Data Analytics and Visualization Project Final Report: Visualizing Stories

Mohammadamin Ghasemzadeh*
York University
Toronto, Canada
Amin74@yorku.ca

Romina Abadi*
York University
Toronto, Canada
rmn@yorku.ca

"Novel, an invented prose narrative of considerable length and a certain complexity that deals imaginatively with human experience, usually through a connected sequence of events involving a group of persons in a specific setting."

Britannica Encyclopedia [4]

ABSTRACT

Characters and their interactions are the core part of all stories. This project aims to extract, visualize, and analyze the interaction network of the characters. Named Entity Recognition (NER) is used to extract character names from the text, and the co-occurrence of the character names is defined as an interaction. This report includes more detail of the character interaction network creation and analysis methods. As discussed in the report, the character interaction networks of 3200 books were analyzed and compared. This project's analyses were focused on the character interaction network size and the changes in the node importance of the main character. Examining the character interaction networks revealed patterns in books of the same authors and similar categories. Also, the analysis showed that the popularity of some author's books was related to the main character patterns. This project's results demonstrate some of the various character interaction networks' use-cases. The analysis methods proposed in this project can be expanded to other character-based written scripts, such as news. Moreover, character interaction networks can be used in automated tasks such as text summarizing.

1 INTRODUCTION

This project is about data analysis on novels. The main goal of the project is to identify the characters and find the network of their interactions. Character interaction network visualization and analyses can be used to discover new insights into the books. The following sections discuss this project's motivation and dataset in more detail.

1.1 Motivation

The main objective of this project is to build a network of the characters in a novel or any character-based script (e.g. plays, news, etc.) and to evaluate and visualize the interaction dynamics and their development through the story. Comparing networks of the characters for the novels of the same category or author allows finding differences and similarities between books. Such information allows answering questions such as: "Is there a pattern in an authors' writing?", "Are some patterns more common in some eras or categories?" and if so, "Does the pattern change in different eras?" It also can be used for a specific novel to see if the patterns are similar to or different from other books. Finally, combining relations graphs with books' popularity information makes it possible to find popular or unpopular character interaction patterns.

1.2 Dataset

A subset of Project Gutenberg is used as the dataset [16][19]. This dataset contains 3036 books from 142 authors. It has been cleaned from metadata, license information, and transcriber's notes as much as possible by the gatherer. All books are in English, in "txt" format. Books' metadata, including subjects; death and birth years of the author; and the number of downloads from Gutenberg dataset in the past month, are extracted using Gutendex[1]. We assume the number of downloads represents the popularity of the books. The books in the dataset are written between the years 1640 and 1955.

1.3 Potential Applications

The idea of analyzing the characters' network of a book to determine the storyline is new and could have many potential applications. For instance, a book recommendation system can use such information as an indicator. Also, another application that might be interesting is filtering books based on their storyline similarity to other books. Additionally, it can be determined if a book is different from most books in the same category. Characters' interaction networks can also be used in text summarising [15]. This project's method is extendable to any written text containing characters, i.e. novels, history books, and news articles involving humans and interactions. It is also extendable to scripts of movies or comic books if the characters' names are present. The only limitation is that the texts must have compatible formats. Also, analyzing character interaction networks of actual events, such as historical texts or news articles, can give new insights into the importance and impact of characters.

2 PROBLEM DEFINITION

A "character" is a Human or Human-like (e.g. "The ghost of the Christmas past" in A Christmas Carole [5]) entity in a writing. An interaction between two characters can be defined in various ways. For example, an interaction can be defined as the two characters

*Both authors contributed equally to this report.

[1] Gutendex is an open-source web API for pulling project Gutenberg's book catalogue information: http://gutendex.com/.

having a conversation (this is most relevant in works of fiction), doing something together, or simply the names of the characters appearing together in the text (co-occurrence)[15]. In this project, the co-occurrence approach is used, i.e. every time two characters' names appear closer than a fixed number of sentences, an interaction between the two characters is identified. This approach is easier to implement, and it is extensible to non-fiction books, in which conversations do not happen necessarily (i.e. historical books). The character interaction network is created, which is represented as an undirected, unweighted, dynamic multigraph. Each vertex in the interaction network represents a unique character. Each edge represents an interaction between the two characters it connects (formal definition is included in section 2.1). Once the graph is created, meaningful information is extracted by performing various analyses on the networks and comparing the networks. Details of this project's analyses can be found in section 4.2.

2.1 Formal Definition of Character Interaction Networks

Character Interaction Network is represented as an undirected, unweighted, dynamic multigraph G. Each vertex v represents a unique character that has appeared at least once in the text. Different vertices v_{c1} and v_{c2}, must represent different characters, i.e. if $v_c1 \neq v_c2$ then $c1$ and $c2$ are different characters. For each interaction between any two characters, a new edge is created between the two corresponding nodes. Each edge represents exactly one interaction, so the edges do not have weight. Additionally, edges have an attribute "time" (timestamp). The edge's timestamp shows when the interaction happens in the book. Timestamps let us create snapshots of the graph, which allows analyzing the graph throughout the book's story. The formal definition of the character interaction network is as follows:

$G(V, E)$: The graph representing the network

$v \in V$: a unique character

$\{u, v, i\}_t \in E$: an interaction between characters u and v at time t

In the above definition, the index i indicates that more than one interaction can exist between two characters at one time, which is discussed in more detail in section 4.1.2. A snapshot of G at time interval t_1 to t_2 is a graph that has the same nodes as G and only edges with time stamps between t_1 and t_2; formally:

$$Snapshot_{t_1,t_2}\big(G(V,E)\big) = G'(V,E')$$
$$E' \subseteq E$$
$$E' = \{e \in E | e.time \in [t_1, t_2)\}$$

3 RELATED WORK

Creating character interaction networks from texts can be viewed from technical and application perspectives. A complete survey on character network extraction in works of fiction (including texts and movies) is provided in [15]. The survey summarizes applications of such networks, including literature analysis, pedagogical purposes, and usage in the Artificial Intelligence area for tasks such as summarizing texts. As an example of using character interaction networks for pedagogical purposes, [2] visualizes characters in Shakespeare plays. However, from a technical perspective, this work is different from ours because the structure of plays is different from novels and history books which is the focus of this project. From the technical view, automatically extracting character names from raw text lies within the Natural Language Processing line of research. We used implementations of machine learning methods in [17] and [13].

