INTERNATIONAL BURCH UNIVERSITY
FACULTY OF ENGINEERING AND NATURAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGIES

SNA Harry Potter Project

Mentor

Assist. Prof. Dr. Zerina Mašetić

SARAJEVO

January 2022

Contents

Abbreviations

Social Network Analysis → SNA

Abstract

In this project, I use social network analysis to analyze one of the biggest fantasy novels Harry Potter. For this analysis, I used python libraries for analysis, and take characters from books to find their relationships.

Keywords

Python, SNA, Networkx, Harry Potter characters, books, Nodes, Edges

Introduction

Social network analysis (SNA), also known as network science, is a field of data analytics that uses networks and graph theory to understand social structures. In order to build SNA graphs, two key components are required: actors and relationships. Actors are one primary link that contains many relationships to create one network. In fact actions and relationships in network science, it reference nodes which are actors, and edges which are relationships.

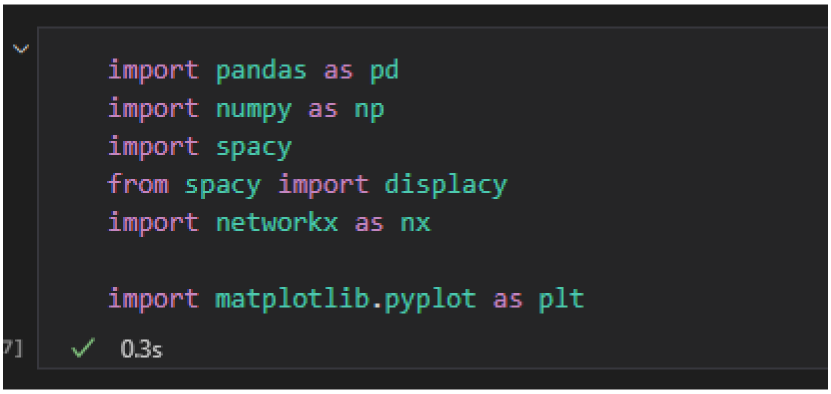
In my case nodes are characters from the novel, and relationships are where in the book they meet.

Methods

For this project I used python to analyze and create graphs, using libraries such as pandas, numpy, spacy, network, and matplotliob. For data, I found books in form of .txt which I found from the GitHub repo. And as characters I found the most common characters in every book and in that way, I found relationships.

The characters from the book are found online and sorted in a CSV file, which contains two columns, which are book and character. For books, there are seven and to get data from them I used spacy to read from those books and to find relationships of characters.

Results



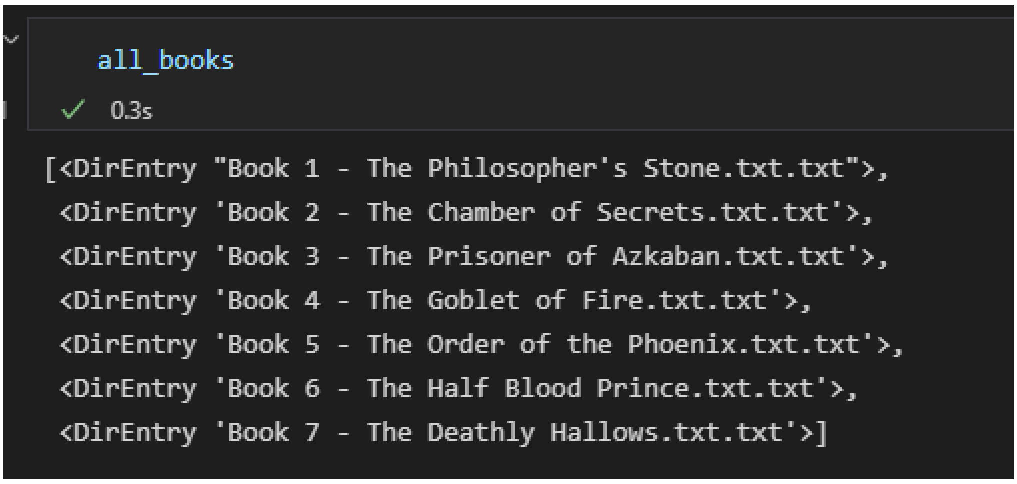
```
import pandas as pd
import numpy as np
import spacy
from spacy import displacy
import networkx as nx

import matplotlib.pyplot as plt
```

✓ 0.3s

Figure 1Imports

First I import all necessary libraries which I show in Figure 1.



