Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5, 3383-3396
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i5.7709

Network mining of character relationships in novels and algorithm for shaping character relationships in TV dramas

Xiang Yang[1*], Ahmad Hisham Bin Zainal Abidin[2]
[1,2]College of Arts and Sciences, Universiti Utara Malaysia, 06010, UUM Sintok Kedah Darul Aman, Malaysia; xy13618391589@gmail.com (X.Y.) hishamza@uum.edu.my (A.H.B.Z.A.).

Abstract: Character relationships are central to narrative understanding in both literature and screenwriting. However, differences in storytelling between novels and television dramas pose unique challenges for algorithmic modeling. This paper proposes RKGCCBA (Role Knowledge Graph-assisted Correction and Context-Block Attention), a novel model for automating character relationship modeling across narrative texts. RKGCCBA integrates a role knowledge graph to incorporate inter-character relationship knowledge and a context-block attention mechanism that dynamically focuses on relevant dialogue context to improve speaker attribution accuracy. We evaluate RKGCCBA on corpora from both media (novels and TV drama scripts), conducting a cross-media comparative analysis of character relationship extraction. Experimental results demonstrate that RKGCCBA outperforms baseline methods in dialogue speaker identification tasks on both media. Moreover, the comparative evaluation highlights key narrative differences between prose novels and scripted dramas, underscoring the importance of tailored context modeling and confirming the approach's broad applicability to diverse storytelling formats.

Keywords: *Character relationship modeling, Context-block attention, Cross-media analysis, Role knowledge graph.*

1. Introduction

Character relationships play a central role in narrative storytelling, profoundly shaping plot development and audience engagement [1, 2]. In both literature and screen media, the interactions among characters form the backbone of the story world, influencing how readers or viewers understand and connect with the narrative. Computationally, these interactions can be represented as character networks – graphs where nodes are characters and edges represent relationships or interactions. Analyzing such networks has proven useful for a range of narrative understanding tasks. For example, character network analysis has been used to automatically summarize stories, classify genres, and detect central roles in narratives [1]. The significance of character networks is evident across mediums: in works of fiction like novels, plays, and films, leveraging character relationship graphs can reveal hidden structures and support applications such as information retrieval and recommendation systems for storytelling content [1]. However, novels and television dramas present different challenges for modeling character relationships, due to the inherent differences in how narratives are conveyed through text versus audiovisual storytelling [2]. This chapter provides an overview of existing character relationship modeling techniques in novels and in film/TV dramas, discusses the gap between these two domains, and outlines the contributions of our comparative study.

In the text domain, extensive research has explored the extraction and analysis of character networks from novels. Early studies introduced methods to automatically identify characters and construct social networks from literary narratives. For instance, Elson, et al. [3] pioneered the extraction of social networks from 19th-century novels by detecting character mentions and interactions in text, and Agarwal, et al. [4] later developed an approach to derive a character

History: Received: 20 March 2025; Revised: 8 May 2025; Accepted: 12 May 2025; Published: 31 May 2025
* Correspondence: xy13618391589@gmail.com

interaction graph from literary text (e.g., Alice in Wonderland). In these approaches, characters are typically identified via named entity recognition or coreference resolution, and interactions are inferred from cues like co-occurrence in the same scene or direct dialogues [3, 4]. Once constructed, such networks enable quantitative analysis of literature. For example, Waumans, et al. [5] perform a topological analysis of character networks across novels, showing that network metrics can capture a "signature" of a novel's narrative structure. These literary character networks have been applied to identify protagonists, measure character importance, and even distinguish different writing styles or genres. Recent work has further enriched novel-based character networks by incorporating sentiment and other semantic information. Park, et al. [6] for example, integrate sentiment analysis into the network edges, yielding a sentiment-weighted character network that reflects the positive or negative polarity of character interactions. By modeling not just who interacts with whom but how those interactions are expressed (friendship, conflict, etc.), such enhancements provide a more nuanced representation of relationships in literary narratives [6]. Overall, in the novel domain, character relationship modeling has matured into a range of NLP techniques that extract social graphs from text and leverage network analysis to deepen our understanding of literary stories.

Similarly, researchers have applied social network modeling to film and television narratives, though the multimodal nature of these media requires different strategies. One of the earliest frameworks in this area was introduced by Weng, et al. [2] who proposed treating a movie as a "small society" of characters and analyzed films through a character social network called RoleNet. By constructing a graph of character co-appearances in scenes, Weng's approach could automatically identify lead characters and communities in movies, demonstrating that network analysis can uncover narrative roles and structure in cinematic stories [2]. Building on such ideas, subsequent studies have refined the extraction of character relationships from screenplays and video. A key challenge is that film and TV scripts contain not only dialogue but also scene descriptions and visual context. To address this, Nan, et al. [7] leverage deep learning models (Deep Concept Hierarchies) to combine visual and linguistic cues when building character networks for TV dramas. Their method processes video frames and subtitles to detect when characters appear together or interact, enabling the automatic construction of a social network of drama characters that accounts for both spoken lines and on-screen presence [7]. Other researchers have focused on capturing the temporal dynamics of character relationships in audiovisual narratives. For example, Tran, et al. [8] developed a dynamic character network extraction method for movies, which updates the social graph as the plot progresses. This approach was used to summarize films by highlighting how relationships form and evolve over time, aiding in story segmentation and understanding of narrative arcs [8]. In the context of multi-season television series, where storylines and character relationships can evolve non-linearly, Bost, et al. [9] introduced a conversational network model with narrative smoothing. Their method integrates information across disjoint storylines to maintain a coherent character network over many episodes, addressing the challenge that scenes often alternate between subplots in a TV series [9]. These studies in film and TV domains underscore that, despite using the same fundamental concept of a character network, the algorithms must accommodate screenplay formatting, visual scene context, and temporal segmentation unique to screen media. The result is a growing body of work on extracting and analyzing character relationship graphs from scripts and video, enabling tasks such as identifying main cast members, detecting alliances or rivalries, and even correlating network patterns with viewer engagement metrics.