4 METHODOLOGY

For the analyses in this project, a character interaction network (defined in section 2.1) was created for each book. The graph creation consists of three main parts, 1. extracting character names from raw text, 2. processing input text and creating raw character interaction graph, and 3. merging the similar character vertices in the graph. The main parts are discussed separately in sections 4.1.1, 4.1.2, and 4.1.3. For each part, possible improvements are included. Moreover, the analysis methods implemented are discussed in 4.2.

4.1 Character Interaction Network Creation

4.1.1 Extracting Character Names From The Text. First, to create the network, a method to extract characters' names from the raw text was needed. Named Entity Recognition(NER) was used to extract characters' names. There are various libraries in python that provide NER. Three packages were compared, and the Stanford CoreNLP[17] Named Entity Recognizer[13] was chosen for its accuracy. The details of evaluations are provided in section B. Stanford Core NLP is implemented in Java, and it is accessible in python via its server (StanfordCoreNLPServer). For interacting with the server, the Natural Languages Toolkit (NLTK) library[3]'s interface to the Stanford CoreNLP server was used.

For identifying the character names from the text, the raw text was first turned into tokens, using Stanford CoreNLP Tokenizer. The tokens are words or punctuation marks. For example, for the sentence:

"Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much.",
Harry Potter and the Philosopher's Stone[18]

the tokens are:

(Mr.), (and), (Mrs.), (Dursley), (,), (of), (number), (four), (,), (Privet), (Drive), (,), (were), (proud), (to), (say), (that), (they), (were), (perfectly), (normal), (,), (thank), (you), (very), (much), (.).

Then, the tokens were passed to the NER tagger, which adds tags to each token separately. The tags are: 'PERSON,' 'NUMBER,' 'ORGANIZATION,' 'LOCATION,' 'STATE_OR_PROVINCE,' and 'O' for the tokens that are not named entities. The tagged tokens for the above-mentioned sentence are:

('Mr.', 'O'), ('and', 'O'), ('Mrs.', 'O'), ('Dursley', 'PERSON'), (',', 'O'), ('of', 'O'), ('number', 'O'), ('four', 'NUMBER'), (',', 'O'), ('Privet', 'LOCATION'), ('Drive', 'LOCATION'), (',', 'O'), ('were', 'O'), ('proud', 'O'), ('to', 'O'), ('say', 'O'), ('that', 'O'), ('they', 'O'), ('were', 'O'), ('perfectly', 'O'), ('normal', 'O'), (',', 'O'), ('thank', 'O'), ('you', 'O'), ('very', 'O'), ('much', 'O'), ('.', 'O').

For identifying the character's names, all consecutive tokens with the 'PERSON' tag were added together as one name in this project. For example, the occurrences of "Ebenezer Scrooge" (the main character's name in *A Christmas Carol*[5]), were tokenized and tagged

as ('Ebenezer', 'Person'), ('Scrooge', 'Person'). The two words were added together to have one character name "Ebenezer Scrooge". Different forms of the same character's names are detected as different characters in the character name extraction process. For example, the character "Ebenezer Scrooge" appears in the text as "Scrooge", "Ebenezer Scrooge", "Mr. Scrooge", and "Uncle Scrooge", which are all detected as different characters in this stage.

Shortcomings of Character Name Extraction: The Named Entity Recognizer is trained to identify "person" names, mostly limited to identifying human names. However, some fictional characters are not necessarily humans; they can be human-like characters such as "Ghost of Christmas Present" and "Ghost of Christmas Past" in A Christmas Carol[5]. This project's method fails to identify these characters. Also, the employed approach ignores pronouns. In most cases, this is not troubling because the character's name appears before the pronoun; however, in novels with the first-person narrator, the narrator's name is rarely mentioned. In this case, the employed approach fails to detect the occurrences of the first-person narrator. Ignoring the pronouns decreases accuracy significantly in character interaction networks of texts where pronouns rather than their names refer to the characters. As a solution, advanced NLP methods must be incorporated to find the pronouns' references in the text.

4.1.2 Creating A Basic Interaction Graph. We defined interaction as any co-occurrence of two names. In other words, if two names appeared closer than k lines together, an interaction was identified (k is set to 10 in our analyses). In the used dataset's books, each line consists of around 10 words, and the lines are not consistent with the sentences. The detailed algorithm is provided in algorithm 1. In step 3 of the algorithm, NER is used to extract character names. The repetition of similar names was not removed, so if a name appeared more than one time, more than one edge was created between that name and other names; this, in a sense, added more weight to the interaction.

Algorithm 1: Interaction Network Creation Algorithm

```
1  Let G(V,E) = empty graph, t = 1, previous-names = empty list
2  foreach chunk (k/2 lines) in the text do
3      names-list = all names in the chunk
4      foreach name in names-list do
5          if name ∉ V then
6              add name to V
7      foreach n1 and n2 in names-list do
8          if (n1 ≠ n2) then
9              add one edge between n1 and n2 with time=t
10     foreach n in names-list and p in previous-names do
11         if (n ≠ p) then
12             add one edge between n and p with time=t-1
13     update previous-names = names-list
14     update t += 1
```

Possible Improvements for Creating the Interaction Graph Alternative approaches can be used for identifying interactions. Using more complex NLP methods allows including information about the types of interactions, such as doing something together, talking to each other, etc. Moreover, analyzing each chunk's overall sentiments allows adding sentiment information to the interactions, allowing further analysis of the storylines.

