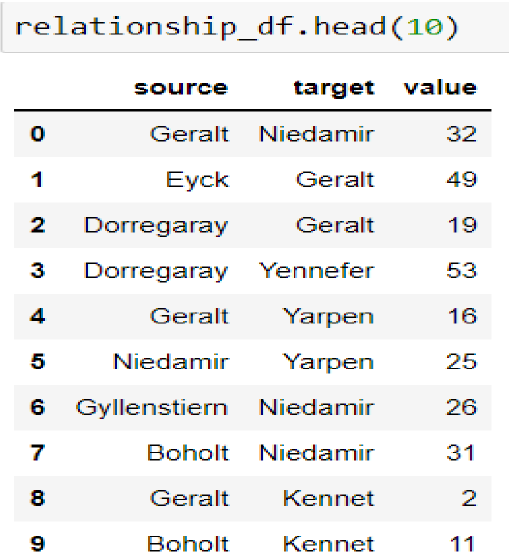
Community detection is a very useful technique in graph analytics so basically if we have a social network we can identify the communities that form this graph and from that we can get a lot of different insights about the dynamics of the network and how the different communities interact with each other. There are several algorithms available for community detection. For this we will use the louvain detection algorithm. It was visualized using Pyvis python library as shown in Fig 7.

7.5 Character’s importance over time :

We also did an analysis on the evolution of the character's importance over time throughout the book series. To do that basically we just need to create different graphs for all the different books and then we see what are the importance of characters in each of the books and then we plot them on a graph. To be able to do that we finally need to loop through all the books in our data and basically perform all the steps that we have done in this till now. We created different functions for different steps and we made a separate python file so that we can import this python file into our notebook just to avoid that our notebook would be too big. We did a for loop and created graphs for each of the books and appended those graphs into a list. And then we plotted the degree centrality evolution of some characters to find out the importance of those characters over time. The graph of some character's importance over time is shown in Fig 8.

7. Experimental Results :

The results from this experiment came quite good. As discussed in the methodology, we are able to create the relationships between characters and store them in a dataframe as shown in Fig 1. The relations are shown in the form of a network graph in Fig 2 and Fig 3.



```
relationship_df.head(10)
```

	source	target	value
0	Geralt	Niedamir	32
1	Eyck	Geralt	49
2	Dorregaray	Geralt	19
3	Dorregaray	Yennefer	53
4	Geralt	Yarpen	16
5	Niedamir	Yarpen	25
6	Gyllenstiern	Niedamir	26
7	Boholt	Niedamir	31
8	Geralt	Kennet	2
9	Boholt	Kennet	11