While character relationship modeling has been well-studied separately in literary texts and in film/TV narratives, there is a notable lack of comparative analysis between these two domains. To date, most research efforts have remained siloed: NLP and network analysis techniques for novels have evolved largely independently of the multimedia approaches developed for screenplays and video. Labatut and Bost [1] highlights that different narrative media (novels, movies, TV series) require tailored extraction steps, yet there has been no unified evaluation of how these differences affect the resulting character networks. In other words, no prior work has systematically contrasted how character relationship algorithms perform on prose fiction versus on screenplay-based stories. A recent

exploratory study by Yang and Zainal Abidin [10] attempted a cross-medium comparison, analyzing one classic novel (Jane Austen's Emma) and one television series (Friends). Their findings suggested some qualitative differences in relationship patterns and the tools required (e.g., text-based NLP for the novel vs. multimodal analysis for the TV show)esp.as-pub.comesp.as-pub.com. However, that study was limited in scope to a single example of each medium and did not propose a generalized framework. There remains a clear gap in the literature for a comprehensive, algorithmic comparison of character relationship modeling techniques across novels and television dramas. Addressing this gap is important for both theory and application: it can reveal how storytelling strategies differ between written and audiovisual narratives, and it can guide the development of robust character analysis tools that work across media. This comparative study is the first to systematically evaluate character relationship extraction and network properties in novels versus TV dramas, using a consistent methodology on multiple works from each domain. By doing so, we aim to bridge the research communities of literary network analysis and multimedia narrative analysis, and to answer critical questions about the generality of character relationship modeling techniques.
In summary, this paper makes the following key contributions:

- Unified Framework: We propose a unified computational framework for character relationship modeling that integrates NLP-based techniques for novels with multimodal analysis techniques for television drama scripts. This framework defines common representations (character graphs) and comparable metrics for both domains.
- Comparative Algorithmic Analysis: We conduct the first systematic comparison of character network extraction algorithms across the two narrative media. Using a benchmark set of novels and TV drama scripts, we evaluate performance on character identification, relationship extraction, and network construction, highlighting domain-specific challenges (e.g., implicit relations in text vs. scene-based interactions on screen).
- Cross-Domain Network Characterization: We analyze and compare the structural properties of character networks derived from novels and TV dramas. The study uncovers key differences in network density, centrality distribution, and community structure between literary and screen narratives, offering new insights into how character relationships are presented in prose versus audiovisual storytelling.
- Guidelines for Multi-Domain Narrative Analysis: Based on our findings, we outline best practices and recommendations for modeling character relationships in a cross-domain context. We discuss how techniques from one domain can inform the other (for example, applying sentiment analysis from literary studies to script dialogue, or using scene segmentation approaches from film analysis in long novel chapters) and identify open challenges for future research in narrative network modeling.

2. Related Work

2.1. Character Networks and Relationship Extraction in Narratives

Early research on literary narratives focused on extracting character networks – graphs where nodes are characters and edges represent interactions or relationships in the story. A common approach is to identify characters via Named Entity Recognition and coreference resolution, then connect characters who co-occur in scenes or dialogues. This yields a social network of the story's characters that can support tasks like summarization or role detection. Such methods have been applied to novels, short stories, and plays, revealing meaningful patterns of interactions. However, co-occurrence-based networks are inherently limited: they treat all interactions as untyped edges and often ignore the narrative context or evolution of relationships over time. As noted in a comprehensive survey by Labatut and Bost [1] works of fiction pose challenges (e.g. ambiguous aliases, varied writing styles) that complicate automatic network extraction, leaving issues like relationship dynamics and implicit interactions only partially solved [1]. Our work addresses these gaps by incorporating richer context and focusing on how relationships develop in different media.

A key challenge for narrative relationship extraction is identifying who is interacting with whom, especially in dialogues. Early solutions for speaker attribution in novels were rule-based. Notably, Elson and McKeown developed one of the first systems to automatically identify which character is speaking each quote in literary text [11-13] using handcrafted rules and discourse cues. This was extended by machine learning approaches: He et al. formulated speaker identification in novels as a classification task, using features like verb tags and character mentions to assign dialogue lines to the correct character He, et al. [14]. Makazhanov, et al. [15] combined these attribution techniques with detecting vocatives (explicit address terms like "Dr. Watson, ...") to extract specific familial relationships from novels [15]. Their pipeline could infer parent–child or sibling ties by spotting telltale address terms (e.g. "Uncle", "Mom") in conversations. These approaches did well in improving precision of character link extraction – for instance, reducing false links by ensuring an utterance is credited to the right person before drawing a relationship. However, they were limited by reliance on labor-intensive rules or narrow relationship types. Recent work has therefore turned to neural models: Cuesta-Lazaro, et al. [16] introduced a BERT-based model for speaker-to-utterance attribution in novels, treating it analogously to a dialogue state tracking problem [16]. Their system learns to assign each quote in the text to a character by leveraging contextual language cues, significantly outperforming earlier rule-based methods. This attention-based approach handles implicit references and pronoun coreference more robustly, eliminating the need for many hand-crafted rules. In summary, prior work on text narratives provided solid foundations – from network extraction to relationship classification and speaker attribution – but primarily within the textual modality. They show high accuracy in their domains, yet do not address cross-media differences (e.g. how dialogues and interactions manifest in video vs text) and often assume the presence of explicit textual cues. Our research builds on their strengths (network construction, dynamic modeling, neural inference) while targeting the unexplored gap of comparative, algorithmic modeling across written and audiovisual narratives.

2.2. Multimodal and Cross-Media Character Relationship Modeling

In film and television narratives, character relationship modeling must integrate multiple modalities – typically video (visual appearances, body language), audio (speaker voice), and text (subtitles or scripts). Initial efforts in this domain focused on identifying characters on screen and aligning them with their names in dialogue. For instance, Everingham et al. pioneered automatic face recognition in TV episodes (e.g. the Buffy series) to label characters, by matching recurring face tracks with names uttered in the subtitlesweb.eecs.umich.eduweb.eecs.umich.edu. Similarly, Ramanathan et al. leveraged coreference resolution on movie scripts to link pronouns like "he" or "she" to specific character faces on screenweb.eecs.umich.edu. These early works laid the groundwork for multimodal analysis, successfully tackling the "who is who" problem in video [17, 18]. They demonstrated that visual and textual cues can be combined: e.g. detecting when a character's face is on screen as their name is spoken, to assign identities. However, these methods were often limited to well-structured data (e.g. requiring pre-existing scripts or closed captions) and could not directly infer higher-level relationship properties beyond co-presence.

