Nodes of Suspicion: Community Detection in the Character Network of 'The Murder of Roger Ackroyd'

Sartaj Bhuvaji
Computer Science Department
Seattle University
sbhuvaji@seattleu.edu

***Abstract*- This paper introduces an application of network analysis techniques to investigate character dynamics in Agatha Christie's " The Murder of Roger Ackroyd." By constructing a weighted network based on character co-occurrences using the Jaccard index, distinct communities of closely related characters are revealed. Community detection algorithms identify interpretable clusters, including a central group featuring Hercule Poirot and other key figures in the murder investigation, alongside clusters representing the Ackroyd family, suspects, and other characters. Analysis of network topology provides insights into narrative structure. This approach offers a quantitative view to understand character interactions and narrative dynamics, opening new avenues for data-driven literary analysis and interdisciplinary exploration.**

I. Introduction:

In today's digital world social media platforms connect billions of people and facilitate petabytes of information exchange. This vast network acts as a data pipeline presents an opportunity for research into human behavior, social interactions, influence, and information cascade. With users spanning diverse demographics, social media networks reflect the complexities of human behavior in real time. Social media networks such as Facebook, Instagram, and X host billions of users and represent connections between users. For a post on the platform, other users are free to engage by showing support in terms of likes or engage by commenting on posts or sharing the post with their network thus exchanging information.

Central to the study of social media analytics is the graph network analysis. This provides us with a robust frame to understand and measure the dynamics of a network. Network analysis introduces a set of mathematical and computational tools for examining the relationships and interactions between entities, such as people, and organizations which are conceptually represented as nodes. These nodes are connected via edges. These edges can be undirected, directed, or multiple. By understanding these connections, one can identify how information cascades throughout the system, identify central figures, uncover communities, and shed light on the behavior of nodes in the network.

Real-life networks can be immense, consisting of billions of nodes, thus making it computationally expensive to simulate a hypothesis over the network. Thus, we generate a random graph on the principle that 'friendships between nodes are formed randomly'[1]. By assuming that friendship (represented by a connected edge between two nodes) is randomly formed, we simplify the network creation, hoping that this creates a network that exhibits common characters observed in real-world networks.

In this paper, we construct a network of characters from the book 'The Murder of Roger Ackroyd' by 'Agatha Christie'. We also perform network analysis and visualize the connections between different nodes and understand how the network helps us visualize the relationship between characters in the story. In the later part, we experiment with multiple thresholds and understand how communities are formed amongst the characters.

II. Background

Agatha Christie was born in Torquay, England, and was hailed as the 'Queen of Mystery'[3] and stood out as the most prolific author in the genre of detective stories. With a career of over six decades, her keen understanding of human nature, along with her talent for crafting intricate plots and memorable characters, has earned her a place in history. Christie's iconic detectives, such as Hercule Poirot and Miss Marple, have become enduring symbols

of intelligence and deduction, captivating audiences with their brilliance and wit. Throughout her career, Christie wrote over 60 novels, numerous short stories, and several plays, cementing her legacy as one of the greatest storytellers of the 20th century.

The novel 'The Murder of Roger Ackroyd[2],' was published in 1926 and set in the quaint English village of King's Abbot, the novel introduces readers to the legendary detective Hercule Poirot. When the wealthy industrialist Roger Ackroyd is found murdered in his study, Poirot is called upon to unravel the complex web of secrets and lies that surround the case. As Poirot delves deeper into the investigation, he uncovers shocking revelations and unexpected twists that keep readers guessing until the very end. 'The Murder of Roger Ackroyd' is renowned for its ingenious plot, clever twists, and unforgettable conclusion, solidifying Agatha Christie's reputation as a master of the mystery genre.

Random Graphs

Graph models serve as powerful tools in various fields, including social media analytics, providing a mathematical framework for representing and analyzing complex relationships and interactions among entities. First proposed independently by Edgar Gilbert, Solomonoff, and Rapoport[1] random graph models are used to simulate relations in social media is the random graph generation model. This model is represented as

$$G(n, p)$$

Where n is the number of nodes and p is the probability of an edge forming between two nodes. Another way of generating random graphs is to assume that both the number of nodes n and the number of edges m are selected from the set $\binom{n}{2}$ possible edges. Let Ω denote the set of edges with n nodes and m edges. To generate a random graph with n nodes and m edges is :

$$\Omega = \binom{n}{2} m$$

In a graph of social media, it is important to identify nodes that share similarities. This helps us understand the structural organization of this network. The authors' Kamal Berahmand, et. Al[4] propose a new local community detection algorithm called Expanding Core nodes using Extended Similarity (ECES) to detect communities in complex networks. Local similarity methods are techniques used to measure the similarity between nodes within a network. These methods are particularly useful for analyzing relationships between nodes in large networks. In the paper, the authors use the Extended Jaccard[4] similarity as one of the parts to compute similar nodes.