4.1.3 Merging Similar Nodes. In the basic interaction graph, different appearances of the same character's name were identified as different characters. In this stage, the nodes in the graph that represent the same character were merged. A three-stage merge method was applied to the vertices: 1. group the possibly similar nodes together, 2. divide each group into non-conflicting subgroups, and 3. merge each non-conflicting group of nodes. In stage 1, an empty copy ($G'(V, E)$) of the basic graph ($G(V, E)$) was created that has the same vertices as G and has no edge. Similarity conditions (discussed in section 4.1.3) were used to add an edge in G' between every two nodes that possibly represent the same character (e.g."Rose Maylie", "Maylie"). At this stage, each connected component of G' consists of groups of possibly similar characters. However, some conflicts might appear; for example, "Rose Maylie" and "Harry Maylie" are connected through "Maylie". Conflicting conditions (discussed in section 4.1.3) were used to find conflicts. Then, for every two nodes u and v in G', which were connected and had a conflict, the Min-Cut algorithm was used to remove minimum edges that resolved the conflict (disconnected the two conflicting nodes). Conflicts were removed for each connected component in G' until no conflict existed. Then, in stage 3, all connected groups of nodes in G' were merged in G. Merging the nodes preserves all the edges and edge data. A possible improvement is to remove minimum edges that resolve all conflicts in stage 2 (instead of taking into account only one conflict at a time).

Similarity and Conflicting Conditions

Similarity conditions for two names determine if the names represent the same person. The primary requirement to check is if the first or the last names are equal. However, sometimes, conflicting names are grouped (e.g. similar names with different family names are grouped if the name appears separately in the text). Conflicting conditions were used to remove such inconsistencies. The details of the conditions are as follows.

Similarity Conditions First, names' parts are separated in titles, first name, and last name sections by Python's "HumanParser" library. Then, all are written in lower case. The conditions of similarity are as follows (in order of execution):

- If the names have no words in common, they are not similar.
- If first names or last names are present in both names and are not similar, the names are not similar(e.g. "Bob Marley" and "Peter," or "Bob Marley" and "Mr. Cratchit").
- If the titles have different genders, the names are not similar (e.g. "Mr. Marley," "Mrs. Marley").
- If non of the above conditions hold, the names are similar.

Conflicting Conditions First, names are prepared like the similarity section. Then, two names are considered to have a conflict if first names or last names are different (e.g. "Bob Marley," "Peter Marley"), or they have different genders' titles (e.g. "Mr. Marley," "Mrs. Marley").

4.2 Character Interaction Network Analyses

After creating a graph of the interactions, the following analysis were used:

4.2.1 Important Characters. Important characters were extracted from the text by sorting the characters based on their Page-Rank in the character interaction network (In the analyzed books, the character with the highest page-rank was the main character of the story). Central characters of each writing are the minimum number of characters with the highest Page-Rank whose total page-rank is more than 0.5.

4.2.2 Character Importance Change Through Time. Character importance through time was plotted by finding k characters with the highest Page-Rank scores and plotting their Page-Rank in consecutive snapshots of the story's timeline. The change in Page-Rank of the k most central characters was plotted and qualitatively evaluated in different writings.

4.2.3 Main Character Similarity Graph. For comparing the books based on the changing pattern in the importance of the main character, a graph (main character similarity graph) was created in which nodes are books, and edges indicate the main character importance pattern is similar between the two books. Fifteen snapshots were used for all books to evaluate the similarity of two books' main character importance through time. Then, the ratio of the page-rank to the maximum page-rank in each snapshot was calculated. These numbers were converted to 5 levels. (High importance > 0.8, 0.8 > medium-high > 0.6, 0.6 > medium > 0.4, 0.4 > medium-low > 0.2, and 0.2 > low). Finally, books were compared to each other based on these levels, and a graph of similarity was made. If two books were similar to each other more than 80%, an edge was drawn between them. The accuracy of this method is a function of the number of snapshots, levels, and threshold percentages for the similarity of two books. These parameters (15 snapshots, five levels, and 80%) were selected based on how a human compares two plots. For example, plots for two books of the Sherlock Holmes series (A) look similar for a human. By using 15 snapshots and five levels, the method indicates that they are 83% similar). The Label Propagation algorithm was used to find communities in the main character similarity graph to find different main character importance patterns.

4.3 Single Category for Each Book

As mentioned before, each book in the used dataset has multiple subjects. Some of our analyses required only one category for each book (e.g. 5.2.1). For assigning one category to each book, a category graph was created where nodes are books, and each edge connects two books with at least one subject in common. Then, communities in the resulting graph were detected using the Label Propagation algorithm, and each book's community index was used as its unique category. In total, 68 categories were assigned to the books. Thirteen out of the 68 categories included more than 30 books (The number of books in each category is plotted in figure 14). Books with the same category are about related subjects. However, because each category includes many subjects and subjects appear in multiple categories, it is hard to assign one name to each category community.

Table 1: Statistic information for the number of characters (total), number of central characters (central), and their ratio.

	mean ± σ	q1	mode	q3
total	145.22 ± 177.27	42	91	179
central	20.66 ± 29.32	5	9	24
ratio	0.17 ± 0.12	0.08	0.15	0.24

5 EVALUATION AND RESULTS

In this section the results of the analyses and comparison of different books are provided. The results of analysing single books are included in appendix A.

5.1 Number of Characters

The number of all characters, central characters and the ratio of the number of central characters to the number of characters were calculated for 2905 books with different subject labels. Figure 1 and table 1 summarize the statistics of the three values. The statistics showed that the average number of characters in the books was 177. However, on average, each book had 20 main characters, which was more than our expectation. The plots also indicated that on average, 17% of the characters contained 0.5 of page-rank (i.e. are central to the stories). Furthermore, it was observed that most of the graph size ratio outliers were short stories. No pattern was found in graph size and central graph size outliers.