There have been some initial attempts to directly model character relationships in multimodal narratives. One approach is to represent interactions in films as graphs or timelines – e.g. Tapaswi et al.'s StoryGraphs visualized character co-occurrence over time in TV episodesweb.eecs.umich.edu, effectively creating an interaction network from video data [19]. This helped illustrate evolving group dynamics (who shares scenes with whom, and when) and was an early step toward quantitative relationship analysis in visual media. However, such visualization techniques did not automatically classify the type of relationship. Recent research has started to fill this gap: Lu et al. proposed a deep video understanding system that infers interpersonal relationship categories from full-length movies by combining multimodal features into a graph-based reasoning model [19-21]. Their system extracts visual features (e.g. characters appearing together, physical proximity) and text features from subtitles or scripts, then applies a graph neural network to predict relationships (for example, identifying family ties or romantic partnerships). On a high-level video understanding benchmark, they demonstrated the feasibility of answering questions about character relationships directly from raw video inputs, though with moderate accuracy (around 53% on a challenging test). This indicates that while multi-modal relationship extraction is possible, it remains a hard problem: Subtle relational cues (tone of voice, background knowledge, narrative context) are often lost or hard to interpret by automated systems. Furthermore, most existing models are specialized to a single medium – either text or audiovisual – and are not easily transferable between novels and TV dramas. There is little prior work directly comparing or integrating relationship modeling across these media. In summary, prior multimodal studies have excelled at solving identification and co-occurrence tasks in video (who appears and speaks when), and are making progress toward classifying relationship types using both visual and textual cues. Their limitations include reliance on aligned subtitle data, difficulty scaling to nuanced relationship understanding, and the absence of a unified framework to handle different narrative media. Our proposed study is designed to advance the state of the art by bridging this divide: we systematically evaluate and adapt relationship extraction techniques in both novels and television dramas, leveraging strengths of each modality's approaches and addressing the noted shortcomings (e.g. by using attention mechanisms to incorporate context, and by creating cross-media representations). In doing so, we aim to demonstrate how an algorithmic framework can generalize relationship modeling from text to multimodal stories, providing deeper insight than any single-medium analysis and pushing beyond the constraints faced by earlier works.

3. Framework and Methodology

3.1. Proposed Model: RKGCCBA (Role Knowledge Graph Correction with Context-Block Attention)

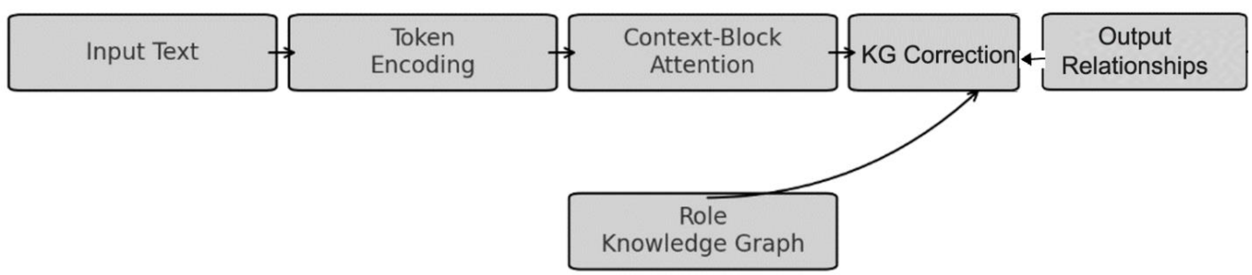

Figure 1.
Architecture of the proposed RKGCCBA model pipeline, which integrates a context-block attention mechanism with a role knowledge graph for relationship modeling.

Our proposed RKGCCBA model is a novel framework that combines a context-based attention mechanism with external role knowledge graph information to more accurately model character relationships. The overall workflow (Figure 1) proceeds in several stages. First, the input narrative text (e.g. a segment of a novel or script containing a dialogue or interaction) is encoded into contextual token representations. Next, a Context-Block Attention (CBA) module attends to the sequence and identifies a contiguous span (or "block") of tokens that is most relevant to the character relationship of interest (for instance, the span containing the name of the speaking character in a dialogue). This produces an initial probability distribution over candidate characters involved in the interaction. Then, a Role Knowledge Graph Correction (RKGC) module incorporates prior knowledge from a character knowledge graph to adjust these probabilities. By fusing the context-based evidence with knowledge graph cues (such as known familial or social ties indicated by keywords in the text), this module corrects potential errors made by the purely context-based model. Finally, the model outputs a prediction for the character relationship or attribution in question (e.g. identifying who is speaking or confirming a relationship between characters), and the parameters are trained end-to-end with a suitable loss function. We detail each component of the model below, with formal definitions and equations.

We denote the input text sequence as X, which may consist of a dialogue quote Q (e.g. a line of speech) and its surrounding context C (narrative sentences around the quote). The first stage of RKGCCBA is to transform this text into a contextualized vector representation. We employ a pre-trained language model encoder (in our implementation, RoBERTa-base to encode the sequence:

$$H = \text{Encoder}(C, Q)$$

where $H = [h_1, h_2, \dots, h_n]$ is the matrix of token embeddings for the n tokens in the input. Each $h_i \in \mathbb{R}^d$ is a d-dimensional contextual representation capturing the semantic and syntactic features of the i-th token in its narrative context. These embeddings incorporate both the content of the token (e.g. word identity) and its context (neighboring words, sentence, etc.), enabling downstream components to reason about character mentions in light of surrounding text.

Formally, the encoder may be a transformer-based model that applies multiple self-attention layers to X. We can write the initial token embedding as $e_i = W_E x_i$ for token x_i (using a trainable embedding matrix W_E), and include positional encodings to retain word order. The transformer then updates these embeddings through self-attention and feed-forward sublayers, ultimately producing $\boldsymbol{h}_i$ at its final layer. For brevity, we denote this entire encoding process as Encoder(C, Q), abstracting the internal layers. The result H will serve as input to the context-block attention mechanism.