$$Similarity(i, j|S) = A_{ij} \frac{N(i, s) \cap N(j, s)}{N(i, s) \cup N(j, s)}$$

Where A_{ij} is the matrix adjacency $N(i, s)$ is the neighborhood of i and $N(j, s)$ is the neighborhood of j. The ECES method of detecting communities uses a node centrality measure based on extended Jaccard similarity to identify core nodes with high embeddedness as the center of communities. The core nodes are expanded by adding neighbors that meet a membership degree threshold. This is repeated iteratively until no more nodes can be added. The algorithm identifies all the communities in a network using only local information about nodes and their connections. The algorithm has a low time complexity of $O(n\ log\ n)$ allowing it to scale well for large networks.

However, for this study, we use the simple Jaccard similarity formula to cluster nodes. This has the advantage of being simple to calculate however the time complexity raises to $O(nm)$. Where n is the number of nodes and m is the number of neighbors each node has.

III. Metric:

Jaccard Similarity:

To calculate the distance/similarity between two characters in the story, we use the simple Jaccard similarity index[1]. For a given book, let P_x be the set of paragraphs where character x appears, P_y be the set of paragraphs where character y appears. Then P_x union P_y is the number of paragraphs where both characters appear and P_x intersect P_y is the number of unique paragraphs where either character appears. Thus, we calculate Jaccard similarity as:

$$Jaccard\ Similarity = \frac{P_x \cap P_y}{P_x \cup P_y}$$

For our experiments, we keep edges between nodes with a similarity greater than 1.5.

Distance metric properties:

1. $D(x,y) = 0\ if\ and\ only\ if\ x = y$:

If $D(x,y) = 0$, then $|P_x \cap P_y| = 0$, which means there are no common paragraphs where both x and y appear. This implies that x and y never appear together in any paragraph. Therefore, x and y must be the same character. Conversely, if $x = y$, then every paragraph where x appears is also a paragraph where y appears (since they are the same character), $so\ |P_x \cap P_y| = |P_x| = |P_y|$, and thus $D(x,y) = 0$.

2. $D(x,y) > 0\ if\ and\ only\ if\ x,y\ are\ not\ the\ same$:

If x and y are different characters, then there will be some paragraphs where one character appears, but the other doesn't, leading to $|P_x \cap P_y| < |P_x \cup P_y|$, which means $D(x,y) > 0$. Conversely, if x and y are the same character, then $D(x,y) = 0$ (as proven in the first point).

3. $D(x,y) = D(y,x)$:

This follows directly from the definition of the Jaccard similarity. The order of x and y in the formula doesn't affect the calculation of the intersection or union of their appearances in paragraphs.

4. $D(x,y) + D(y,z) >= D(x,z)$:

Let,

$$D(A,B) = \frac{A \cap B}{A \cup B}$$

Where A and B are sets.

Now, let's represent the sets in terms of the given problem:

$$A = P_x \cap P_y$$
$$B = P_y \cap P_z$$
$$C = P_x \cap P_z$$

We want to show that:

$$D(A,B) \leq D(A,B) + D(B,C)$$

We can prove this by using properties of set operations and the Jaccard similarity:

$$D(A,C) = (P_x \cap P_y \cap P_z)/(P_x \cup P_y \cup P_z)$$

Now,

$$D(A,B) + D(B,C) = \frac{P_x \cap P_y}{P_x \cup P_y} + \frac{P_y \cap P_z}{P_y \cup P_z}$$

$$= \frac{P_x \cap P_y}{P_x \cup P_y} + \frac{P_y \cap P_z}{P_x \cup P_y}$$

$$= \frac{P_x \cap P_y + P_y \cap P_z}{P_x \cup P_y}$$

Now we can show that the numerator of $D(A,C)$ is less than or equal to the numerator of $D(A,B) + D(B,C)$ And the denominator of $D(A,C)$ is greater than or equal to the denominator of $D(A,B) + D(B,C)$.