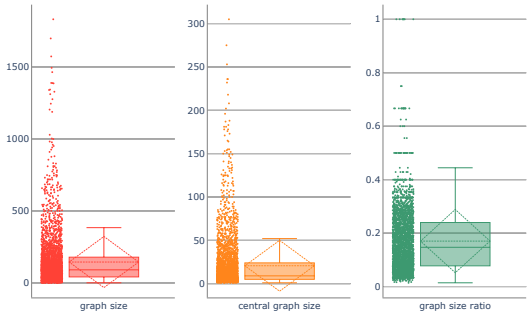

Figure 1: Graph size, central graph size, and their ratio. Mean and variance are annotated with dashed lines.

The size of the graph was further investigated for various subjects (figure 2). It was observed that the size of the graph was affected by the subject. For example, love stories tended to have a relatively small central graph, which means the story is about a limited number of characters. Having a small central graph was also true for "Bildungsromans" books, which depict one character through their life ('Jane Eyre", "David Copperfield", and "Oliver Twist" are examples in our dataset). On the other hand, it was

discovered that short stories did not have many characters, but a relatively large number of characters played an important role in the stories. One unexpected observation was that biography and history books approximately had the same number of characters. However, the analyzed history books had fewer central characters compared to biography books. This observation might be because our dataset includes books about specific historical characters. No correlation was observed between the number of characters and the popularity of subjects (figure 13 in C visualizes graph size and popularity for various subjects).

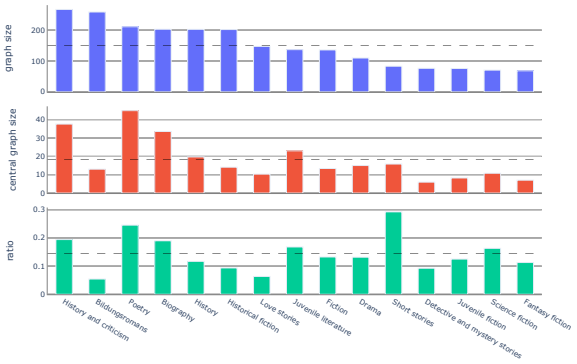

Figure 2: The mean values for graph size, central graph size, and ratio for various subjects. The horizontal dashed lines represent the mean for all books.

Additionally, the number of characters was analyzed in various authors' writings. The authors that had more than 40 books in the project's dataset were selected for the analysis. As shown in figure 3, some authors' works were more diverse in terms of the number of characters (e.g. Charles Dickens and Anthony Trollope) and the ratio of the characters that are important (e.g. Nathaniel Hawthorne). In contrast, other authors used similar patterns in their works. Also, some authors (e.g. Charles Dickens) used a higher number of characters in their books; in contrast, other authors used fewer characters on average (i.e. Henry James). No relation was found between the popularity of the authors with the number of characters they use in their novels.

5.2 Main Character Similarity Graph Analysis

As mentioned in 4.2.3, main character similarity graph was made in which the books with similar main character importance patterns were connected. Two main communities (>1000 nodes) were detected in the main character similarity graph's largest connected component (figure 4). After evaluating many books in each community, it was realized that red nodes were the books with a storyline of one main character with many occurrences in most parts of the book. Further analyses showed that these books were about one main character (single). On the other side, blue area nodes were books with multiple main characters (multiple), in which the one with the maximum page-rank only appears in some parts of the

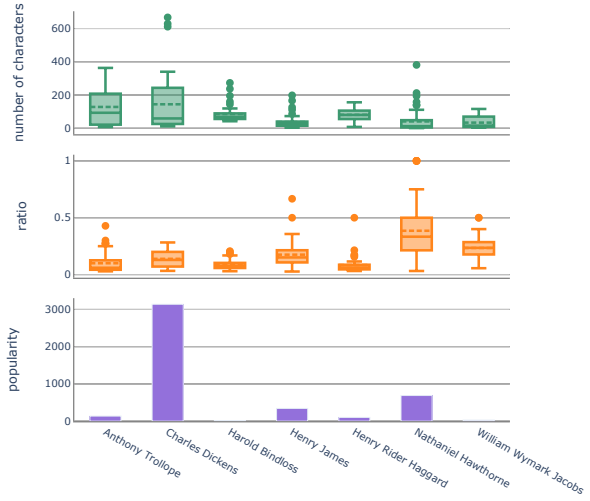

Figure 3: Box plots for graph size, central graph size, and ratio for various authors.

book. Figure 5 confirms this hypothesis, as the average size of the graph for the "multiple" community is bigger. In this section, we discuss patterns found in the main character similarity graph.

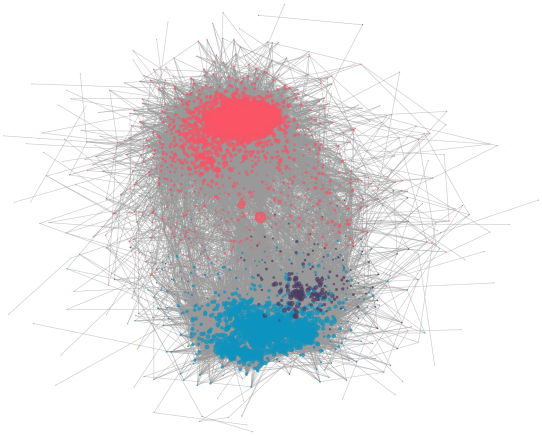

Figure 4: Communities of the main character similarity graph's largest connected component

5.2.1 Category and Main Character Similarity Graph. In figure 6 nodes are coloured based on their category community in the main character similarity graph. It can be seen that each category appears mostly on one side of the graph (figure 15 in appendix C shows the frequency of the categories in the two main communities of the

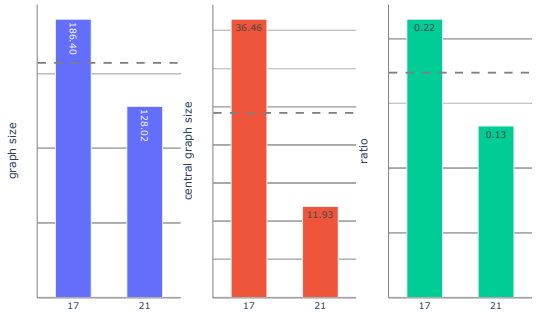

Figure 5: The mean values for graph size, central graph size, and ratio for the main character similarity graph's major communities. 21 represents the "single" community, and 17 represents the "multiple" community. The horizontal dashed lines represent the mean for all books.