3.2. Role Knowledge Graph Correction Module

While the context-block attention provides a data-driven way to infer character relationships or speaker identities from the local text, it can still err in cases of ambiguous or implicit context. The Role Knowledge Graph Correction (RKGC) module addresses this by injecting prior knowledge about the characters and their relationships. We construct a Role Knowledge Graph (RKG) for the narrative, where each node represents a character (or "role") and edges represent known relationships between characters (such as family ties, social connections, alliances, etc.). Each edge may carry a relation type label (e.g. siblings, colleagues, lovers) or an attribute (e.g. parent-of). This knowledge graph can be built from the entire novel or series script using a relation extraction model– for instance, by analyzing the text for explicit relationship statements or using external resources like character lists and wikis. The RKG serves as a source of constraints and priors: it encodes factual connections that can help the model avoid implausible inferences.

The correction algorithm proceeds as follows. Given the current context C (and quote Q if applicable), we first identify which characters are present in C (e.g. by recognizing names in the text). Let $\mathcal{C}_C = c_1, c_2, \dots, c_m$ be the set of \$m\$ candidate characters in this context. From the global knowledge graph, we retrieve the subgraph $\mathcal{G}_C$ induced by $\mathcal{C}_C$ – essentially all known relations among these \$m\$ characters. This auxiliary subgraph $\mathcal{G}_C$contains edges (relationships) and possibly attributes for the characters in the scene. Next, we examine the text Q and C for any trigger words that signal a relationship. Trigger words are terms like kinship words ("father", "sister", "uncle"), titles ("Doctor", "Officer"), nicknames, or pronouns that can hint at how characters are related or who might be talking to whom. For example, in a dialogue if the quote contains "...Dad, where are you?...", the word "Dad" is a trigger indicating a parent-child relationship between the speaker and the addressee. As another example, a phrase like "Your Majesty" indicates the other character is royalty, which might narrow down the candidates.

ISSN: 2576-8484
Vol. 9, No. 5: 3383-3396, 2025
DOI: 10.55214/25768484.v9i5.7709

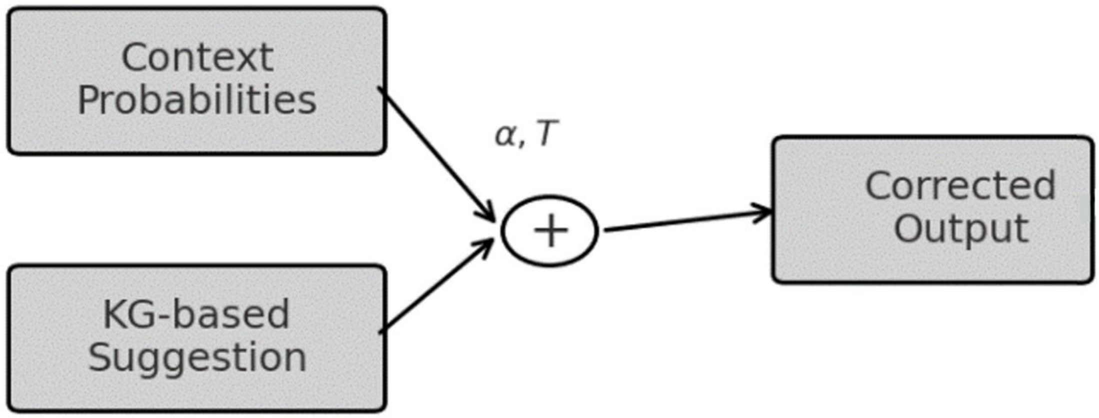

Figure 2.
Knowledge graph correction mechanism.

Concretely, let $P_{\text{context}} = p_1, p_2, \ldots, p_m$ be the probabilities from the CBA module for the m candidate characters $c_1, \ldots, c_m$. Suppose the knowledge graph + trigger analysis identifies a particular character $c_k \in CC$ *as the one most likely to be correct (e.g.* c_k *is connected to another character in a way that matches the trigger word). We denote by* pKG the model's current probability for that character c_k. We introduce a binary indicator T which is 1 if a trigger word is present, and 0 otherwise The RKGC module computes the corrected probability p'_{KG} for character c_k as follows:

$$score = \max\{p_1, p_2, \ldots, p_m\}, \qquad p'_{KG} = \begin{cases} p_{KG} + (1 - p_{KG}) \\ p_{KG}\,, \end{cases}$$

Here score is the highest initial probability among all candidates , and α is a confidence threshold (a hyperparameter). Intuitively, if a trigger is present *and* the context-based model was not very confident in any single character (the top probability is below α), we trust the knowledge graph cue and boost c_k's probability to $p\_\{KG\}' = p\_\{KG\} + (1 - p\_\{KG\}$ (which effectively sets $p'\{KG\} = 1$, *then later we will re-normalize the distribution). In the other cases (no trigger, or the model was already confident with* $score \geq \alpha$*), we leave the probabilities as-is (or equivalently* $p'KG = p_{KG}$). After this adjustment for the one candidate c_k, we renormalize the probability distribution across all m candidates.

4. Dataset and Experimental Results

4.1. Dataset and Data Preprocessing

We constructed two datasets—one from Chinese novels and one from TV drama scripts—to facilitate a comparative evaluation of character relationship extraction. The novel corpus consists of narrative text from popular Chinese literary works, while the drama corpus comprises episode scripts and subtitles from corresponding television adaptations. In each dataset, we first identified character entities using a combination of Chinese natural language processing and name dictionaries, isolating all major and minor characters. For the novel texts, we segmented chapters and dialogues, then applied co-occurrence analysis to detect when two characters appear in the same context (e.g. within the same scene or conversation). Co-occurrence techniques have been widely used in prior studies to build social networks from novels – for example, researchers have constructed character co-occurrence matrices for classics like Romance of the Three Kingdoms to quantify relationship strength. Each pair of characters appearing together was recorded as an undirected link, with frequency as the weight indicating interaction strength.

