Large Language Models Fall Short: Understanding Complex Relationships in Detective Narratives

Runcong Zhao[1*], **Qinglin Zhu**[1*], **Hainiu Xu**[1], **Jiazheng Li**[1], **Yuxiang Zhou**[1]
Yulan He[1,2,3], **Lin Gui**[1]
[1]King's College London, [2]University of Warwick, [3]The Alan Turing Institute
{runcong.zhao, qinglin.1.zhu, hainiu.xu, jiazheng.li, yuxiang.zhou}@kcl.ac.uk
{yulan.he, lin.1.gui}@kcl.ac.uk

Abstract

Existing datasets for narrative understanding often fail to represent the complexity and uncertainty of relationships in real-life social scenarios. To address this gap, we introduce a new benchmark, *Conan*, designed for extracting and analysing intricate character relation graphs from detective narratives. Specifically, we designed hierarchical relationship categories and manually extracted and annotated role-oriented relationships from the perspectives of various characters, incorporating both public relationships known to most characters and secret ones known to only a few. Our experiments with advanced Large Language Models (LLMs) like GPT-3.5, GPT-4, and Llama2 reveal their limitations in inferencing complex relationships and handling longer narratives. The combination of the *Conan* dataset and our pipeline strategy is geared towards understanding the ability of LLMs to comprehend nuanced relational dynamics in narrative contexts.

1 Introduction

Tasks like multi-agent interaction (Park et al., 2023; Xu et al., 2023; Wang et al., 2023) and character-centric narrative understanding (Zhu et al., 2023) have recently gained significant attention. These tasks require a deeper understanding of complex relationships among multiple entities (Worth, 2004), thereby serving as a critical benchmark for assessing the reasoning capabilities of LLMs (Bubeck et al., 2023). Detective stories, where characters often adopt multiple identities or aliases that are revealed at various points, are the most appropriate testbed for assessing LLMs' capability of deducing complex relationships. However, existing datasets designed for character-centric narrative understanding are either built on well-known stories that LLMs have been trained on (Brahman et al., 2021; Sang et al., 2022; Chen et al., 2023; Iyyer et al., 2016), or consist of simpler texts (Bamman et al., 2020; Stammbach et al., 2022; Xu et al., 2022), such as children's stories, where characters and their relationships are typically introduced when the characters first appear in the narrative (Zhao et al., 2023).

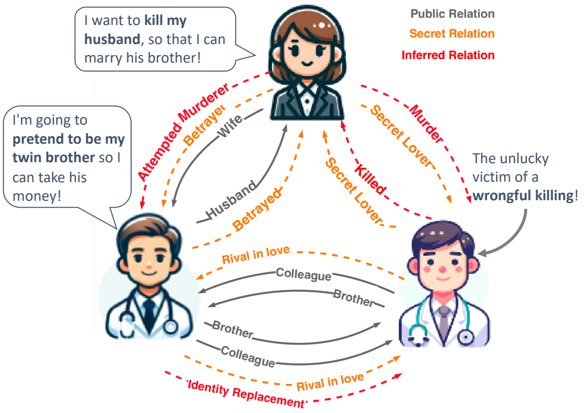


Figure 1: The example illustrates complex relationships of characters in narratives. Gray-colored relationships represent surface-level information, widely known to most characters. Orange-colored relationships, on the other hand, are secrets known to only one or very few individuals, often conflicting with the commonly known relationships; these are referred to as secret relationships. Red-colored relationships represent inferred information, meaning they are not explicitly stated in any character's story but can be deduced by synthesising information from all characters collectively. LLMs struggle with such complex relationships in long narratives.

However, relationships between characters are often characterised by **incomplete and uncertain** information in reality. These references may involve descriptions from another character's perspective, as in *"A middle-aged man walked out of the main gate, with grey and white hair, and a slender build."*. Also, in real social scenarios, this is often not the case. For instance, as depicted in Figure 1, husband A may remain unaware that his brother B is the secret lover of his wife C and the biological father of his son D. Each individual may have differing interpretations of their relationships, with these perceptions potentially **conflicting** based on their own perspectives.

[*] Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2024, pages 7618–7638
August 11-16, 2024 ©2024 Association for Computational Linguistics

Misidentified relationships can greatly impact the core conflicts and plot of a story, as these complex relationships are often central to the narrative. Therefore, we have developed the benchmark, COntextual Narrative ANalysis (*Conan*) to understand complex relationships in detective narratives. We have also outlined the desired input-output format, identified three sub-tasks within this framework, and developed hierarchical relationship categories for evaluation, drawing insights from the field of social science. *Conan* is constructed to extract role-oriented relational graphs from detective narratives. It comprises detective narratives from various characters' perspectives, along with annotated relationships that can be deduced from the narrative. As shown in Figure 1, it includes three types of relationships: (1) *Public Relations* that are known to most people; (2) *Secret Relations* that are known only to a few or even just one person and often conflicting with the widely known relationships; (3) *Inferred Relations*, which are not explicitly articulated in any single character's story and must be deduced by combining information from multiple characters. This dataset can act as a benchmark to test the cognitive and inferential abilities of LLMs. Additionally, it holds the potential to improve LLM capabilities in areas such as narrative comprehension (Zhao et al., 2023), simulation agents (Park et al., 2023), and game agents (Xu et al., 2023; Wang et al., 2023).

Our quantitative experiments and qualitative analysis show our benchmark is challenging for cutting-edge LLMs (GPT-3.5, GPT-4, and Llama2). Our findings reveal that these models struggle primarily due to two reasons: (1) the complexity of information that necessitates inferential reasoning; (2) the length of narratives which demand efficient key information extraction. Additionally, our evaluation across three distinct strategies has pinpointed the most effective strategy for various scenarios, while also highlighting those that are inefficient and underperforming. Given the high operational costs of LLMs, these insights are valuable for saving time and resources in future research[1].