Figure 6: Main character similarity graph's largest connected component with nodes colored based on their category community

main character similarity graph). Analyzing individual categories showed that for some categories (about 2/3 of all the categories), books mainly belong to either "multiple" or "single" main character pattern. Moreover, the main character patterns are similar to each other (they are connected in the main character similarity graph). For instance, figure 7 shows a category primarily including juvenile fiction books. The majority of the books are based on one main character and are connected, which means their main character patterns are similar.

5.2.2 Authors and Main Character Similarity Graph. Plotting the graph of an author's books on the main character similarity graph

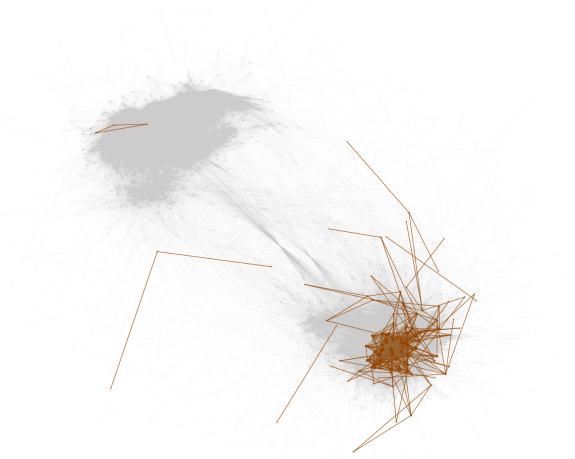

Figure 7: Juvenile fiction books marked brown on the main character similarity graph's largest connected component

showed that 67 out of 127 authors (53%) use similar story-line patterns for the books' main character (all of their books were connected in the main character similarity graph). Also, it was revealed that some authors' books mainly belong to one of the "single" and "multiple" communities of the main character similarity graph. For example, figure 8 shows Edward Stratemeyer's books. He was an American writer in the 19^{th} and 20^{th} centuries, and most of his books are in the category of children's fiction. As the figure shows, his books are in the group of "single" main characters and are mostly connected. Figure 16 in appendix C summarizes the ratio of the books of each author that belong to one of the two largest communities of the main character similarity graph.

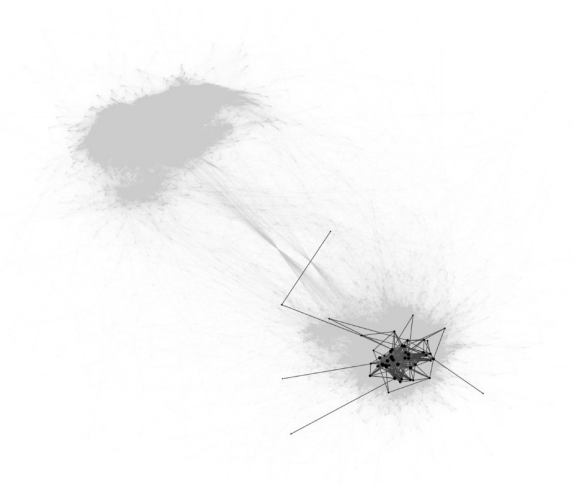

Figure 8: Edward Stratemeyer's books marked black on the main character similarity graph's largest connected component

Visualizing the popularity of the books as node sizes on the main character similarity graph uncovered different patterns for authors.

For instance, as shown in figure 9, for some authors like Robert Louis Stevenson, deviating from most books with similar patterns made a book famous (large nodes). On the other hand, for other authors, like Daniel Defoe, central books in the network of similar books were more popular than the isolated nodes.

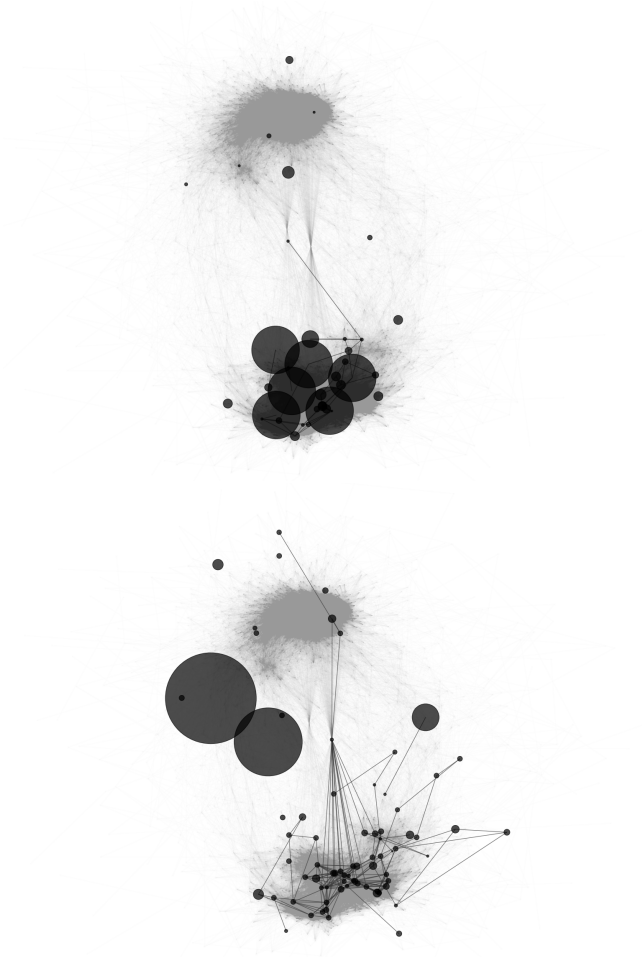

Figure 9: Books of Daniel Defoe (upper) and Robert Louis Stevenson (lower) marked black on the main character similarity graph. The node sizes show popularity.