Fig 1 : Relationships dataframe

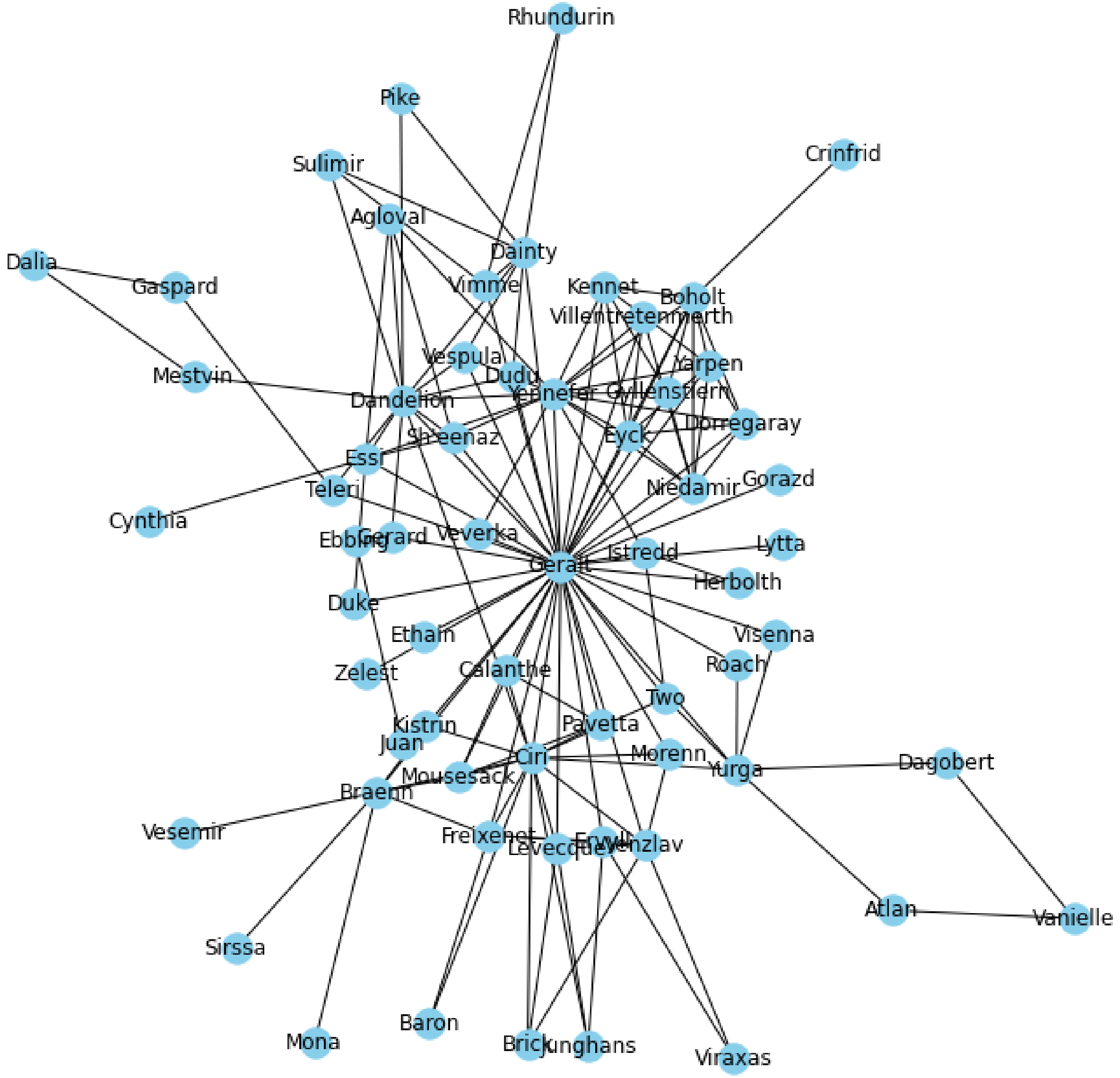


Fig 2 : Networkx Graph(kamada_kawai_layout) for relationships

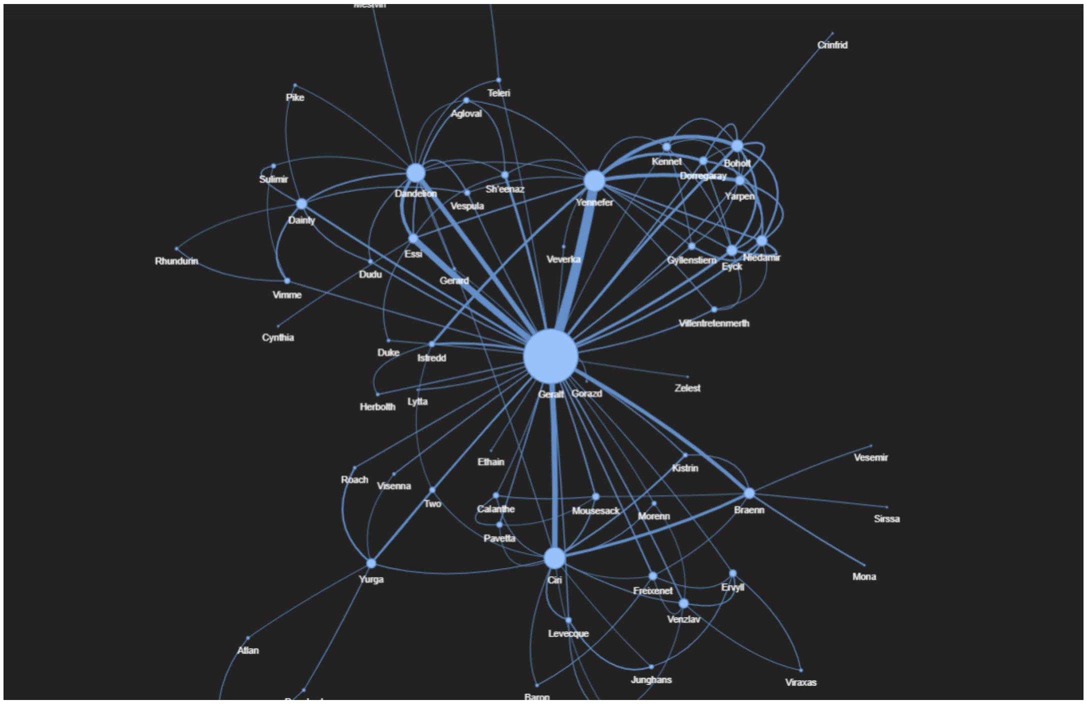


Fig 3 : Visualization using Pyvis library

Using that dataframe we are able to visualize them into Network graph and also plotted the degree centrality, betweenness centrality and closeness