In summary, we have made the following contributions:

- We have constructed and annotated the *Conan* dataset for the task, designed to evaluate and understand the inference capacity of LLMs.
- We have designed hierarchical relationship categories for evaluation, built on insights from social science and necessary empirical observations during the process of manual annotation and LLMs' evaluation.
- To assess the performance of advanced LLMs, namely, GPT-3.5, GPT-4, and Llama2, we have conducted evaluations with three different strategies using our benchmark and have identified the most effective strategy for various scenarios.
- Our findings reveal that LLMs significantly underperform humans on *Conan*. We have carried out a series of experiments to validate our hypotheses regarding the causes of LLMs' failure.

2 Task Definition

As shown in Figure 2, the input narrative N consists of background stories for k characters, represented as $c_i \in C_k = \{c_1, c_2, ..., c_k\}$. The background story of each character N_{c_i} is crafted solely from the perspective of that particular character c_i, with all relationships and events in the story framed based on c_i's perception. Unlike the conventional narrative structure that revolves around a single protagonist, in our setup, the complete narrative N emerges as a collaborative novel, composed by intertwining the perspectives of these k characters, i.e., $N = \cup_{c_i \in C_k} N_{c_i}$.

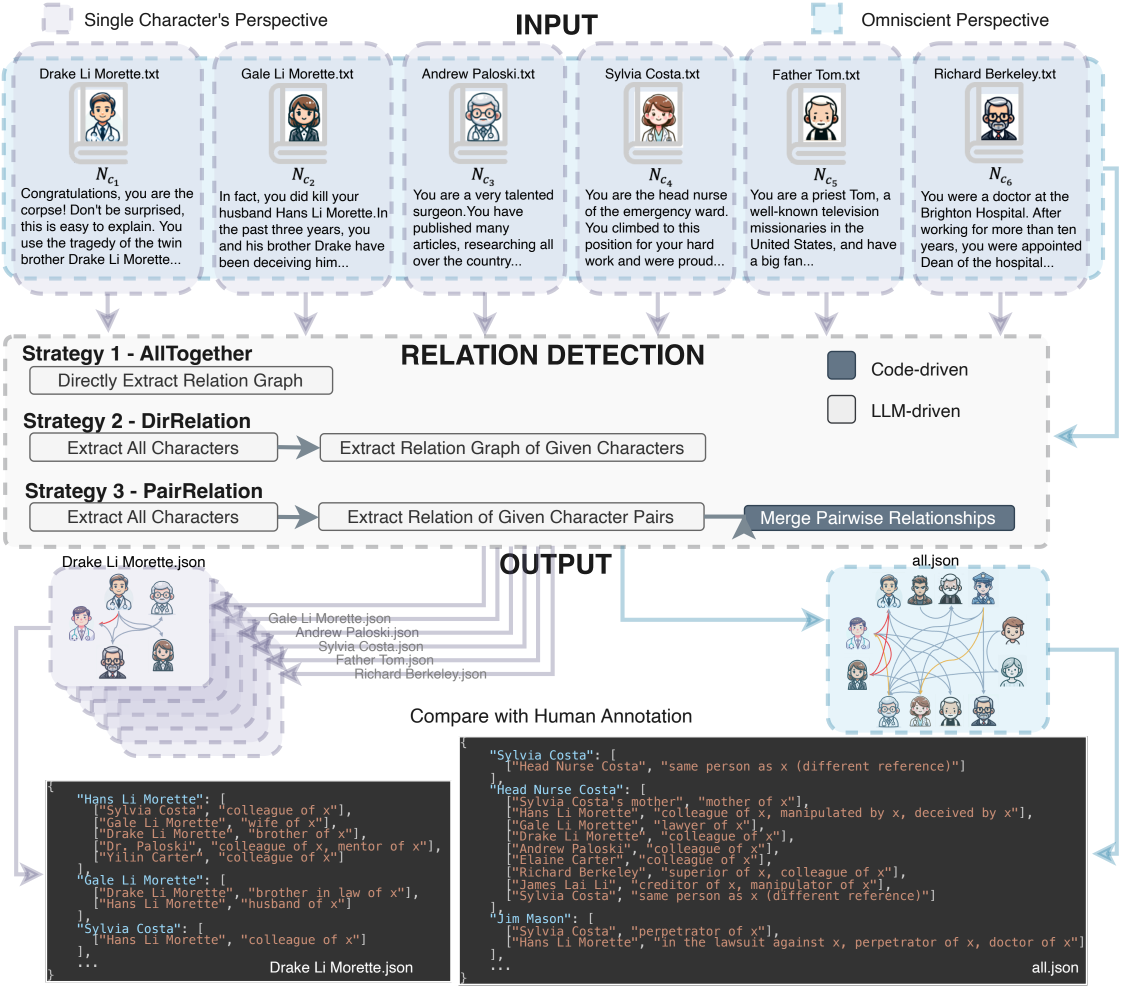
As depicted in Figure 1, for the same set of character relationships, there can be instances where some characters remain completely unaware of these connections, or where the perceptions of different characters are in direct contradiction. Moreover, the narrative may introduce additional characters beyond these k individuals. Some of these extra characters might play peripheral roles, while others could hold pivotal significance, such as the victim. Therefore, we define all characters that appeared in the story as $C = \{c_1, c_2, ..., c_K\}$, where $C_k \subseteq C$. Hence, by defining R_C as the relationships among this group of characters C, we aim to perform the following three sub-tasks:

1. **Character Extraction**. Identify all characters in the given story, which can be N_{c_i} or N.

2. **Entity Linking**. Recognise character relationships from the perspective of a specific character c_i. This equals $R_C|N_{c_i}$ formally.