Numerator:

$$|P_x \cap P_y \cap P_z| \leq |P_x \cap P_y| + |P_y \cap P_z|$$

This holds true because any paragraph that appears in both $P_x \cap P_y$ and $P_y \cap P_z$ is also present in $|P_x \cap P_y \cap P_z|$

Denominator:

$$|P_x \cup P_y \cup P_z| \geq |P_x \cup P_y|$$

This holds true because adding another set (P_z) can only increase the size of the union of the sets.

Therefore,

$$D(A,B) \leq D(A,B) + D(B,C)$$

IV. Experiments:

We read the book as a text file. To get the character names from the text, we split the book into sentences using NLTK's sentence tokenizer. Then for each sentence, we identify the entity label of the person. Once we collect all the names, we store it in a set to avoid duplicates. This initial collection of names might also include special characters. This is then cleaned using regex and only alphabetical characters are maintained. For the final cleaning step, we manually delete words that are nouns but not names of characters. Thus, we get a clean list of character names.

To calculate Jaccard distance, for each character, we loop over the list of character names. For each pair of characters (P_x, P_y), we find the set of paragraphs where character x appears, the set of paragraphs where character y appears, the number of paragraphs where both characters appear, and the number of unique paragraphs

where either character appears. Once done we apply a threshold of 1.5 (to limit the number of edges) and store all the values in a CSV format: source, target, weight.

Nodes	Edges	Modularity
86	882	0.488

Community Detection Algorithm:

To detect communities, we use the Jaccard similar score[1]. For a Graph G, with Vertices $V = \{v_1, v_2, .. v_i\}$ and $N(v_i)$ being the set of neighbors of node v_i, we calculate Jaccard similarity using:

$$\sigma_{Jaccard}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$

For this experiment, we group a node v_i with v_j if, amongst all neighbors of v_i, v_j is above the threshold of 1.5 and has the highest similarity amongst all neighbors of v_i.

If v_j does not exist in a cluster, we create a new cluster with v_i, v_j. If v_j already exists in cluster C_i, we assign node v_i to the same cluster.

Algorithm: Community Detection using Jaccard scores.
Require: dataframe[columns='source','target','weight']

1. **return** A list of nodes and their associated community
2. neighbors = {node: [n_1, n_2, …], …}
3. for row in dataframe:
   a. neighbors[row[1]].append[row[2]]
4. **End for**
5. **communities** = [[ ]]
6. For source, targets in neighbors
   a. *max_jaccard* = 0, *max_jaccard_node* = None
   b. Calculate *max_jaccard*, *max_jaccard_node*
   c. Update community with **[source, max_jaccard_node]**
      i. If *max_jaccard_node* already exists in the list of community, append the source to the existing list.
7. **End for**
8. Return communities which is a list of lists

V. Results

For T_{min} = 1.50, T_{max} = 2.00, δ= 0.05

T_i	E	Avg. D	GC	SG (% visible)		P_i	D_i
1.50	882	20.512	0.739	Node	100	3.1	11
				Edge	100		
1.55	817	19.452	0.749	Node	97.67	3.2	11
				Edge	92.63		
1.60	727	17.31	0.755	Node	97.67	3.6	12
				Edge	82.43		
1.65	608	14.476	0.763	Node	97.67	4.3	14
				Edge	68.93		
1.70	453	13.324	0.764	Node	79.07	3.9	11
				Edge	51.36		
1.75	391	11.5	0.752	Node	79.07	4.7	13
				Edge	44.33		
1.80	291	9.387	0.737	Node	72.09	5.4	15
				Edge	32.99		
1.85	109	10.381	0.752	Node	24.42	1.6	4
				Edge	12.36		
1.90	56	8.000	0.797	Node	16.28	1.3	2
				Edge	6.35		
1.95	38	5.429	0.763	Node	16.28	1.9	4
				Edge	4.31		
2.00	28	7.000	1.000	Node	9.30	1	1
				Edge	3.17		

Table. 1: Evaluating Graph attributes at different thresholds

Ti: Threshold, E: Number of edges, Avg. D: Average Degree, Gc: Global Clustering Coefficient, SG: Size of Giant Component, Pi: Average Path Length in the Giant Component, Di: Network Diameter in the Giant Component.

Giant Component Observation:

At a threshold of 1.7, the Giant Component contains at least 75% nodes. At this stage, the giant component has 68 nodes and 391 edges with an average degree of 11.5. However, in the degree distribution plot below we can see that the power-law distribution is not observed for this dataset. The distribution looks more like a normal distribution.