centrality as shown in Fig 4, Fig 5 and Fig 6 respectively.

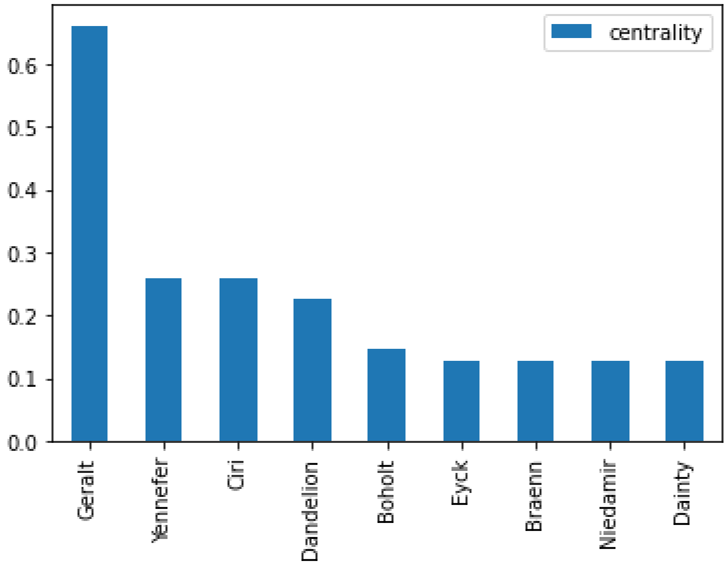


Fig 4 : Degree Centrality

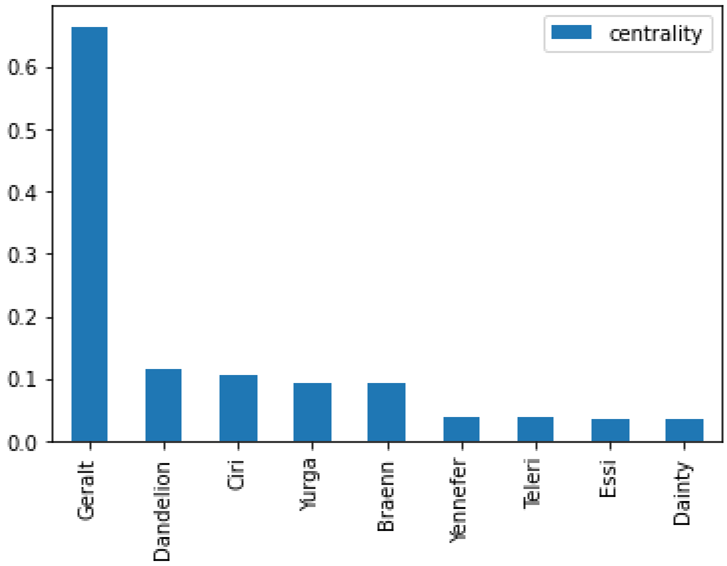


Fig 5 : Betweenness Centrality

In Fig 5 it is clearly shown that Geralt has more betweenness centrality when compared to other characters. Also, Geralt has more closeness centrality than other characters.