3. **Relation Deduction**. Infer the actual relationships between characters by considering the collection of all character-centric narratives. Formally, this corresponds to $R_C|N$.

[1] Our code and dataset are available at https://github.com/BLPXSPG/Conan


Single Character's Perspective
INPUT
Omniscient Perspective
Drake Li Morette.txt
N_{c_1}
Congratulations, you are the corpse! Don't be surprised, this is easy to explain. You use the tragedy of the twin brother Drake Li Morette...
Gale Li Morette.txt
N_{c_2}
In fact, you did kill your husband Hans Li Morette.In the past three years, you and his brother Drake have been deceiving him...
Andrew Paloski.txt
N_{c_3}
You are a very talented surgeon.You have published many articles, researching all over the country...
Sylvia Costa.txt
N_{c_4}
You are the head nurse of the emergency ward. You climbed to this position for your hard work and were proud...
Father Tom.txt
N_{c_5}
You are a priest Tom, a well-known television missionaries in the United States, and have a big fan...
Richard Berkeley.txt
N_{c_6}
You were a doctor at the Brighton Hospital. After working for more than ten years, you were appointed Dean of the hospital...
Strategy 1 - AllTogether
Directly Extract Relation Graph
RELATION DETECTION
Code-driven
LLM-driven
Strategy 2 - DirRelation
Extract All Characters
Extract Relation Graph of Given Characters
Strategy 3 - PairRelation
Extract All Characters
Extract Relation of Given Character Pairs
Merge Pairwise Relationships
Drake Li Morette.json
OUTPUT
all.json
Gale Li Morette.json
Andrew Paloski.json
Sylvia Costa.json
Father Tom.json
Richard Berkeley.json
Compare with Human Annotation

```
{
    "Hans Li Morette": [
      ["Sylvia Costa", "colleague of x"],
      ["Gale Li Morette", "wife of x"],
      ["Drake Li Morette", "brother of x"],
      ["Dr. Paloski", "colleague of x, mentor of x"],
      ["Yilin Carter", "colleague of x"]
    ],
    "Gale Li Morette": [
      ["Drake Li Morette", "brother in law of x"],
      ["Hans Li Morette", "husband of x"]
    ],
    "Sylvia Costa": [
      ["Hans Li Morette", "colleague of x"]
    ],
    ...
}
```

Drake Li Morette.json

```
{
    "Sylvia Costa": [
      ["Head Nurse Costa", "same person as x (different reference)"]
    ],
    "Head Nurse Costa": [
      ["Sylvia Costa's mother", "mother of x"],
      ["Hans Li Morette", "colleague of x, manipulated by x, deceived by x"],
      ["Gale Li Morette", "lawyer of x"],
      ["Drake Li Morette", "colleague of x"],
      ["Andrew Paloski", "colleague of x"],
      ["Elaine Carter", "colleague of x"],
      ["Richard Berkeley", "superior of x, colleague of x"],
      ["James Lai Li", "creditor of x, manipulator of x"],
      ["Sylvia Costa", "same person as x (different reference)"]
    ],
    "Jim Mason": [
      ["Sylvia Costa", "perpetrator of x"],
      ["Hans Li Morette", "in the lawsuit against x, perpetrator of x, doctor of x"]
    ],
    ...
}
```

all.json

Figure 2: Input-Output Format and Benchmark Relation Detection Strategies. The input narrative consists of k background stories N_{c_i} that are uniquely created from the perspective of the character c_i. For each narrative, we have labelled it $k+1$ times, including k relationship graphs from each character's perspective and one relationship graph from omniscient perspective. Our objective is to extract all characters from the given story, including those beyond the initial k characters, subsequently detect the relationships among all the extracted characters, even when they involve false or multiple identities, and finally uncover conflicting relationships to deduce the genuine nature of these relationships.

To understand the relationships of characters within detective narratives, the challenges mainly manifested in three types, corresponding to the aforementioned three sub-tasks: First, in detective narratives, a character might have different identities or aliases, and these identities may surface at different points in the story. Secondly, LLMs struggle with basic inference based on existing information due to a phenomenon known as reversal curse (Berglund et al., 2023). For example, when we know that A is B's father, it does not necessarily imply that B is A's child according to these models. Finally, the third challenge involves drawing inferences from the perspectives of multiple characters, which sometimes yield conflicting or inconsistent information. For instance, from A's perspective, B might be regarded as his mother's friend. However, from B's perspective, he has been the lover of A's mother and is actually A's biological father.

3 Dataset Construction

To investigate the capabilities of LLMs in comprehending detective narratives, we developed the first open benchmark dataset for this task. The construction process includes data collection and processing, and annotation of characters and their relationships.

3.1 Data Collection and Processing

Collection We gathered the original narratives from the popular Chinese murder mystery games, where each player has a predefined role description. We utilised the background stories of these playable characters to construct our dataset for the following reasons. Firstly, they feature complex character relationships that often pose challenges for LLMs. Secondly, these narratives offer more realistic scenarios compared to existing synthetic benchmarks. Lastly, while these stories are considerably longer than current benchmarks, they still maintain a level of conciseness suitable for human reading. These equilibria ensure that the workload for human annotators, in terms of reading and comprehending these narratives, remains manageable.

Filtering We selected 100 high quality narratives from a pool of $2,135$, following the filter process described in Appendix A. Then we employed Adobe to extract text from scanned narratives and leveraged the capabilities of the ChatGPT model GPT-3.5 to rewrite and refine the text.