5.3 Analyzing The Centrality of Historical Characters

Analyzing individual character's importance in the character interaction networks of historical events or news articles (primarily political articles) can reveal information about the importance or centrality of individuals involved. As an example, figure 10 shows the importance of Darius III (the king of Persia) in "The Biography of Alexander the Great" (by Jacob Abbott)[1]. The increase in the centrality of Darius shows his importance in Alexander's life, and this corresponds to the historical fact that Alexander defeated Darius III.

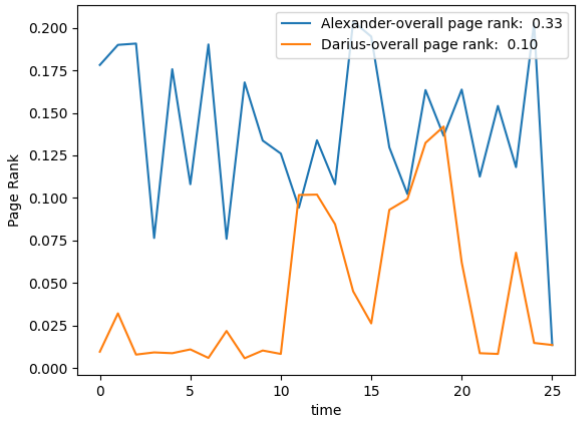

Figure 10: Importance of Alexander the Great and Darius III in "The Biography of Alexander the Great"

6 CONCLUSIONS

In this project, we discussed the process and challenges of creating character interaction networks. We also discussed our method's shortcomings and possible improvements. Moreover, we showed the character interaction networks analysis could reveal interesting patterns in the authors' narration styles or in books with different subjects. Also, we provided an example of how analyzing the character interaction network of historical books can reveal new insights into the importance of individuals in historical events. This project used a simplistic approach for detecting interactions. As a future direction, advanced NLP methods can be used to determine the type of interactions and each interaction's sentiment. Moreover, we only analyzed the importance of the main character. Our analysis method can expand to evaluating the centrality of other characters or the importance of interactions.

REFERENCES

[1] Jacob Abbott. 1803-1879. *Alexander the Great; Makers of History.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[2] Xanthos Aris, Pante Isaac, Rochat Yannick, and Grandjean Martin. 2014. Visualising the Dynamics of Character Networks. In *In Digital Humanities 2016: Conference Abstracts.* Jagiellonian University and Pedagogical University, Kraków, 417–419. https://dh2016.adho.org/abstracts/407

[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.". http://nltk.org/book.

[4] Anthony Burgess. 2020. "Novel". https://www.britannica.com/art/novel

[5] Charles Dickens. 1843. *A Christmas Carol.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[6] Sir Arthur Conan Doyle. [n.d.]. *Beyond the City.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[7] Sir Arthur Conan Doyle. [n.d.]. *A Desert Drama.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[8] Sir Arthur Conan Doyle. [n.d.]. *A Duet with an Occasional Chorus.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[9] Sir Arthur Conan Doyle. [n.d.]. *The Hound of the Baskervilles.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[10] Sir Arthur Conan Doyle. [n.d.]. *The Sign of the Four.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[11] Sir Arthur Conan Doyle. [n.d.]. *The Valley of Fear.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

[12] Sir Arthur Conan Doyle. 1887. *A Study In Scarlet.* Retrieved January 30, 2021 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html
[13] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (Ann Arbor, Michigan) *(ACL '05)*. Association for Computational Linguistics, USA, 363–370. https://doi.org/10.3115/1219840.1219885
[14] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python.* https://doi.org/10.5281/zenodo.1212303
[15] Vincent Labatut and Xavier Bost. 2019. Extraction and Analysis of Fictional Character Networks: A Survey. *ACM Comput. Surv.* 52, 5, Article 89 (Sept. 2019), 40 pages. https://doi.org/10.1145/3344548
[16] Shibamouli Lahiri. 2014. *Gutenberg Dataset.* Retrieved Mar 1, 2020 from https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html
[17] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010
[18] J. K. Rowling. 1997. *Harry Potter and the Philosopher's Stone* (1 ed.). Vol. 1. Bloomsbury Publishing, London.
[19] Lahiri Shibamouli. 2014. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, 96–105. http://www.aclweb.org/anthology/E14-3011

A SINGLE BOOK ANALYSIS

As an example of character interaction network usage for literary analysis of books, we summarize our analysis results on four books from the Sherlock Holmes series: "A Study in Scarlte"[12], "The Valley of Fear"[11], "The Hound of the Baskerville"[9], and "The Sign of the Four"[10]. Figure 11 shows the character interaction network for "A Study in Scarlte".

Importance (Page-Rank) through the time is plotted for the main character of the four books. Each book is divided into 40 sections. As shown in figure 12 for "A Study in Scarlet" novel, Sherlock Holmes is involved in the first sections of the story. Then, his participation was lower in the middle (mainly did not participate). He again appeared in the final sections of the story. This narration pattern also applies to "The valley of fear" and "The Hound of the Baskervilles." But in the "The Sign of the Four" novel, there is a different narration pattern for the main character. The appearance of Sherlock increases till the middle of the story; then, it decreases slowly.

B EVALUATION OF NAMED ENTITY RECOGNITION AND THE CHARACTER INTERACTION GRAPH

To evaluate three NER taggers (NLTK[3], spaCy[14], and Stanford CoreNLP[17]), a chapter of "A desert drama"[7], "A duet"[8], and "Beyond the city"[6] novels by Sir Arthur Conan Doyle were used. For each book, one team member identified all characters and plotted a graph of the interactions. Taggers were evaluated based on the number of correctly identified characters, the total number of occurrences of characters in the text, and the number of total interactions. The evaluation results are summarized in table 2. CoreNLP was selected since it has the best results among the NER taggers. The number of wrong identified characters is not reported in the table. This number was considerable for the NLTK tagger, but the numbers were low for the spaCy and CoreNLP.