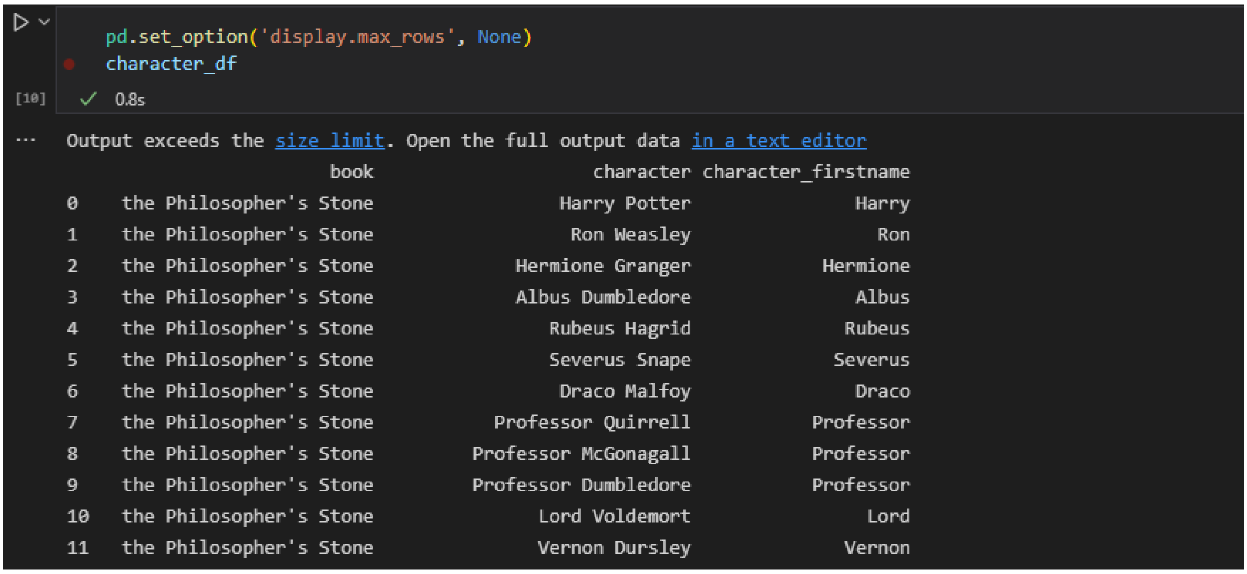
```
all_books
```

✓ 0.3s

```
[<DirEntry "Book 1 - The Philosopher's Stone.txt.txt">,
 <DirEntry 'Book 2 - The Chamber of Secrets.txt.txt'>,
 <DirEntry 'Book 3 - The Prisoner of Azkaban.txt.txt'>,
 <DirEntry 'Book 4 - The Goblet of Fire.txt.txt'>,
 <DirEntry 'Book 5 - The Order of the Phoenix.txt.txt'>,
 <DirEntry 'Book 6 - The Half Blood Prince.txt.txt'>,
 <DirEntry 'Book 7 - The Deathly Hallows.txt.txt'>]
```

Figure 2List of books

In figure 2 we can see all books and their names imported and ready to analyze.



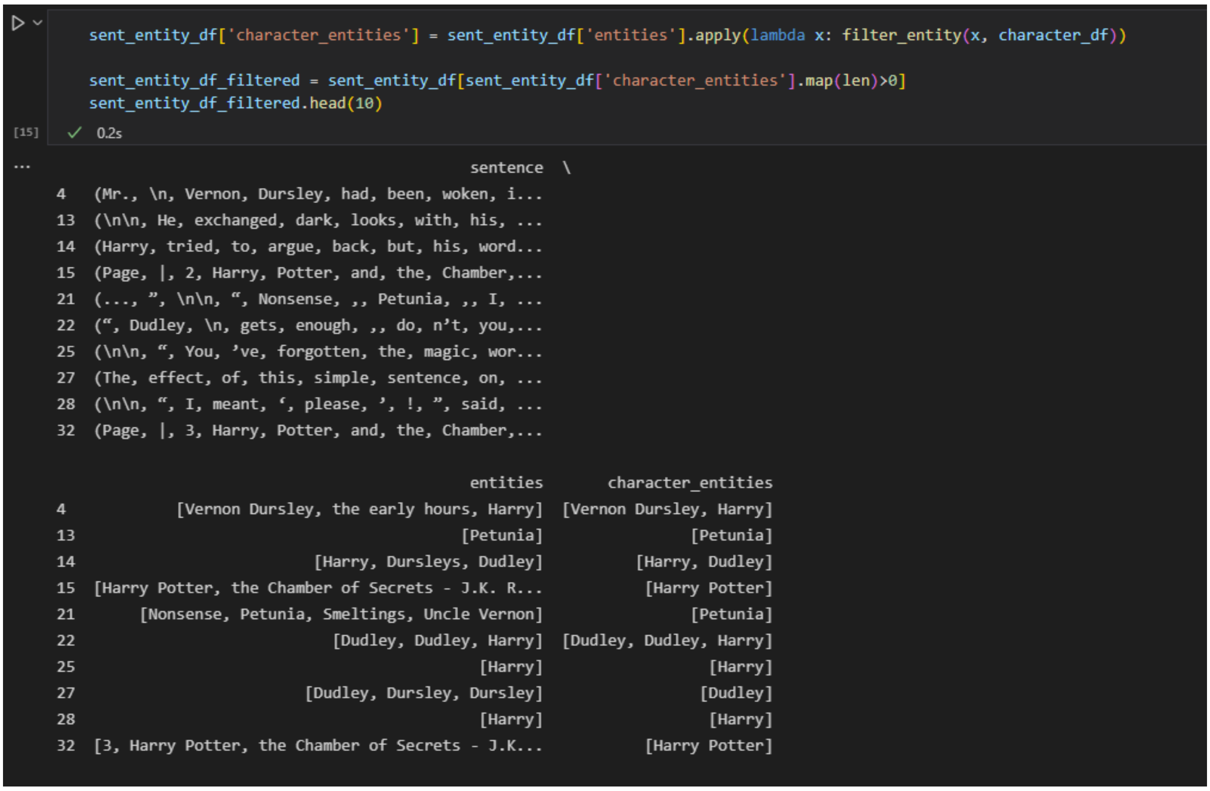
```
pd.set_option('display.max_rows', None)
character_df
```