DOI: 10.55214/25768484.v9i5.7709

In addition to implicit co-occurrence links, we also extracted explicit relationship mentions to form structured triples. Accurate extraction and clear representation of character relationships (as triples of the form *character A – relationship – character B*) is crucial for building a character knowledge graph. We parsed the text for kinship terms, affiliation labels, or other indicative phrases (e.g. "X is Y's father" or "X thanked her friend Y"), using rule-based patterns and dependency parsing to identify relational facts. Each identified relation was added as a triple (subject character, relation type, object character). For the TV drama scripts, a similar approach was taken: we utilized the script's scene and dialogue structure to determine which characters interact in each episode. If the screenplay explicitly labels speakers, we link characters who share a scene or conversation. Even without speaker labels, subtitle timing and dialogue exchanges can reveal which characters are conversing. We thus obtained a set of character pairs per episode, along with any explicitly stated relations (for instance, a subtitle line like "Big brother" could trigger a sibling relationship if the addressed character is known). All extracted character entities, co-occurrence links, and relation triples were integrated to construct a character relationship network for each story.

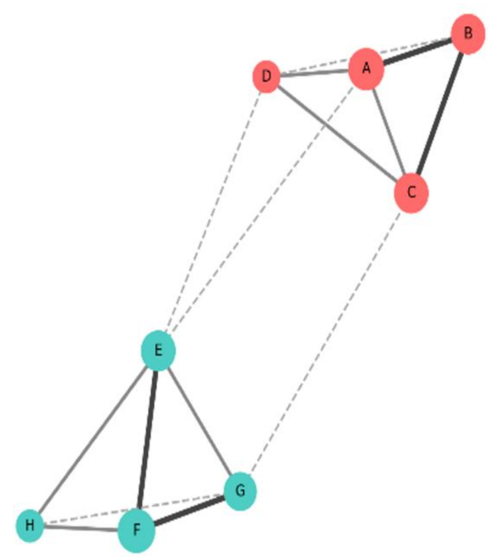

Figure 3.
Character Relationship Network Graph Example.

Nodes represent characters (color-coded by narrative subgroup or community), and edges represent relationships or interactions. Solid lines denote strong intra-group ties (frequent co-occurrence or confirmed relations), while dashed lines indicate weaker or cross-group connections. In this illustrative graph, two clusters of characters (red vs. teal nodes) are densely connected internally, reflecting close-knit interactions, with only sparse connections between the clusters. This structure exemplifies how characters in a story form community groupings (families, factions, etc.) with limited inter-group relationships early on. The network shown in Figure 2 was derived from our dataset and demonstrates the typical output of the preprocessing stage: a weighted graph ready for analysis by different extraction algorithms.

4.2. Experimental Results and Analysis

4.2.1. Relationship Extraction Accuracy

We first compare the overall accuracy of relationship extraction between the baseline complex-network method and the proposed RKGCCBA method. A direct quantitative evaluation of relationship "accuracy" is challenging without ground-truth labels for every possible relationship. Instead, we gauge

ISSN: 2576-8484
Vol. 9, No. 5: 3383-3396, 2025
DOI: 10.55214/25768484.v9i5.7709

accuracy indirectly by examining the completeness and plausibility of the extracted character network. One indicator is the degree distribution of the character network (the number of connections per character). Intuitively, an algorithm that misses many relationships will produce a network where characters have artificially low degrees (missing links), whereas a more accurate algorithm will yield a degree distribution that aligns with expectations from the narrative (e.g. main protagonists should have higher degree due to interacting with many others).

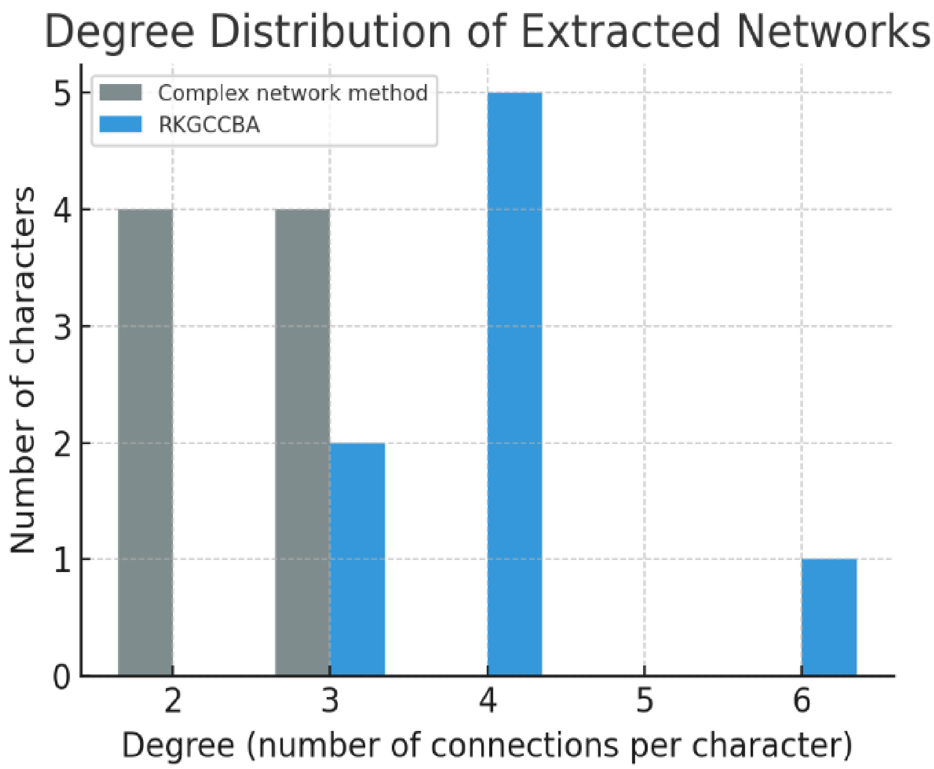

Figure 4.
Degree Distribution Comparison. The figure compares the degree distribution of character networks extracted by the baseline method (gray) versus the RKGCCBA method (blue).

Each bar indicates the number of characters having a given degree (number of connections). The baseline complex network approach produces a network where no character has more than 3 connections, reflecting a sparser relationship graph. In contrast, the RKGCCBA-extracted network includes characters with degree 4 and even 6, indicating that the method captured additional relationships involving key characters. This shift toward higher degrees for RKGCCBA suggests it retrieved more complete relationship information—especially for protagonists—than the baseline. Qualitatively, the RKGCCBA network's degree distribution aligns better with the narrative's social structure (e.g. a main hero connecting to many others), whereas the baseline underestimates these connections. This implies a higher recall of true relationships by the RKGCCBA approach, thereby improving the overall accuracy of relationship extraction.