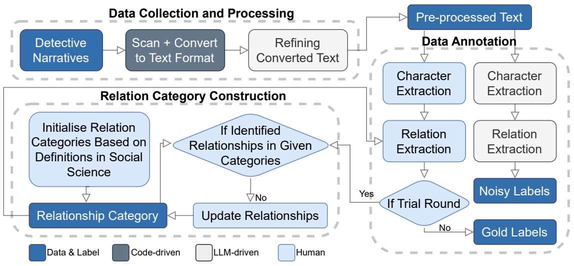


Figure 3: Dataset Construction.

3.2 Data Annotation and Evaluation

Relation Category Construction We started with relationship scheme extraction. Based on a relationship scheme of 53 casual relationships (Berscheid, 1994; DeVito, 2015), we carried out a trial run using 10 detective novels to expand upon these relationships. The resulting scheme contains 5 primary relationship categories, namely *Romantic*, *Family*, *Allies*, *Adversarial*, and *Business*, which expand to 163 fine-grained relationships denoted as R.

During this process, we found that the boundaries between some categories were not clearly defined, such as *"mother"*, *"mother-in-law"*, and *"adoptive mother"*. In contrast, mistaking *"mother"* for *"stranger"* is a more significant error than confusing these three specific relationships. To mitigate such ambiguities, we consolidated similar categories into broader ones until no further merging was feasible. For instance, we merged *"lover"*, *"boyfriend"*, and *"girlfriend"* into a single category: *"romantic relationships"*.

Our goal for constructing this hierarchical category was to better evaluate LLM capabilities. For instance, recognising *"biological daughter"* as *"child"* at a higher level would not be considered incorrect. However, a singular category could not achieve this balance. We reported F1 scores for each level. Consequently, we established a hierarchical structure of relationship categories, comprising 5 top-level categories, 54 intermediate categories, and 163 detailed categories. Such a well-defined, detective-oriented relationship scheme helps to reduce the potential subjectivity during the annotation process.

Evaluating free-form relationships generated by LLMs can be extremely challenging, requiring either human evaluation or using LLMs for auto-evaluation. However, human evaluation is costly and subjective, while evaluation by LLMs has been shown to have notable disparities compared to human judgement. Therefore, it is better to annotate the relationships between characters, providing a consistent basis for evaluating this dataset.

Labelling We recruited four annotators, all of whom were fans of detective narratives, and conducted training sessions for them. Our complete annotation pipeline involves three tasks: (1). *Character Extraction*: Annotators read given detective novels to identify all characters appearing in the narrative. (2). *Entity Linking*: annotators closely examine the story from the perspective of a single character, and extract both explicit and implicit relationships. These relationships are structured as triplets in the format of $(c_i, c_j, r_{i,j})$, where $r_{i,j}$ signifies the relationship between characters c_i and c_j. We specifically account for scenarios where $i = j$ to handle the common detective novel plot where a single character has multiple identities (refer to Appendix C for details). (3). *Conflict Detection and Relationship Refinement*: As our dataset comprises narratives from multiple character perspectives, annotators are tasked with considering all available information to form a unified relationship graph from the k individual detective narratives. This step often involves resolving conflicting information that arises from the imperfect knowledge shared among characters and refining the final relationships through inference.

Consequently, for each story, our team of experts would generate k distinct relationship graphs at the individual character level, and one consolidated graph that merges these individual perspectives.

Inter-annotator Agreement Following the agreement measure for triplets in previous works (Girju et al., 2007; Gurulingappa et al., 2012), we used F1-score as a criterion to measure inter-annotator agreement (IAA). We selected one challenging narrative and asked three annotators to annotate it following the annotation guidelines. This step served the purpose of evaluating the inter-annotator agreement and ensuring the quality of the annotations. The detailed annotation guideline and the calculation of the inter-annotator agreement score are in Appendix B. We calculate IAA for three steps in our task: 1). identifying characters, $IAA(c_i)$; 2). determining if there are relationships between a given character pair, $IAA(c_i, c_j)$; and 3). classifying the relationships between two characters when there are relationships between them, $IAA(c_i, c_j, r_k)$. $IAA(c_i, c_j, r_k)_1$ assesses whether annotators agree on at least one relationship for each character pair. $IAA(c_i, c_j, r_k)_{all}$ measures agreement on all relationship triplets. As shown in Table 1, annotators demonstrate high agreement in both character and relationship extraction. However, agreement decreases as the task becomes more subjective. For instance, identifying characters present in the narrative is more straightforward, but *"father's friend of x"* might be labelled as *"acquaintance of x"* by one annotator and as unrelated by another.

Annotator	Character	Relation		
	c_i	(c_i, c_j)	$(c_i, c_j, r_k)_1$	$(c_i, c_j, r_k)_{all}$
1 & 2	0.978	0.894	0.962	0.873
1 & 3	0.978	0.800	0.916	0.738
2 & 3	0.966	0.756	0.907	0.736
Average	0.974	0.817	0.928	0.782

Table 1: Inter-annotator Agreement Score.

3.3 Data Statistics

The detailed statistics are shown in Table 2. We collected a total of 2,135 narratives, but the majority of them exhibited low quality. After manual selection, we identified 100 high-quality narratives. We first generated annotations automatically using GPT-4, categorised under the column *All* in Table 2. Our dataset has an average of 27,695 tokens per narrative, making it significantly longer and more complex than earlier datasets, particularly considering its detective-themed content.

We also recruited human experts to annotate 25 narratives, resulting in a total of 8,254 annotations. They were compensated at an hourly rate of \$31.92, with each narrative estimated to take about 10 hours to complete. However, due to quality concerns, we removed one annotated narrative from our final dataset as it ambiguously described the relationships between characters, making the annotation highly subjective. Consequently, our final dataset consists of 24 annotated narratives encompassing 7,951 relationships.

Given the quality concerns, we did not use GPT-4's annotations for evaluation purposes. All experimental results reported in this paper were assessed using data annotated by human experts. However, the GPT-4 annotations could serve as an initial foundation for further annotation of this dataset in future research. The original narrative is in Chinese, but we also provide an English-translated version and conduct experiments on it, detailed in Appendix E.3.