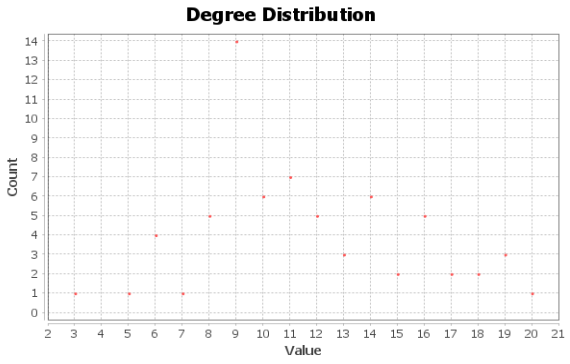

Fig. 1: Degree Distribution at $T_{1.7}$

Community Detection based on Jaccard similar score:

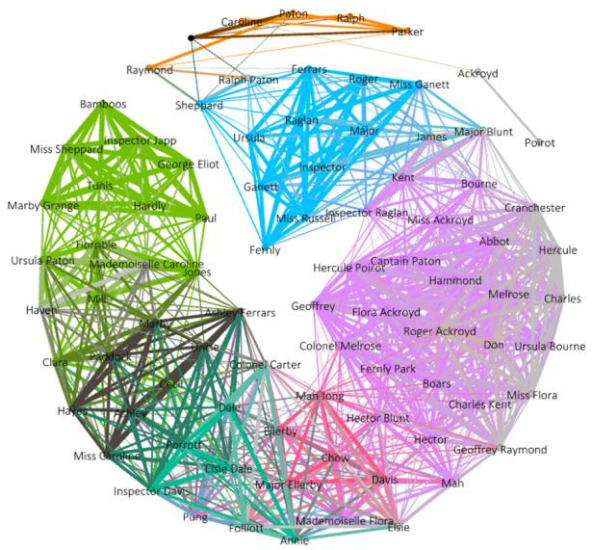

Fig. 2: Community Detection using Jaccard Similarity

ID	Characters
1	Ralph, Caroline, Parker, Paton, Raymond
2	Major, Roger Ackroyd, Fernly, Sheppard, Inspector Ganett, Ursula, Ferrars, Raglan, Miss Ganett, Miss Russell
3	Bourne, Miss Ackroyd, Inspector Raglan, Hammond, Melrose, Ursula Bourne, Flora Ackroyd, Geoffrey, Kent, Mah, Hector, Charles Kent, Mademoiselle Flora, Fernly Park, Colonel Melrose, Captain Paton, Boars, Abbot, Hector Blunt, Pung, Don
4	Don, Hammond, Roger Ackroyd, Paul, Tunis, Hardly, Ursula Paton, Inspector Japp, Bamboos, George Eliot, Marby Grange, Miss Sheppard, Jones, Cecil, Clara, Mill, Horrible
5	Ralph Paton, Sheppard
6	Haven, Mademoiselle Caroline, Ursula Paton
7	Davis, Ellerby, Chow, Major Ellerby, Mah Jong, Mah, Miss Caroline, Ashley, Hayes, Paddock, Ashley Ferrars, Marby, Uncle, Cecil, Poirot, Ackroyd
8-17	James, Major Blunt, Inspector Davis, Porrott, Elsie Dale, Annie, Dale, Geoffrey Raymond, Cranchester, Elsie, Charles, Mademoiselle Flora, Charles, Fernly Park

Table 2: Characters in respective clusters

Main Characters and Investigators (Cluster 1):
This cluster seems to contain the primary characters[2] involved in the investigation of Roger Ackroyd's murder. It includes Hercule Poirot, the main detective, along with other key figures like Inspector Ganett, Major, Roger Ackroyd himself, and others closely associated with the investigation.

Ackroyd Family and Associates (Cluster 2):
This cluster appears to revolve around characters directly related to Roger Ackroyd, his family members, and those associated with Fernly Park, Roger's estate. It includes characters like Miss Ackroyd, Flora Ackroyd, Geoffrey Raymond, and others closely linked to the Ackroyd family or Fernly Park.

Suspects and Associated Characters (Cluster 3):
This cluster likely contains characters who are potential suspects in the murder or those who have significant interactions with the main investigation. It includes characters like Ralph Paton, Don, and Hammond, who might have been under suspicion or involved in some way with the events surrounding Roger Ackroyd's death.

Secondary Characters and Locations (Clusters 4-6):
These clusters seem to encompass a mix of secondary characters, locations, and possibly minor plot elements. For example, Cluster 4 includes characters like Paul, Tunis, and Inspector Japp, along with some place names like Marby Grange. These characters and locations might play smaller roles in the story or have less direct involvement in the central mystery.