[10] ✓ 0.8s

```
Output exceeds the size limit. Open the full output data in a text editor
                        book                character character_firstname
0    the Philosopher's Stone                 Harry Potter               Harry
1    the Philosopher's Stone                  Ron Weasley                 Ron
2    the Philosopher's Stone             Hermione Granger            Hermione
3    the Philosopher's Stone             Albus Dumbledore               Albus
4    the Philosopher's Stone                Rubeus Hagrid              Rubeus
5    the Philosopher's Stone                Severus Snape             Severus
6    the Philosopher's Stone                 Draco Malfoy               Draco
7    the Philosopher's Stone           Professor Quirrell           Professor
8    the Philosopher's Stone         Professor McGonagall           Professor
9    the Philosopher's Stone         Professor Dumbledore           Professor
10   the Philosopher's Stone               Lord Voldemort                Lord
11   the Philosopher's Stone               Vernon Dursley              Vernon
```

Figure 3 Get the first name of a character

Here from my CSV file, I separate the first name from the characters to get better results and to get a better understanding of the data.



```
sent_entity_df['character_entities'] = sent_entity_df['entities'].apply(lambda x: filter_entity(x, character_df))

sent_entity_df_filtered = sent_entity_df[sent_entity_df['character_entities'].map(len)>0]
sent_entity_df_filtered.head(10)
```

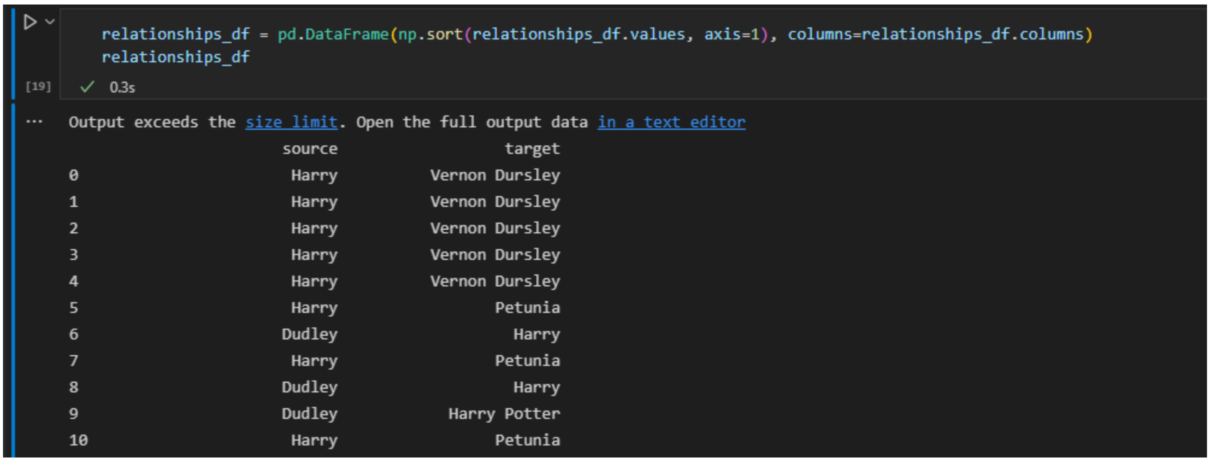
[15] ✓ 0.2s

```
                                             sentence  \
4   (Mr., \n, Vernon, Dursley, had, been, woken, i...
13  (\n\n, He, exchanged, dark, looks, with, his, ...
14  (Harry, tried, to, argue, back, but, his, word...
15  (Page, |, 2, Harry, Potter, and, the, Chamber,...
21  (..., ”, \n\n, “, Nonsense, ,, Petunia, ,, I, ...
22  (“, Dudley, \n, gets, enough, ,, do, n’t, you,...
25  (\n\n, “, You, ’ve, forgotten, the, magic, wor...
27  (The, effect, of, this, simple, sentence, on, ...
28  (\n\n, “, I, meant, ‘, please, ’, !, ”, said, ...
32  (Page, |, 3, Harry, Potter, and, the, Chamber,...

                                             entities       character_entities
4            [Vernon Dursley, the early hours, Harry]  [Vernon Dursley, Harry]
13                                          [Petunia]                [Petunia]
14                           [Harry, Dursleys, Dudley]          [Harry, Dudley]
15  [Harry Potter, the Chamber of Secrets - J.K. R...           [Harry Potter]
21       [Nonsense, Petunia, Smeltings, Uncle Vernon]                [Petunia]
22                            [Dudley, Dudley, Harry]  [Dudley, Dudley, Harry]
25                                            [Harry]                  [Harry]
27                          [Dudley, Dursley, Dursley]                 [Dudley]
28                                            [Harry]                  [Harry]
32  [3, Harry Potter, the Chamber of Secrets - J.K...           [Harry Potter]
```