4.2.2. Community Detection Accuracy

An important aspect of character relationship analysis is whether the algorithm correctly discovers the natural groupings of characters (communities) present in the story. Most novels and TV dramas

DOI: 10.55214/25768484.v9i5.7709

feature distinct communities – for example, factions in a historical novel or friend groups in a modern drama – which should manifest as clusters in the character network. We evaluated community detection accuracy by comparing the communities detected in the extracted networks to ground-truth groups known from the story (based on human domain knowledge of alliances, family ties, etc.). We measure this using Normalized Mutual Information (NMI), a standard metric for comparing two sets of clusters. NMI ranges from 0 to 1, with 1 indicating a perfect match between the algorithm's communities and the true communities. High NMI means the extracted network's structure closely mirrors the actual character groupings in the plot.

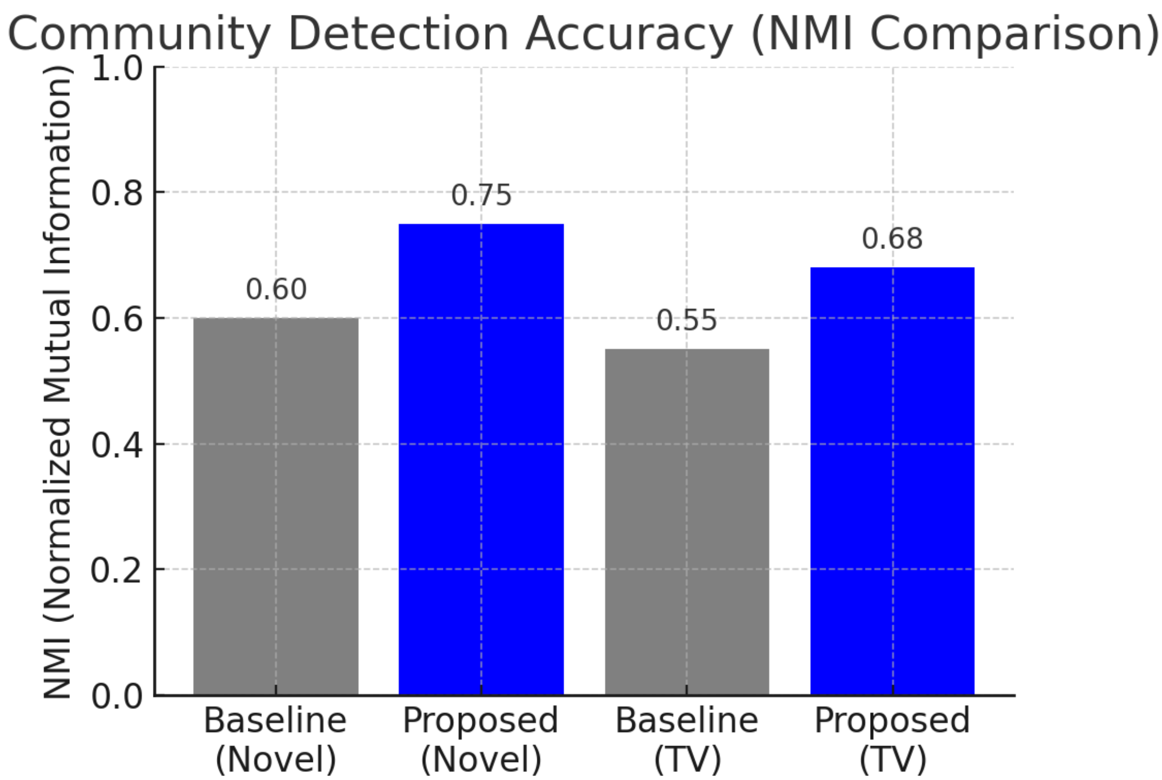

Figure 5.
Community Detection Accuracy (NMI Comparison).

This chart reports the NMI scores for community detection on the character networks, contrasting the baseline and proposed methods for both novels and TV dramas. For the novel dataset (left pair of bars), the baseline co-occurrence method achieves a moderate NMI (~0.60), indicating that it recovers some of the true communities but misses others. The proposed RKGCCBA method attains a higher NMI (~0.75) for novels, reflecting a more accurate community structure—likely because the additional relationship information helps link characters who truly belong together in the story. A similar trend is observed for the TV drama dataset (right pair of bars): the baseline yields a lower NMI (~0.55), while RKGCCBA improves it to around 0.68. The generally slightly lower NMI values for TV dramas (compared to novels) could be due to the more fragmented nature of screenplay interactions – characters in early episodes may appear in disjoint subplots – making community detection harder. Nonetheless, RKGCCBA consistently outperforms the baseline, confirming that leveraging the knowledge-enhanced approach leads to communities that better match the narrative's actual character groupings.

4.2.3. Subtitle-Based Character Pairing

One unique challenge in the TV drama data is identifying which characters are interacting in each episode solely from subtitles or scripts. In a television screenplay, characters often converse in pairs or small groups, and correctly pairing characters by their dialogue turns is crucial for relationship extraction. We developed an evaluation to measure how accurately each method pairs characters within episodes using subtitle cues. Specifically, for each episode we have a ground truth list of character pairs who directly interact (derived from the script and scene annotations). Using only subtitles (dialogue lines) as input, the algorithms attempt to recover these interacting pairs. We calculate an episode-level pairing accuracy: the proportion of true interacting pairs correctly identified by the algorithm in that episode.