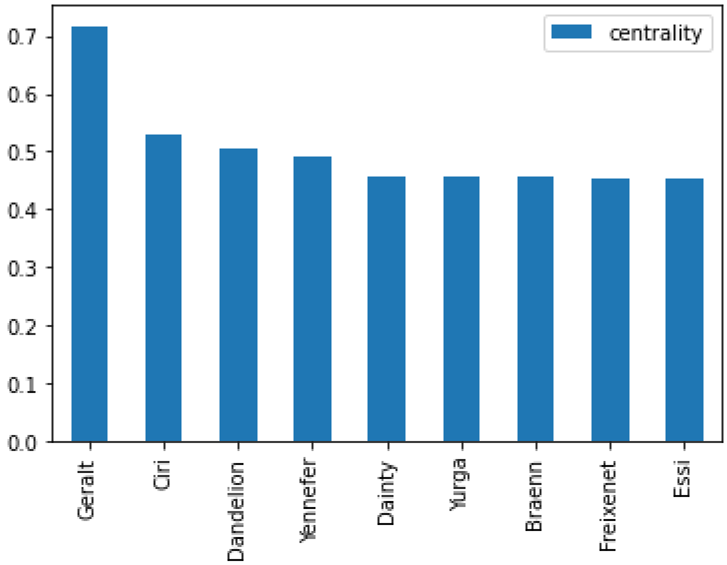


Fig 6 : Closeness Centrality

Using the relationships, we are able to find out the communities in the books and visualize them as shown in Fig 7. The importance of some of the characters over time is shown in Fig 8.