Figure 4 Find all names in the sentence

Here I go through all books, get every sentence, and try to find all the names from them. After that, I look at my CSV file and see if there is that name set in my file.

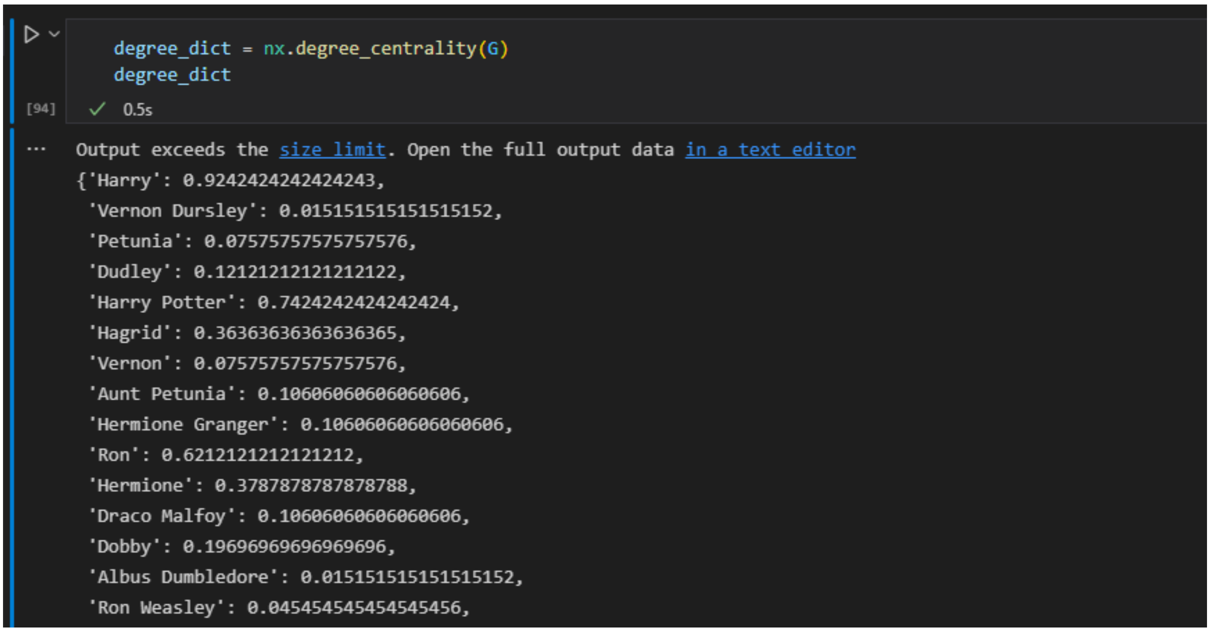


```
relationships_df = pd.DataFrame(np.sort(relationships_df.values, axis=1), columns=relationships_df.columns)
relationships_df
```

```
Output exceeds the size limit. Open the full output data in a text editor
                       source            target
0                       Harry    Vernon Dursley
1                       Harry    Vernon Dursley
2                       Harry    Vernon Dursley
3                       Harry    Vernon Dursley
4                       Harry    Vernon Dursley
5                       Harry           Petunia
6                      Dudley             Harry
7                       Harry           Petunia
8                      Dudley             Harry
9                      Dudley      Harry Potter
10                      Harry           Petunia
```

Figure 5 Get source and target

Here I get all names to find relationships from a source that has connections in any way in the book.



```
degree_dict = nx.degree_centrality(G)
degree_dict
```

```
Output exceeds the size limit. Open the full output data in a text editor
{'Harry': 0.9242424242424243,
 'Vernon Dursley': 0.015151515151515152,
 'Petunia': 0.07575757575757576,
 'Dudley': 0.12121212121212122,
 'Harry Potter': 0.7424242424242424,
 'Hagrid': 0.36363636363636365,
 'Vernon': 0.07575757575757576,
 'Aunt Petunia': 0.10606060606060606,
 'Hermione Granger': 0.10606060606060606,
 'Ron': 0.6212121212121212,
 'Hermione': 0.3787878787878788,
 'Draco Malfoy': 0.10606060606060606,
 'Dobby': 0.19696969696969696,
 'Albus Dumbledore': 0.015151515151515152,
 'Ron Weasley': 0.045454545454545456,
```

Figure 6 Degree of characters

As we can see here in Figure 6, the degree of characters is represented and it tells us how often is some names mentions in the book.