4 Experiments

4.1 Baselines

We conducted our experiments using GPT-3.5, GPT-4 and Llama2-chat. We accessed GPT-3.5 and GPT-4 through the Azure API [2] with model version gpt-35-turbo-16k 0613(Default) and gpt-4-turbo 1106-Preview. The experiments were carried out with the default parameters of the interface, between October and December 2023. For Llama2, we used the HuggingFace model Llama-2-70b-chat-hf [3]. Inferences on this model were run directly using greedy decoding at a temperature setting of 0.

Dataset	**All**	**Human**
#Narratives	100	24
#Background stories	640	149
Avg. #Character per narrative	18.72	18.84
w/ narrative	7.40	7.21
w/o narrative	11.32	11.63
#Relationships	27,444	7,951
Avg. #Token per character story	4,327	4,539
Avg. #Token per narrative	27,695	28,182

Table 2: Dataset Statistics.

Given that the length of some narratives exceeds the input limit of LLMs, we segment the original

[2] https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models

[3] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

narrative input into multiple parts. The relation graph is initially extracted from the first segment. Subsequently, this pre-established relation graph, coupled with the ensuing narrative segment, is used as input to instruct the model to update and refine the relationship graph. This iterative process allows for comprehensive relationship mapping despite the constraints of LLM input limitations.

We evaluate relation detection using three strategies: First, *AllTogether*, where we ask the model to directly output a relationship graph, including characters and their relationships. Second, *DirRelation*, to distinguish and minimise the influence of errors occurring in two separate stages, character extraction, and relationship extraction, we extract the characters from the narrative first, then utilise it alongside the narrative script to generate the relationship network. Third, *PairRelation*, where we initially extract characters, inquire about the relationship of each character pair, and finally, aggregate the results, merging them into a comprehensive relationship map. This strategy targets LLM's problem of ignoring relationships in narratives, especially with long narratives.

The total running costs for using GPT-3.5 and GPT-4 in our experiments are approximately $400 and $2000, respectively. In addition, the running time for Llama2 in our experiments totalled 490 hours, utilising two 80G A100 graphics cards.

4.2 Corruption Rate

For generative language models, there is always a possibility that the output format may not follow the given instructions. Even though these models, are trained to follow specific formats like JSON, complex tasks such as *Conan* still demonstrate instances where models fail to comply with the provided format guidelines. Therefore, we classify cases failing to produce the desired format as corrupted cases. We calculate the corruption rate for various models and strategies as $\frac{n_c}{n_{all}}$, where n_c is the number of corrupted relationships and n_{all} is the number of total generated relationships. For each model, we add a post-processing step to remap the relationships not in the specified categories into desired categories. However, some outputs remain corrupted even after recategorising.

As indicated in Table 3, there was a significant reduction in the corruption rate following post-processing. However, our primary focus is the F1-score after self-correction. This step proved beneficial for Llama2 and GPT-3.5, but counterproductive for the best-performing model, GPT-4. Consequently, we chose to implement the self-correction step only for Llama2 and GPT-3.5 in our subsequent experiments.

Strategy	Llama2		GPT-3.5		GPT-4	
	before	after	before	after	before	after
			Corruption Rate			
AllTogether	0.445	0.316	0.310	0.032	0.143	**0.006**
DirRelation	0.448	0.286	0.266	0.022	0.164	**0.009**
PairRelation	0.563	0.336	0.160	**0.021**	0.180	0.032
			F1-score			
AllTogether	0.030	**0.035**	0.028	**0.031**	**0.125**	0.119
DirRelation	0.050	**0.057**	0.052	**0.053**	**0.110**	0.103
PairRelation	0.020	**0.021**	0.025	**0.025**	**0.027**	0.026

Table 3: Comparison of corruption rate and F1-score before and after self-correction. The corruption rate is the percentage of output relationships that fail to comply with the provided format guidelines before and after self-correction.

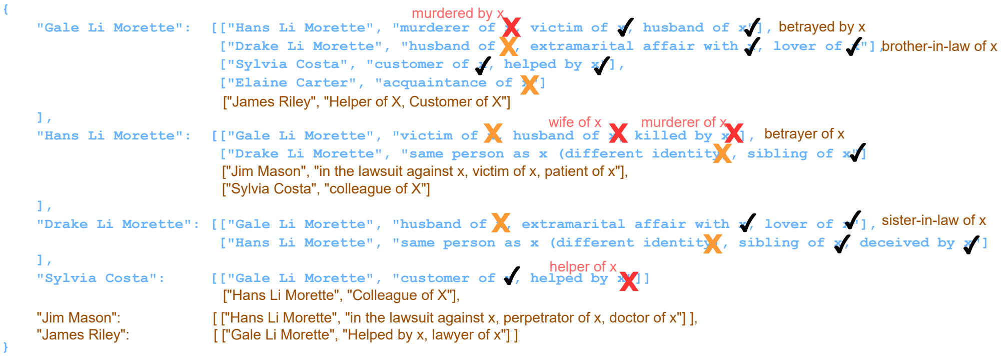
4.3 Character Extraction

We hypothesised that character extraction would be a simple task for LLMs, and planned to use it primarily for relationship extraction. Surprisingly, LLMs struggled with this initial step. These findings contribute to the research community by highlighting unanticipated challenges in character extraction for LLMs that may not be immediately apparent. Besides the F1-score of character extraction for three baseline relation detection strategies, we also report the results that we ask LLMs to directly extract characters from the given narrative, which were used in DirRelation and PairRelation. We denote it as DirCharacter in Table 4.