For evaluating the effectiveness of the similarity and merge methods, "A Christmas Carol"[5] novel by Charles Dickens was selected. The number of each character occurrences in the text was the parameter of the evaluation. In table 3, the results for three main characters are presented. As presented in the table, using similarity and merge methods improve the accuracy of CoreNLP.

C ADDITIONAL GRAPHS

This section includes additional graphs used for analyses.

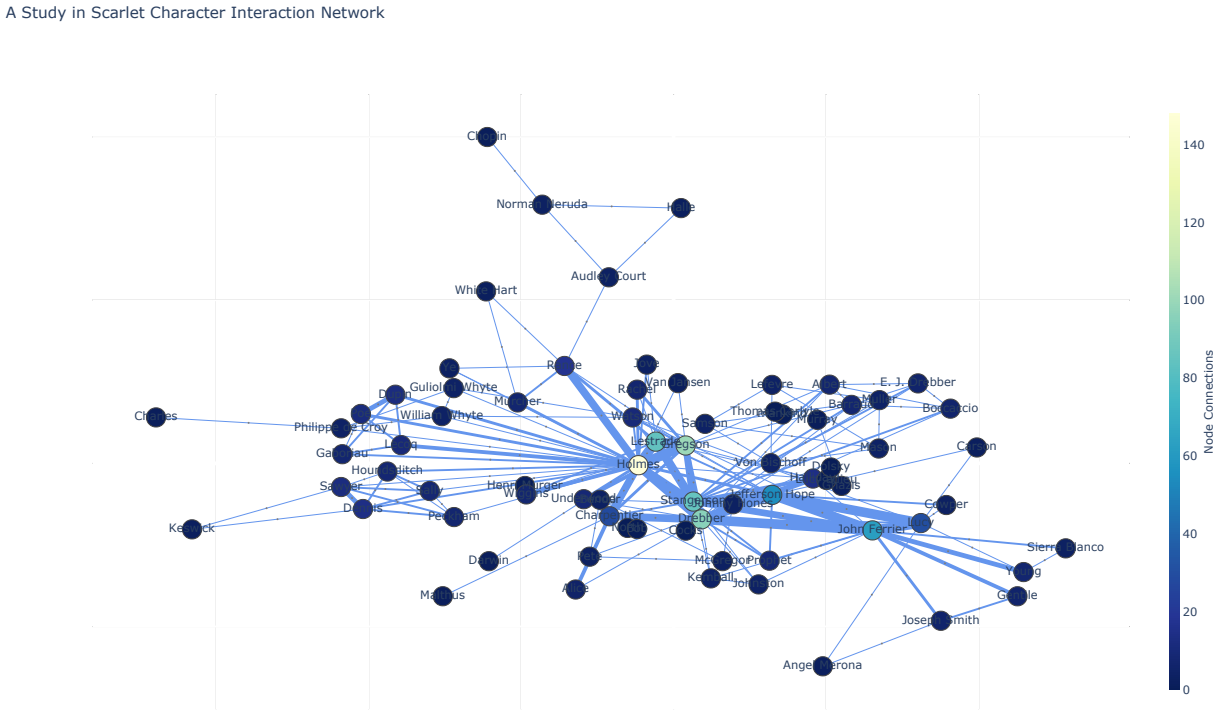

Figure 11: The character interaction network for A Study in Scarlet[12]

Table 2: Results of NER methods comparison in "A Desert Drama", "A Duet", and "Beyond the City" from left to right.

	Truth	NLTK	Spacy	CoreNLP
# Identified Characters	10-6-2	7-6-2	9-6-2	10-6-2
# Character Names Occurrences	23-22-9	19-18-9	20-18-9	20-18-9
# Total Interactions	12-16-10	7-11-5	10-13-10	11-13-10

Table 3: The number of character name occurrences before and after merge in "A Christmas Carol".

	Truth	Simple CoreNLP	CoreNLP after Merge
# "Ebenezer Scrooge"	375	278	283
# "Jacob Marley"	52	27	46
# "Bob Cratchit"	53	43	51

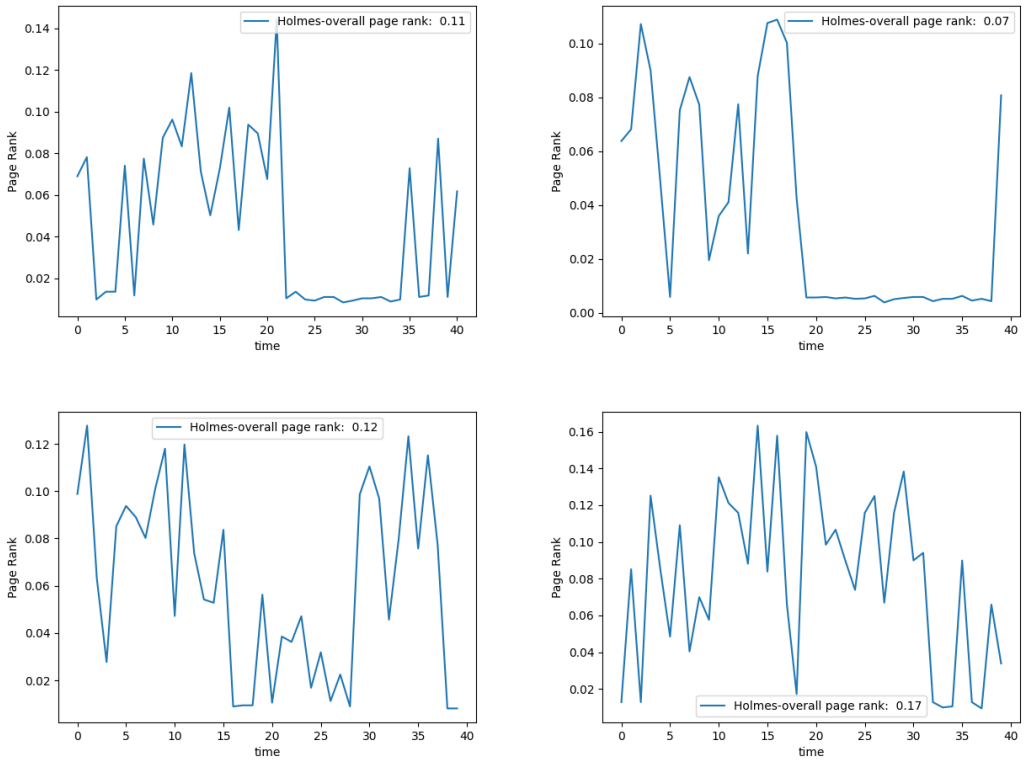

Figure 12: Sherlock Holmes's Page-Rank by time in *A Study in Scarlet*[12] (top-left), *The Valley of Fear*[11] (top-right), *The Hound of the Baskervilles*[9] (bottom-left), and *The Sign of The Four*[10] (bottom-right)