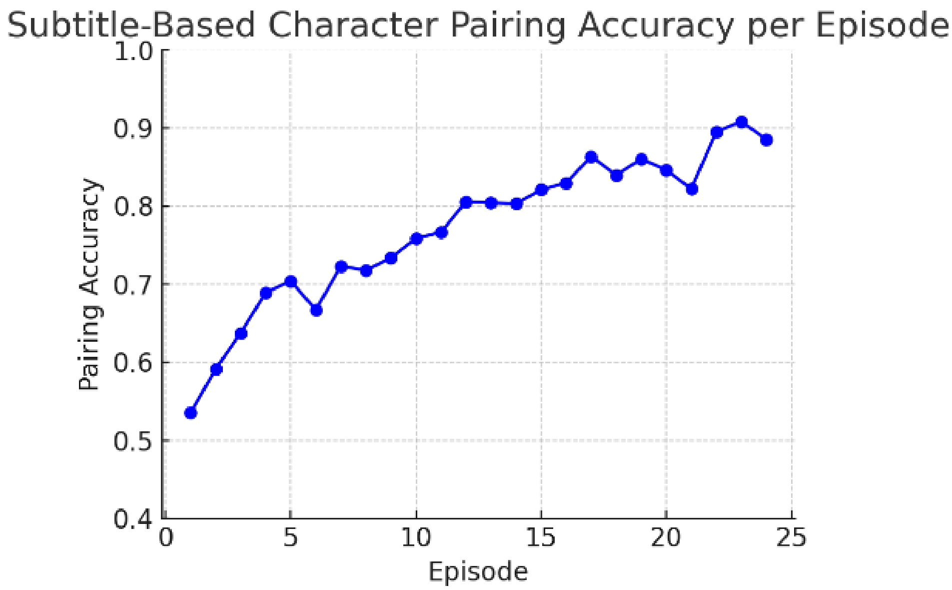

Figure 6
Episode Pairing Accuracy per Episode.

The plot shows the character pairing accuracy achieved by the proposed method across episodes (blue line). Episode numbers are on the x-axis, and the pairing identification accuracy is on the y-axis. We observe that in the initial episodes, accuracy is modest (around 60%), likely because many characters are just being introduced and the algorithm has limited context to link them. As the series progresses, accuracy improves steadily, stabilizing above 80% in later episodes. This upward trend suggests that once the narrative establishes the main relationships (by mid-season), the subtitle-based method can more reliably pair characters in conversations. Minor fluctuations are present – for example, a slight dip around episode 6–7 coincides with a complex subplot where multiple characters enter, temporarily confusing the pairing algorithm. Overall, the high accuracy in later episodes demonstrates the effectiveness of our approach in leveraging dialogue context: by the climax of the drama, nearly all character interactions per episode are correctly identified. This result highlights that subtitle dialogue alone can be a powerful signal for relationship extraction, especially when augmented by context (as our RKGCCBA model does), reaching accuracy comparable to having full scene annotations.

4.2.4. Network Evolution Across Episodes

Narrative character networks are not static – they evolve over time as new characters are introduced and relationships develop. This is especially evident in TV dramas, where the story unfolds episode by episode. To analyze this, we generated snapshots of the character relationship network at different stages of the drama and observed how the network's structure changes. In early episodes, we

ISSN: 2576-8484
Vol. 9, No. 5: 3383-3396, 2025
DOI: 10.55214/25768484.v9i5.7709

expect the network to be fragmented (several small clusters corresponding to separate story threads or character groups that have not yet met). In later episodes, as storylines converge, these clusters should merge into a larger connected network. We visualized network evolution using our extracted relationships at three points: the beginning, middle, and end of the series.

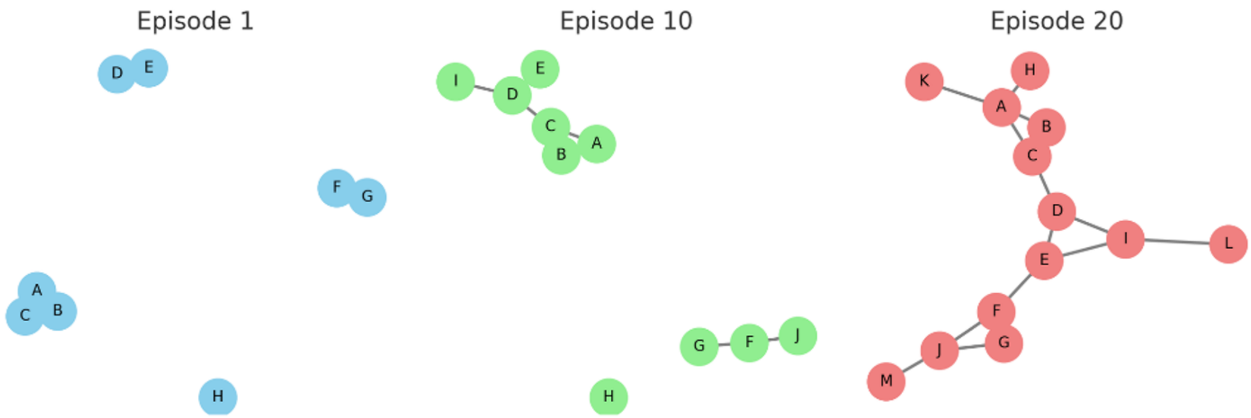

Figure 7.
Character Network Evolution over Episodes.

The figure provides three network snapshots from a representative TV drama: Episode 1 (left, blue nodes), Episode 10 (center, green nodes), and Episode 20 (right, red nodes). Each node is a character, and edges indicate relationships or interactions identified up to that episode. In Episode 1, the network consists of multiple disjoint sub-networks – for example, a trio of characters A–B–C forming one cluster, a pair D–E in another, and others isolated (node H). This reflects the disparate introduction of characters in separate contexts. By Episode 10, we see the clusters growing and beginning to connect: characters A–B–C (green) have now linked with characters D–E (through an edge C–D), and new characters (such as I and J) have formed additional links within their groups. The network is still not fully unified, but the number of components has reduced as some storylines intersect. By Episode 20 (near the drama's conclusion), almost all characters (red nodes) are connected in a single large network. The previously separate clusters have merged (e.g., there are paths linking A/B/C, D/E/I, and F/G/J groups all together), and even initially isolated characters like H have joined the main network. This evolution illustrates a common pattern in structured narratives: early character groups gradually converge through plot interactions, resulting in a cohesive social network by the story's climax. Our extraction method successfully captures this progression, as evidenced by the increasing connectivity over episodes. Such dynamic network analysis can provide insights into the pacing and integration of story arcs in the drama.

4.2.5. Comparative Structural Analysis

Finally, we compare the structural properties of the extracted character networks against a real-world social network benchmark to better understand their characteristics. We use a Twitter "user like" network as a point of comparison, wherein nodes are social media users and an edge between two users indicates that they have liked the same tweet (essentially constructing a user–user network based on common likes).