Strategy	Llama2			GPT-3.5			GPT-4		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
				Information from Single Character's Perspective					
DirCharacter	0.636	0.605	0.620	0.664	0.697	0.680	0.613	0.782	**0.687**
AllTogether	0.665	0.489	0.564	0.229	0.373	0.283	0.755	0.789	**0.772**
DirRelation	0.725	0.540	0.619	0.644	0.678	0.660	0.658	0.768	0.709
PairRelation	0.575	0.615	0.594	0.643	0.714	0.677	0.612	0.780	0.686
				Information from All Characters' Perspectives					
DirCharacter	0.449	0.703	0.548	0.522	0.836	**0.643**	0.444	0.870	0.588
AllTogether	0.671	0.261	0.376	0.240	0.238	0.239	0.736	0.455	0.562
DirRelation	0.730	0.332	0.457	0.502	0.517	0.509	0.681	0.442	0.536
PairRelation	0.371	0.598	0.458	0.478	0.844	**0.610**	0.430	0.869	0.575

Table 4: Character extraction results. Note that simply instructing LLMs to extract relationships of given characters, as in our "DirRelation" and "PairRelation" approaches, doesn't ensure consistent compliance with the instruction (the characters in the output could be different from what was given). Therefore, in addition to the "DirCharacter" approach, which directly asks LLMs to extract characters, we also evaluate the characters in the final output of the three strategies.

We can see that GPT-4 performs the best when



```
{
    "Gale Li Morette":   [["Hans Li Morette", "murderer of x ✗, victim of x ✓, husband of x ✓"],
                          ["Drake Li Morette", "husband of x ✗, extramarital affair with x ✓, lover of x ✓"],
                          ["Sylvia Costa", "customer of x ✓, helped by x ✓"],
                          ["Elaine Carter", "acquaintance of x ✗"]
                          ["James Riley", "Helper of X, Customer of X"]
    ],
    "Hans Li Morette":   [["Gale Li Morette", "victim of x ✗, husband of x ✗, killed by x ✗"],
                          ["Drake Li Morette", "same person as x (different identity ✗, sibling of x ✓"]
                          ["Jim Mason", "in the lawsuit against x, victim of x, patient of x"],
                          ["Sylvia Costa", "colleague of X"]
    ],
    "Drake Li Morette": [["Gale Li Morette", "husband of x ✗, extramarital affair with x ✓, lover of x ✓"],
                          ["Hans Li Morette", "same person as x (different identity ✗, sibling of x ✓, deceived by x ✓"]
    ],
    "Sylvia Costa":      [["Gale Li Morette", "customer of x ✓, helped by x ✗"]]
                          ["Hans Li Morette", "Colleague of X"],

    "Jim Mason":         [ ["Hans Li Morette", "in the lawsuit against x, perpetrator of x, doctor of x"] ],
    "James Riley":       [ ["Gale Li Morette", "Helped by x, lawyer of x"] ]
}
```

Figure 4: Error Analysis. An example output for *"Gale Li Morett.txt"* using GPT-4 and *"AllTogether"*, the best-performing combination. Output relationships are labeled as correct (✓) or incorrect (×). The three main error types are: relationships identified by LLM but not in the narrative, relationships in the narrative but not identified by LLM, and relationships with incorrect direction (e.g., LLM outputs Gale as Hans' husband instead of wife).

extracting information from a single character's perspective. However, its performance drops significantly when extracting information based on all characters' perspectives, a more pronounced drop than observed with Llama2 and GPT-3.5. Upon comparing extracted characters based on all characters' information using different LLMs, we noted that GPT-4 tends to extract more details, averaging 31.92 characters per narrative. In contrast, Llama2 extracted 25.54 characters, and GPT-3.5 extracted 26.08 characters, which are significantly less than the GPT-4's extraction. Consequently, these results demonstrate higher recall and precision for GPT-4 with shorter inputs, in comparison to the aforementioned models. Yet, for longer narratives, GPT-4's precision suffers due to issues like character duplication (e.g., "Costa", "Head Nurse Costa", and "Sylvia Costa" all referring to the same character) and misclassification of entities such as organisation names.

Errors can include personal pronouns such as "You", locations such as "Pavilion", organisations such as "Xiao", and objects like "Casket". Additionally, hallucinations and low Recall also harm the performance. Common error examples can be found in Appendix E.1.

4.4 Relation Extraction

After removing the corrupted cases, we calculate the F1-score based on the derived relationship triples. These triples consist of character i, character j, and the relationships between them, represented as $(c_i, c_j, r_{i,j})$. Here we define precision as $\frac{n_p}{n_g}$, where n_g is the number of generated triples after removing corrupted ones, n_p is the number of correct relationship triples among generated ones; and recall as $\frac{n_r}{n_l}$, where n_l is the number of labelled triples, n_t is the number of matched triples among labelled ones.

Strategy	Llama2			GPT-3.5			GPT-4		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Information from Single Character's Perspective									
AllTogether	0.129	0.054	0.076	0.056	0.041	0.047	0.283	0.269	**0.276**
DirRelation	0.160	0.085	0.111	0.119	0.076	0.093	0.219	0.267	0.240
PairRelation	0.025	0.089	0.039	0.029	0.099	0.045	0.047	0.258	0.080
Information from All Characters' Perspectives									
AllTogether	0.121	0.020	0.035	0.064	0.020	0.031	0.267	0.082	**0.125**
DirRelation	0.203	0.033	0.057	0.092	0.037	0.053	0.202	0.075	0.110
PairRelation	0.012	0.092	0.021	0.014	0.132	0.025	0.014	0.238	0.027

Table 5: F1-score of Relation Extraction.