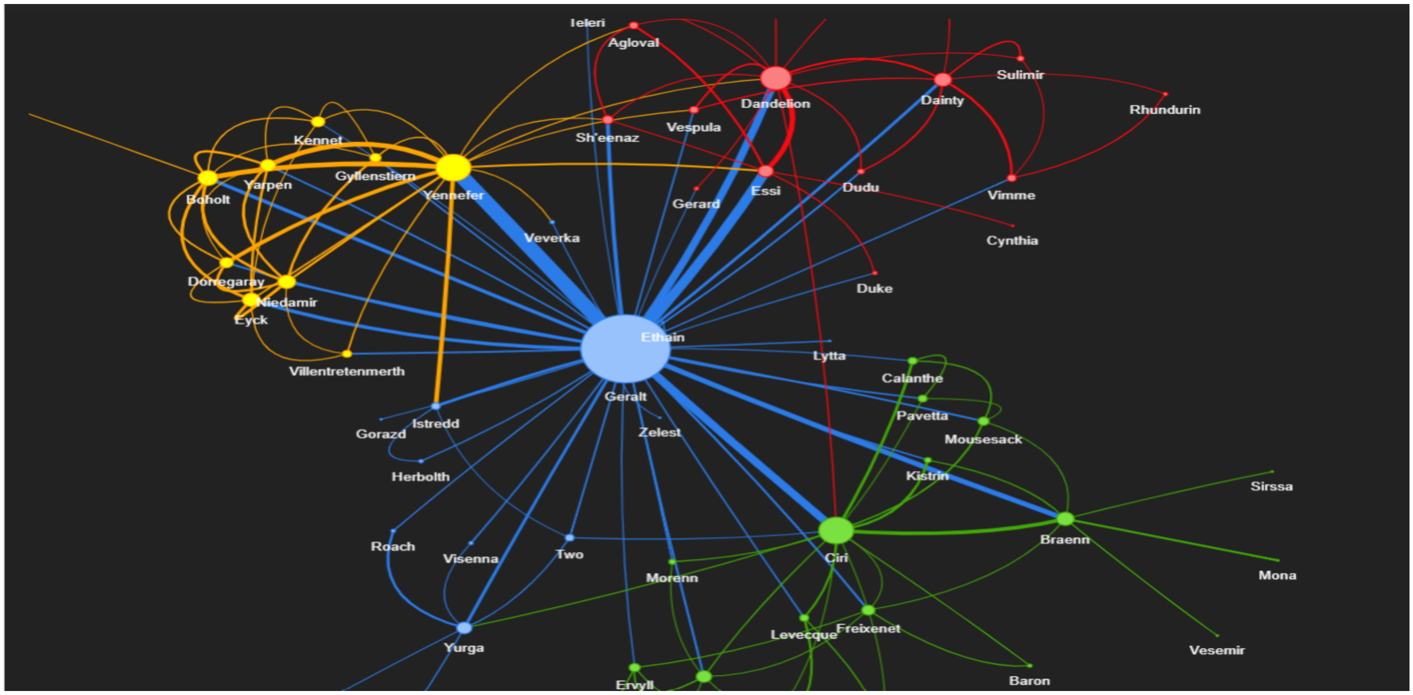


Fig 7 : Visualization of Community Detection using Pyvis

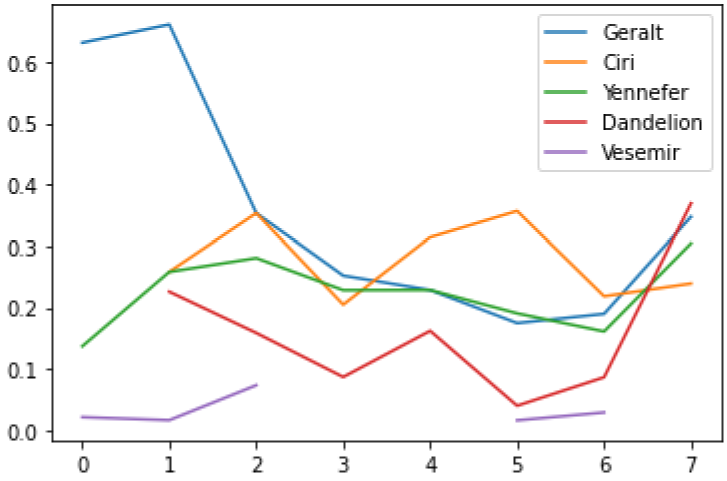


Fig 8 : Importance of characters over time

8. Conclusion and Future Scope :

This experiment helped us to use Social Network Analysis in extracting relationships between characters in a book and if there are series of books then we can find the importance of characters over time using Network Analysis. In addition to this, we have detected communities in the book.

We can further develop this paper to develop relationships for modern novels and modern story books. But we need advanced NLP models to identify the relationships.

9. Challenges :

- Data Scraping from the website using Selenium library in python.
- Data preprocessing and large data handling are the challenges we have faced in the dataset part.
- Visualization using Pyvis for the characters network.

10. References :

(1) Elson, David K., Kathleen McKeown, and Nicholas J. Dames. "Extracting social networks from literary fiction." (2010).

(2) Muhuri, Samya, Susanta Chakraborty, and Sabitri Nanda Chakraborty. "Extracting social network and character categorization from Bengali literature." *IEEE Transactions on Computational Social Systems* 5.2 (2018): 371-381.

(3) Fernandez, Matt, Michael Peterson, and Ben Ulmer. "Extracting social network from literature to predict antagonist and protagonist." *Recuperado de: https://nlp. stanford. edu/courses/cs224n/2015/reports/14. pdf* (2015).

(4) Labatut, Vincent, and Xavier Bost. "Extraction and analysis of fictional character networks: A survey." *ACM Computing Surveys (CSUR)* 52.5 (2019): 1-40.

(5) Dekker, Niels, Tobias Kuhn, and Marieke van Erp. "Evaluating named entity recognition tools for extracting social networks from novels." *PeerJ Computer Science* 5 (2019): e189.

Individual Contributions :

S. No	Name	Contribution
1.	Sura Chandra Sekhar Reddy	Code to preprocess the data and find out relationships dataframe and visualization part. Contributed in writing the report.
2.	Vesalapu Likhit	Code for community detection and visualizing it. Contributed in writing the report.
3.	Annamareddy Nikhil Chowdary	Code to find the importance of characters over time. Contributed in writing the report.