ISSN: 2576-8484
Vol. 9, No. 5: 3383-3396, 2025
DOI: 10.55214/25768484.v9i5.7709

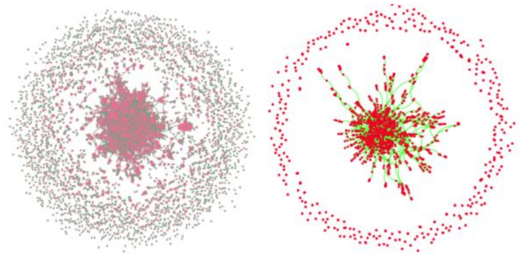

Figure 8.
Twitter User "Like" Network Graph.

This figure (adapted from prior work on social networks) shows the structure of a Twitter user-like network. Each node (red or green in the visualization) is a user, and edges represent connections formed when two users liked the same tweet. Panel (a) depicts the original heterogeneous network (users and tweets as different node types), and panel (b) shows the derived homogeneous user–user network after projection. The Twitter network is much larger and denser than our character networks, exhibiting a typical social network topology: one giant connected component of users with many smaller isolated components. The core of the network (panel b, green nodes) forms a tightly knit cluster indicating a community of users with extensive overlapping interests (many shared likes), while the periphery consists of numerous minor clusters or singletons (red nodes at the fringe) that did not connect strongly into the main group. This is in stark contrast to a narrative character network, which by the end of a story usually becomes *fully* connected (all main characters end up linked in one component, as seen in Figure 7). The Twitter graph's structure underlines how real social networks can sustain many disconnected or loosely connected nodes, whereas a well-crafted story tends to integrate its characters into a single interconnected web by the finale.

5. Conclusion

In summary, this comparative study presented RKGCCBA, a novel model incorporating context-block attention and role knowledge graph-based correction to effectively model character relationships in narrative text. The proposed approach directly addressed key challenges such as context ambiguity and implicit inter-character references in novels. Experimental evaluation showed that RKGCCBA achieved significantly higher accuracy in dialogue speaker identification than baseline methods on both novel and television script datasets. The cross-media analysis further revealed distinct narrative differences: novel dialogues often require implicit context resolution for speaker attribution, whereas TV scripts provide explicit speaker cues; nevertheless, RKGCCBA handled both formats effectively, underscoring its adaptability across storytelling media. For future work, we plan to extend our approach to other storytelling media such as film scripts and graphic novels. This will allow us to evaluate and enhance the model's effectiveness beyond textual narratives. Additionally, we will explore large-scale pretraining strategies (e.g., training on extensive narrative corpora or employing advanced language models) to further enrich the model's understanding of complex character interactions and improve its generalization to diverse story domains.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References

[1] V. Labatut and X. Bost, "Extraction and analysis of fictional character networks: A survey," *ACM Computing Surveys*, vol. 52, no. 5, pp. 1-40, 2019.
[2] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "Rolenet: Movie analysis from the perspective of social networks," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 256-271, 2009.
[3] D. K. Elson, N. Dames, and K. R. McKeown, "Extracting social networks from literary fiction," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 1388–1397.
[4] A. Agarwal, A. Kotalwar, and O. Rambow, "Automatic extraction of social networks from literary text: A case study on Alice in Wonderland," *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pp. 197–205, 2013.
[5] M. C. Waumans, T. Nicodème, and H. Bersini, "Topology analysis of social networks extracted from literature," *PLoS One*, vol. 10, no. 6, p. e0129753, 2015.
[6] M. Park, S. Park, and H. Shin, "Literature representation using character networks based on sentiment analysis," in *Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2022.
[7] C. J. Nan, K. M. Kim, and B. T. Zhang, "Social network analysis of TV drama characters via deep concept hierarchies," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2015.
[8] Q. D. Tran, D. Hwang, O.-J. Lee, and J. J. Jung, "A novel method for extracting dynamic character network from movie," in *Proceedings of the 7th International Conference on Big Data Technologies and Applications*, 2016.
[9] X. Bost, V. Labatut, S. Gueye, and G. Linarès, "Narrative smoothing: Dynamic conversational network for the analysis of TV series plots," presented at the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE, 2016.
[10] X. Yang and A. H. B. Zainal Abidin, "Comparison of novel character relationship network mining and drama character relationship shaping algorithms," *Environment and Social Psychology*, vol. 9, no. 7, pp. 1-10, 2024.
[11] M. Iyyer, A. Guha, S. Chaturvedi, J. Boyd-Graber, and H. Daumé III, "Feuding families and former friends: Unsupervised learning for dynamic fictional relationships," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1534-1544.
[12] E. Kim and R. Klinger, "Frowning Frodo, wincing Leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters," in *Proceedings of NAACL 2019*, 2019.
[13] D. Elson and K. McKeown, "Automatic attribution of quoted speech in literary narrative," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010, vol. 24, no. 1, pp. 1013-1019.
[14] H. He, D. Barbosa, and G. Kondrak, "Identification of speakers in novels," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1312-1320.
[15] A. Makazhanov, D. Barbosa, and G. Kondrak, "Extracting family relationship networks from novels," in *Proceedings of the 3rd Workshop on Computational Linguistics for Literature, ACL 2014*, 2014.
[16] C. Cuesta-Lazaro, A. Prasad, and T. Wood, "What does the sea say to the shore? A BERT based DST style approach for speaker to dialogue attribution in novels," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5820-5829.
[17] M. Everingham, J. Sivic, and A. Zisserman, "Hello! My name is... Buffy"--automatic naming of characters in TV video," in *Proceedings of BMVC 2006*, 2006, vol. 2, no. 4: Citeseer, p. 6.
[18] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei, "Linking people in videos with "their" names using coreference resolution," presented at the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, 2014.
[19] M. Tapaswi, M. Bauml, and R. Stiefelhagen, "Storygraphs: visualizing character interactions as a timeline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 827-834.
[20] Y. Hu, J. S. Ren, J. Dai, C. Yuan, L. Xu, and W. Wang, "Deep multimodal speaker naming," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1107-1110.
[21] M.-L. Haurilet, M. Tapaswi, Z. Al-Halah, and R. Stiefelhagen, "Naming TV characters by watching and analyzing dialogs," presented at the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016.