As illustrated in Table 5, it is evident that both Llama2 and GPT-3.5 struggle significantly to extract relationships from long narratives. In comparison, GPT-4 demonstrates considerably better performance, although there remains a substantial gap compared to human understanding. As shown in Figure 4, even extracting relationships from a single character's perspective is challenging. As anticipated, extracting relationships based on information from all characters' perspectives is a more challenging task compared to extracting relationships based on information from a single character's perspective. This increased difficulty arises due to two factors:

More Complicated Information Narratives from a single character's perspective typically don't contain self-contradictory content. This is because the information is based on that character's perception, which often tends to be consistent and self-explanatory. Therefore, LLMs can just extract what is stated in the text. In contrast, information from all characters includes secrets, misunderstandings, lies generated for self-interest or specific goals, and even delusions caused by illnesses. Deriving accurate character relationships from these potentially repetitive or contradictory pieces of information is naturally more complex. LLMs must navigate through various narrative layers, distinguishing between truth, deception, and perception to accurately infer relationships.

To validate our assumption, we also calculate the accuracy of the relationships that are inconsistent across different characters' perspectives, which we identify as secret or inferred relations, as illustrated in Figure 1. We can see that the accuracy of those complicated relationships is lower compared to all labelled relationships in Table 6.

Strategy	Llama2		GPT-3.5		GPT-4	
	all	complex	all	complex	all	complex
AllTogether	0.020	0.012	0.020	0.017	0.082	0.072
DirRelation	0.033	0.012	0.037	0.028	0.075	0.064
PairRelation	0.092	0.045	0.132	0.081	0.238	0.157

Table 6: Comparison with Complex Relationships. The accuracy of all relationships and only the complex relationships.

Longer Narrative One reason we suspect for the low F1 scores is the "lost in the middle" phenomenon observed in long narratives, which has also been noticed earlier (Liu et al., 2023), where LLMs tend to put more attention on the beginning and the end of long inputs, often ignoring information in the middle. Consequently, additional information can enhance judgement for humans, however, when all characters' information is combined as input for LLMs, this extra data does not lead to improved results.

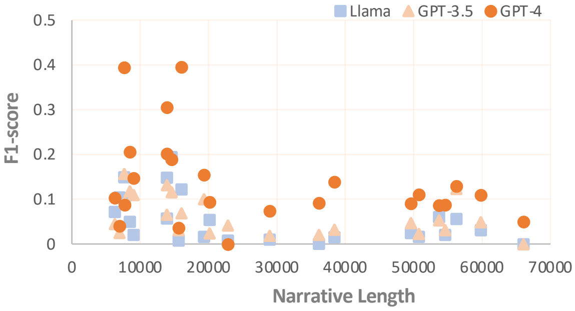


Figure 5: F1-score against the length of given narrative.

To validate our assumption, we plotted the F1-scores for each narrative against their respective lengths. The results suggest that longer narratives tend to yield lower F1-scores. However, shorter narratives do not guarantee higher F1-scores, as the results also heavily depend on the complexity of each narrative. Therefore, narratives with short lengths but complicated relationships can exhibit lower F1-scores due to their inherent difficulty.

4.5 Ablation Studies

Impact of Character Extraction To investigate the impact of character list quality on relationship extraction outcomes, we assessed the performance using both a gold standard and a model-generated noisy character list, as detailed in Table 7. Results show a significant performance increase with the provided characters. Table 5 reveals that while both Llama2 and GPT-3.5 showed improved results with DirRelation, GPT-4 performed better using AllTogether. Given GPT-4 did not do well in character extraction, its performance sets a ceiling for the efficacy of DirRelation and PairRelation, as their outcomes hinge on the quality of separately extracted characters.

Strategy	Llama2		GPT-3.5		GPT-4	
	gold	noisy	gold	noisy	gold	noisy
Information from Single Character's Perspective						
DirRelation	0.124	0.111	0.123	0.093	**0.315**	0.240
PairRelation	0.071	0.039	0.083	0.045	0.167	0.080
Information from All Characters' Perspectives						
DirRelation	0.053	0.057	0.064	0.053	**0.171**	0.110
PairRelation	0.046	0.021	0.055	0.025	0.113	0.027

Table 7: Impact Assessment of Character Extraction. We evaluated the impact on relationship extraction's F1-score using both the gold standard character list and the model-generated noisy character list.

Impact of Strategies To investigate the impact of various relation detection strategies, we compared the approaches previously discussed. We noticed that directly asking models to extract all relationships resulted in low recall. We also found that when inquiring complex relationships between the given two characters, they perform better. Therefore, a straightforward solution was to inquire about the relationships between each pair of characters, which we termed as the PairRelation strategy. While this approach did increase recall, it also significantly amplified hallucinations. For instance, when the relationship between characters a and b wasn't explicitly mentioned, the model often fabricated one. Additionally, this method is more costly and time-consuming: while originally we needed to calculate it at a complexity of $O(k)$, it now escalates to $O(k^3)$, where k is the number of characters. This translates to a hundredfold increase in cost for narratives with 10 characters.

Consequently, we infer that PairRelation is the least effective strategy. For GPT-4, when a reliable list of characters is available, DirRelation should be employed. In scenarios where a high-quality character list cannot be ensured, the AllTogether approach is preferable. For Llama2 and GPT-3.5, DirRelation is always the better strategy.

5 Related Works

Character Extraction Named Entity Recognition (NER) is a longstanding challenge in NLP (Wu et al., 2020; Aly et al., 2021). It involves locating and classifying named entities present in unstructured text. Character extraction is a specific area within NER, focusing on identifying characters involved in a given narrative (He et al., 2013; Bamman et al., 2020; Sang et al., 2022). In this process, models often confuse the targeted entities (characters) with other types of entities, such as organisations, items, locations, and so on. Another challenge is to merge all expressions in a given text that refer to the same entity, which is known as coreference resolution (Chen et al., 2017; Bohnet et al., 2022). This task becomes even more challenging when dealing with long-distance mentions (Massey et al., 2015; Yu et al., 2023a). In detective narratives, misidentified characters, entwined relationships, and secrets hidden by characters can further complicate this task (Zhao et al., 2023).