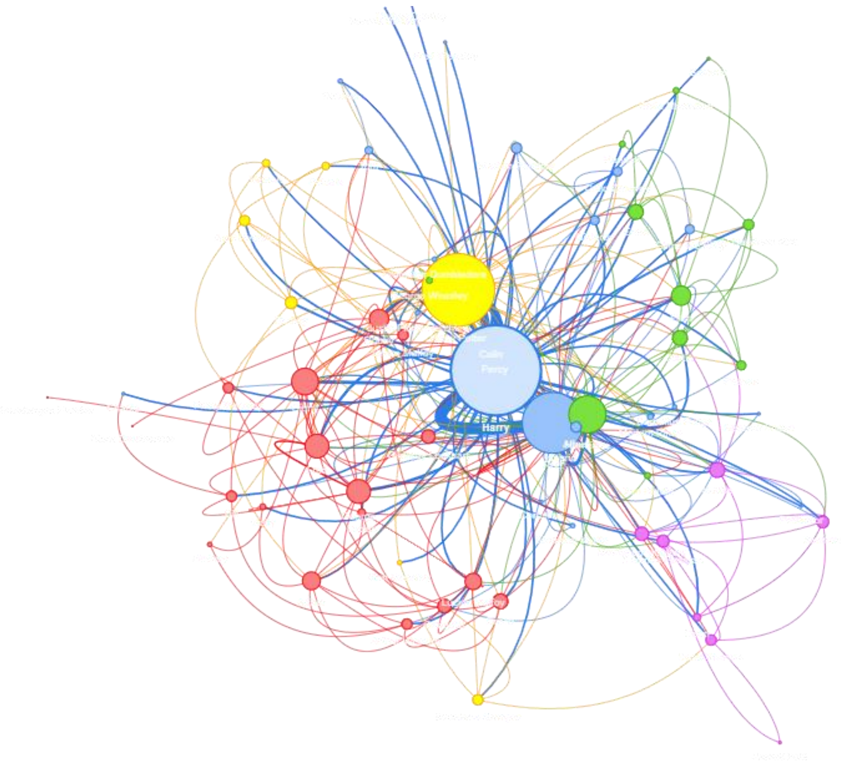
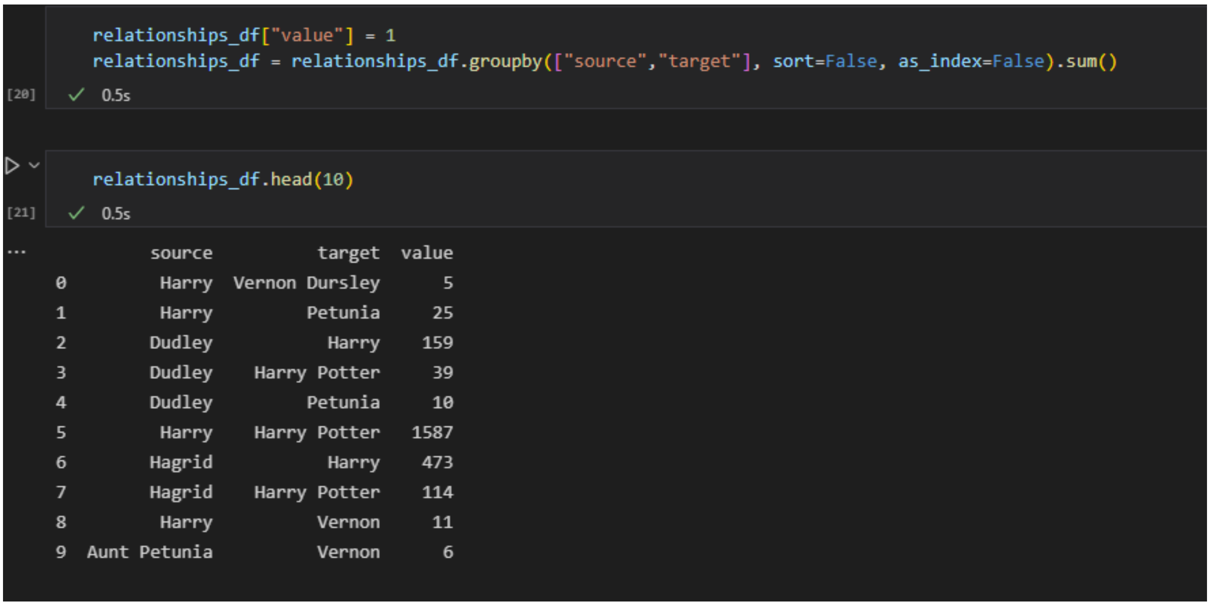


Figure 7 Graph of relationships

Here we can see a visual presentation of all the nodes and edges which are connected. Bigger nodes are more important and have more relationships in this book.