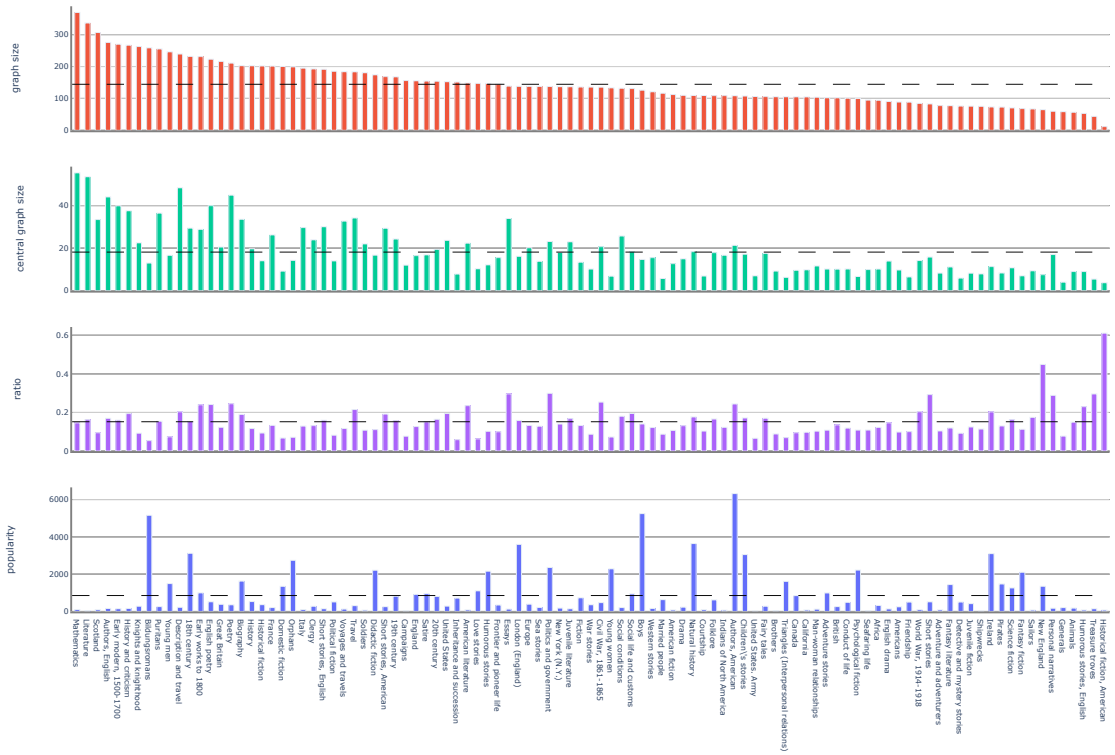

Figure 13: The mean values for graph size, central graph size, ratio, and popularity for various subjects. The horizontal dashed lines represent the mean over all books.

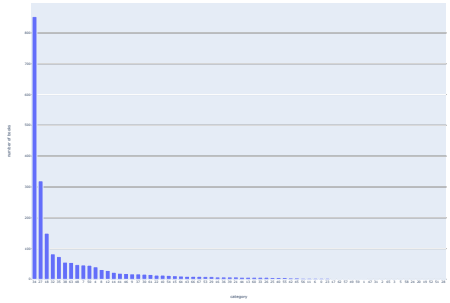

Figure 14: The number of books in each category community.

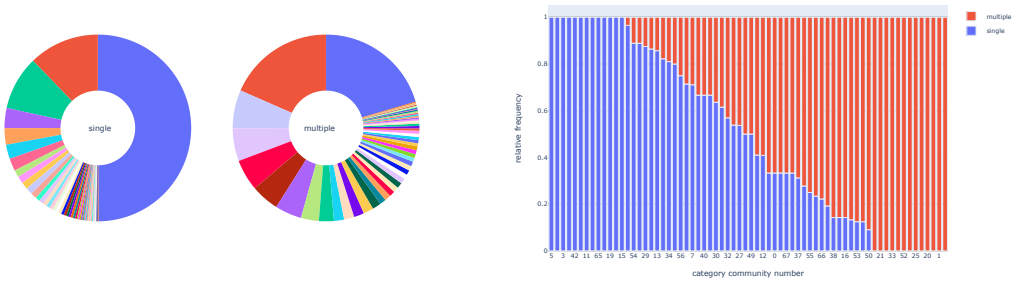

Figure 15: Left: Categories' frequency in the two main communities of the main character similarity graph. Right: Main character pattern for different categories. It can be seen that the two main character patterns include different categories. And for different categories, different main character patterns are used.

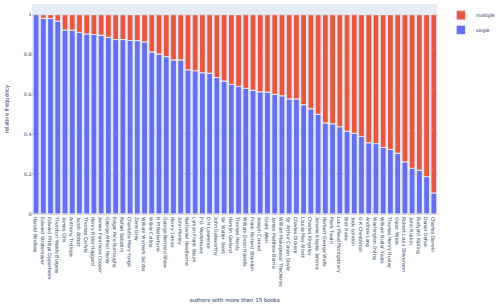

Figure 16: Right: Main character pattern normalized frequency for authors with more than 15 books