Family and Associates of Ralph Paton (Cluster 5):
This cluster seems to focus on characters related to Ralph Paton, possibly indicating a subgroup of characters associated with him or his storyline within the narrative.

Caroline Sheppard and Associates (Cluster 6):
This cluster appears to be centered around Caroline Sheppard, a character in the story, and possibly includes other characters closely associated with her or her storyline.

Miscellaneous Characters (Clusters 7-17):
These clusters contain a variety of characters, potentially with less significant roles or connections to the central plot. They include minor characters, references, or locations mentioned in passing throughout the story.

Network Analysis:
As we raise the threshold for connecting edges in the network, the average distance between nodes in the largest connected component initially increases. This

increase indicates that shorter connections are being added, linking smaller clusters together. However, this trend reverses around a threshold of approximately 1.8. This unexpected change is due to the addition of longer connections, which, although enhancing overall connectivity, can bypass existing shorter paths, resulting in a decrease in the average distance between nodes in the largest connected component.

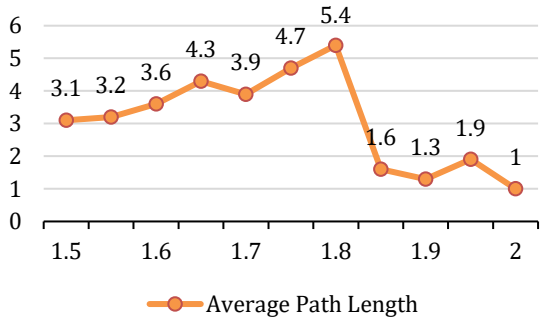

Fig. 3: Average Path Length over threshold

Graph Connectivity:

We analyzed the graph connectivity with increasing threshold. We observed that from $T_{1.5} – T_{1.7}$, the graph is totally connected and has a single giant component. However, after $T_{1.7}$, the graph does breaks into 3 separate major components.

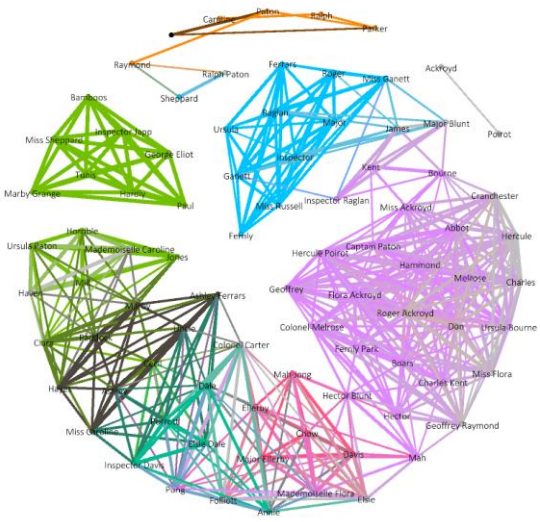

Fig. 4: Graph connectivity analysis at $T_{1.7}$

VI. Future Work:

While the implemented Jaccard similarity method provides good clustering results, there are several avenues for future research and improvement:

a. ECES-Method-implementation:
The authors[4] implement a modified Jaccard score in their algorithm. This new algorithm can be further tested on the dataset.

b. Chapter-Wise-Clustering
To better understand character introduction and role, we can cluster characters for each chapter. This would allow us to visualize how relationships among characters change.

VII. Conclusion:

This research introduces a novel application of network analysis techniques to explore the character relationships graph in Agatha Christie's "The Murder of Roger Ackroyd." By constructing a character co-occurrence network and using the Jaccard similarity metric, distinct communities of interconnected characters were uncovered, shedding light on their roles and connections within the narrative. Notably, central clusters emerged, including renowned detective Hercule Poirot and other key figures involved in the murder investigation, alongside separate clusters representing the Ackroyd family, potential suspects, and secondary characters. Through visualizing the character network and analyzing its topological properties, insights into the flow of information and relationships within the story's social fabric were gained. These findings highlight network analysis' importance in literary studies, offering a quantitative view to unravel relationships and explore subtle interactions.

VIII. References:

[1] Social Media Mining An Introduction by Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, April 20, 2014, IEEE

[2] The murder of Roger Ackroyd by Agatha Christie, https://www.gutenberg.org/ebooks/69087

[3] The home of Agatha Christie
https://www.agathachristie.com/, March 2nd, 2024

[4] Community Detection in Complex Networks by Detecting and Expanding Core Nodes Through Extended Local Similarity of Nodes Kamal Berahmand, Asgarali Bouyer, and Mahdi Vasighi