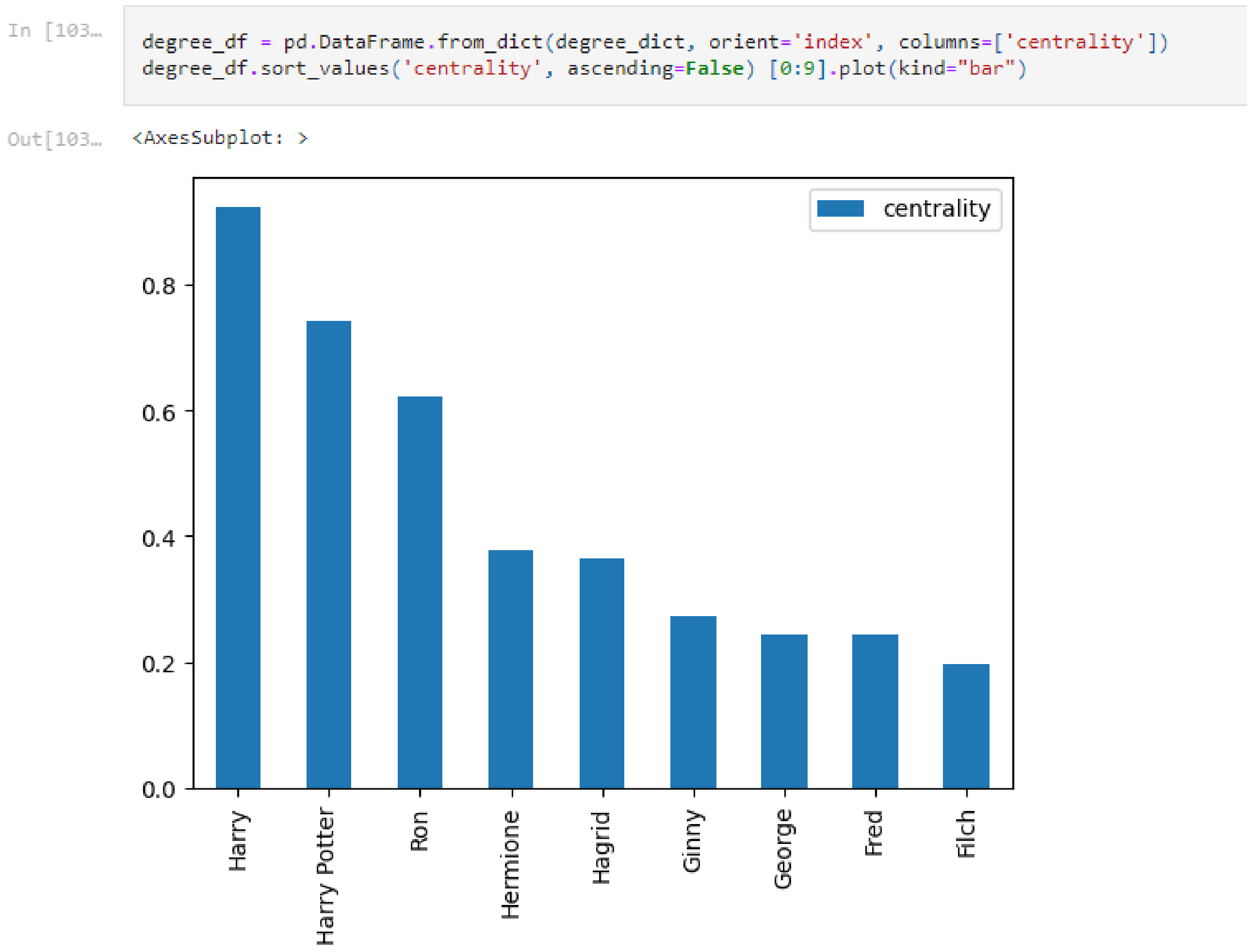
```
[20]
    relationships_df["value"] = 1
    relationships_df = relationships_df.groupby(["source","target"], sort=False, as_index=False).sum()
✓ 0.5s

[21]
    relationships_df.head(10)
✓ 0.5s

          source          target  value
0          Harry  Vernon Dursley      5
1          Harry         Petunia     25
2         Dudley           Harry    159
3         Dudley    Harry Potter     39
4         Dudley         Petunia     10
5          Harry    Harry Potter   1587
6         Hagrid           Harry    473
7         Hagrid    Harry Potter    114
8          Harry          Vernon     11
9   Aunt Petunia          Vernon      6
```

Figure 8 Relationships values

As we see in Figure 8 the values of relationships are presented, and here we can see who has the most concessions in books.

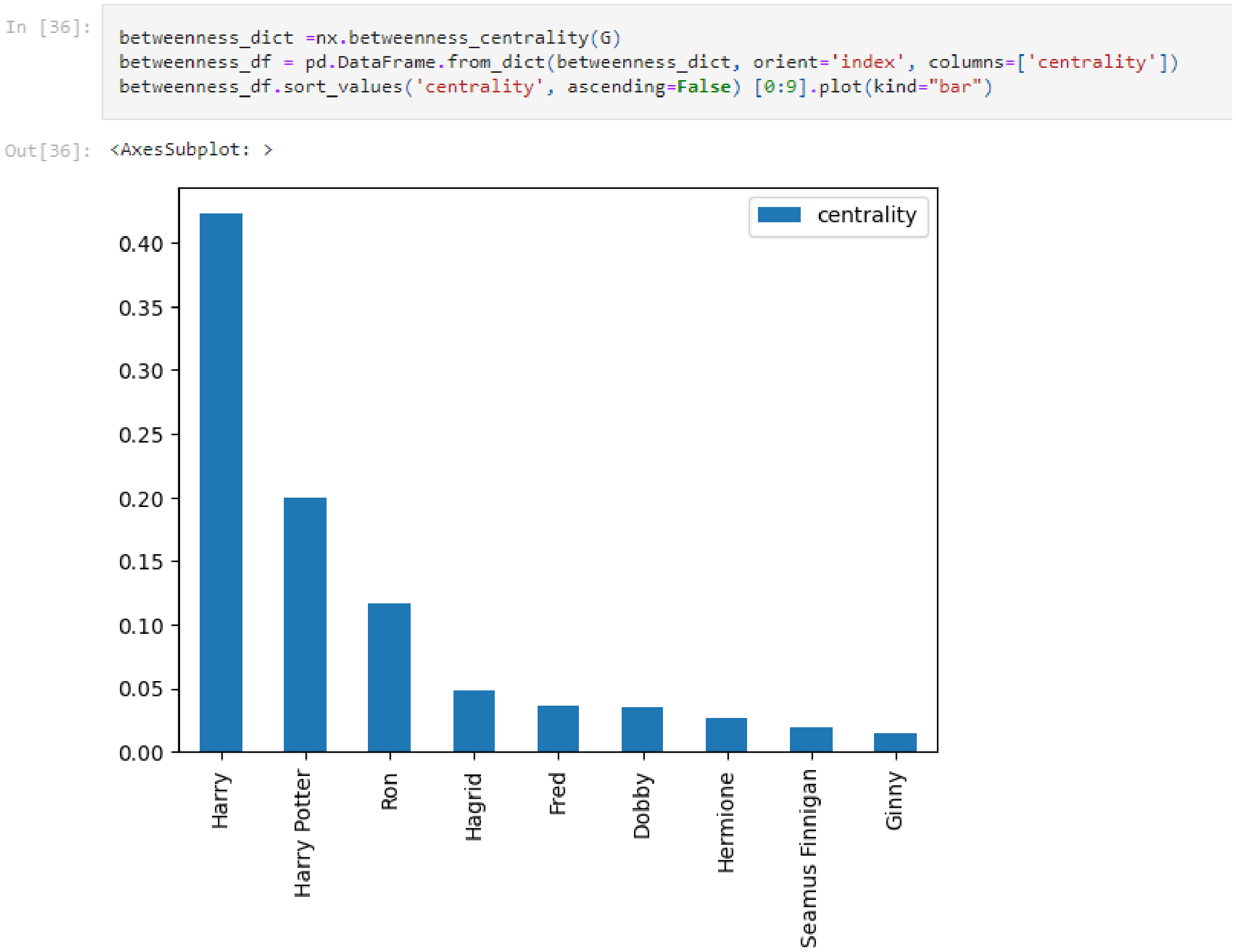


```
In [103…  degree_df = pd.DataFrame.from_dict(degree_dict, orient='index', columns=['centrality'])
          degree_df.sort_values('centrality', ascending=False) [0:9].plot(kind="bar")

Out[103…  <AxesSubplot: >
```

Figure 9 Degree bar char

Here is degree bar char base on centrality.

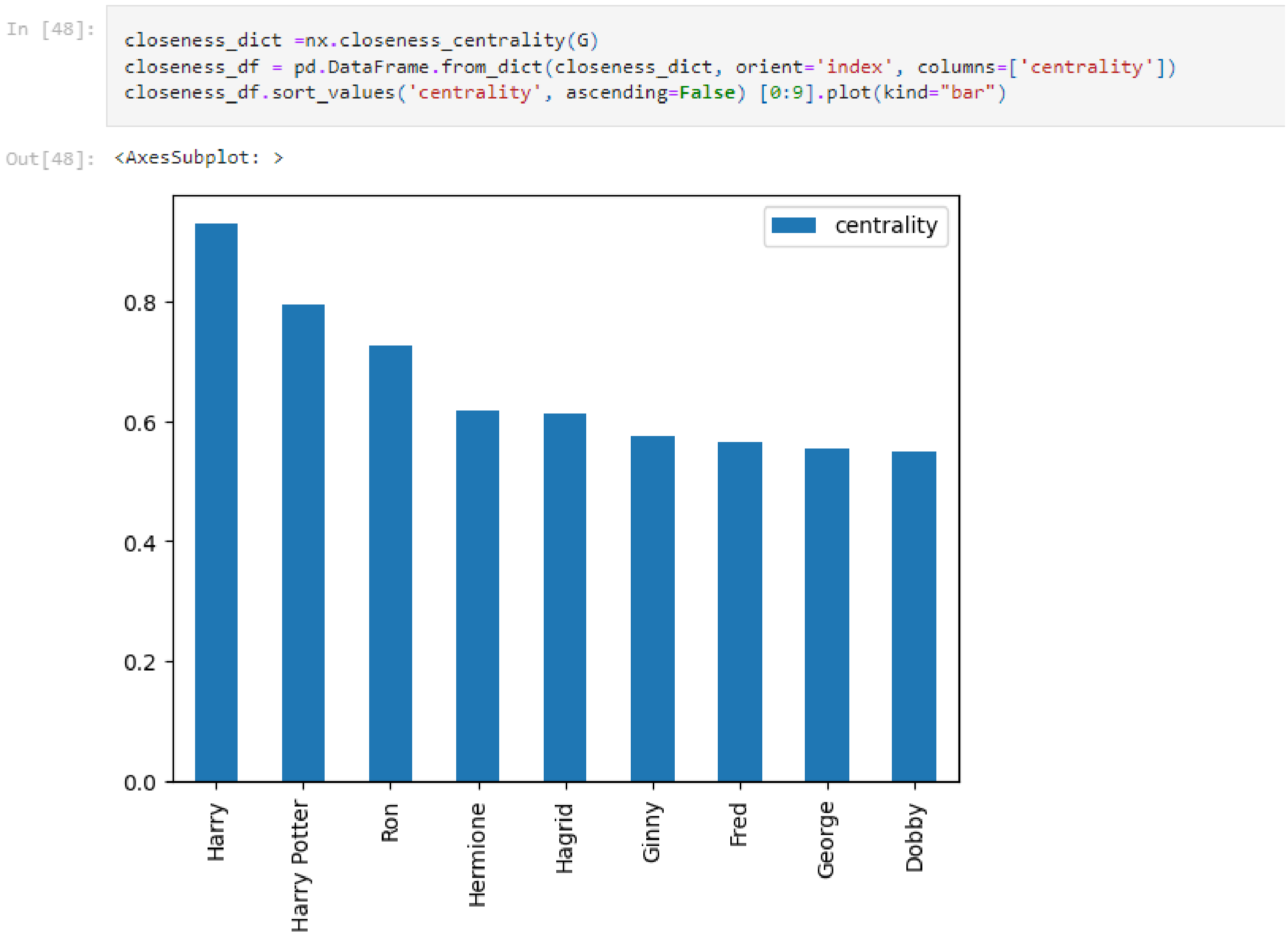


```
In [36]: betweenness_dict =nx.betweenness_centrality(G)
         betweenness_df = pd.DataFrame.from_dict(betweenness_dict, orient='index', columns=['centrality'])
         betweenness_df.sort_values('centrality', ascending=False) [0:9].plot(kind="bar")

Out[36]: <AxesSubplot: >
```

Figure 10 Betweenness bar char

Here is betweenness bar char base on centrality.



```
In [48]:  closeness_dict =nx.closeness_centrality(G)
          closeness_df = pd.DataFrame.from_dict(closeness_dict, orient='index', columns=['centrality'])
          closeness_df.sort_values('centrality', ascending=False) [0:9].plot(kind="bar")

Out[48]:  <AxesSubplot: >
```

Figure 11 Closeness bar char

Here is closeness bar char base on centrality.

Conclusion

This project was fun for me, and I enjoy working on it. SNA is a powerful tool and can be done in many ways to find a lot of pieces of information. In combination with python which I used to analyze, and graphs for the visual presentation we can see a lot of interesting relationships.

Reference

https://github.com/formcept/whiteboard/tree/master/nbviewer/notebooks/data/harrypotter

https://towardsdatascience.com/how-to-get-started-with-social-network-analysis-6d527685d374

https://pandas.pydata.org/docs/

https://numpy.org/

https://spacy.io/

https://spacy.io/models/en

https://matplotlib.org/

https://pypi.org/project/communities/

https://python-louvain.readthedocs.io/en/latest/