Character-centric Information Extraction Depending on various motivations, character-centric information extraction such as identifying motivations and emotional reactions (Rashkin et al., 2018; Kim and Klinger, 2019), roles (Stammbach et al., 2022), appearance (Zhao et al., 2023), or personalities (Flekova and Gurevych, 2015; Yu et al., 2023b) can be conducted from the perspectives of these identified characters. Relation extraction plays a key role in character-centric information extraction (Chang et al., 2009; Labatut and Bost, 2019). Previous research exploring the narrative comprehension and inferential abilities of LLMs has typically relied on question answering (Yang and Choi, 2019; Xu et al., 2022; Gandhi et al., 2023), which may include questions about relationships but do not cover all aspects of character relationships. Alternatively, some research has been conducted on extracting relationships directly from sentences (You and Goldwasser, 2020; Mellace et al., 2020), typically focusing on verbs, such as *say*, *smile*, *look*. Relation extraction is frequently integrated with other character-centric tasks like dialogue generation (Chen et al., 2023), summarisation (Brahman et al., 2021), and tracking the evolution of relationships between characters over time (Iyyer et al., 2016).

6 Conclusion and Future Directions

This research introduces a task for LLMs to comprehend complex character relationships in detective narratives, utilising our newly created *Conan* dataset. This dataset highlights the current challenges for LLMs and aims to improve their ability in narrative contexts.

Challenges (1). *Enhancing Inference Capabilities of LLMs.* Our research reveals that LLMs face challenges in deciphering complex relationships, particularly when the input narrative contains conflicting information. To address this, recent advancements (Wei et al., 2022; Yao et al., 2023; Wang et al., 2024) can be leveraged to augment the inference capabilities of LLMs. (2). *Optimising Key Information Identification.* We observed that LLMs struggle to pinpoint key information in lengthy inputs. Therefore, employing methods like cosine similarity (Park et al., 2023) or Retrieval-Augmented Generation (RAG)[4] to help LLMs retrieve and focus on the most relevant or crucial information could be beneficial.

Applications (1). *Enhancing Narrative Understanding.* Our work can be used to analyse complex narratives in literature, films, and video games. It helps in understanding character dynamics and plot development. (2). *Interactive Agents.* AI-driven agents are widely used in many sectors, including chatbots that cater for both professional and emotional needs (Qian et al., 2023; Tu et al., 2023), interactive game development (Gao and Emami, 2023; Zhao et al., 2023), and digital life simulation (Cai et al., 2023). Understanding the relationships between characters in user inputs can enhance conversation quality, making these systems more empathetic and context-aware. (3). *Theory of Mind.* Our dataset, built on various characters' perspectives, naturally includes insights into how characters perceive relationships and how they think others perceive them (Premack and Woodruff, 1978). This makes it well-suitable for theory of mind tasks (Wimmer and Perner, 1983; Onishi and Baillargeon, 2005).

[4] https://github.com/weaviate/Verba

Limitations

We identify some of the limitations below:

Annotator Influence Despite our efforts to design clear annotation guidelines through trial rounds, individual interpretations of relationships in *Conan* may still differ. This is because annotators' perceptions can influence how they see character dynamics, leading to some variations in how relationships are categories.

Limited Annotation Scope Due to the cost of manual annotation, we could not annotate all 100 narratives with human labelers. To address this, we employed the best-performing model and strategy for annotation, but this may not fully capture all the relationships in the given narratives.

Limited Evaluation of LLMs Our experiments were limited to a single run of each LLM, GPT-3.5, GPT-4, and LLaMa. They represent the cutting-edge models at the time. But due to the high cost, we could not run more extensive evaluations across a broader range of popular LLMs, potentially missing insights from newer or less-known models.

Ethics Statement

To respect copyright restrictions on the original Murder Mystery Game content, we follow established practices used in previous research (Frermann et al., 2018; Chen et al., 2023). Those interested in using the dataset will needs to obtain the original game narratives themselves. But we are sharing the content for which we hold the copyright, including annotations we created for the game's characters and their relationships; and content generated by LLMs based on those relationships. Additionally, we offer the code for preprocessing the original narratives to ensure others can replicate our experimental setup.

Please be aware that the detective narrative dataset may contain descriptions of violence, including violent events, actions, or characters. This content is provided solely for academic research and narrative analysis purposes. It is not intended to promote violence in any way. Users should be aware of potential exposure to violent content and use the dataset professionally and responsibly. This dataset is unsuitable for minors or those sensitive to violence.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) through a New Horizons grant (grant no. EP/X019063/1) and a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2).

References

Rami Aly, Andreas Vlachos, and Ryan McDonald. 2021. Leveraging type descriptions for zero-shot named entity recognition and classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, page 1516–1528, Online. Association for Computational Linguistics.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *CoRR*, abs/2309.12288.

Ellen Berscheid. 1994. Interpersonal relationships. *Annual Review of Psychology*, 45(1):79–129.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2022. Coreference resolution through a seq2seq transition-based system. *CoRR*, abs/2211.12142.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Zhongang Cai, Jianping Jiang, Zhongfei Qing, Xinying Guo, Mingyuan Zhang, Zhengyu Lin, Haiyi Mei, Chen Wei, Ruisi Wang, Wanqi Yin, Xiangyu Fan, Han Du, Liang Pan, Peng Gao, Zhitao Yang, Yang Gao, Jiaqi Li, Tianxiang Ren, Yukun Wei, Xiaogang Wang, Chen Change Loy, Lei Yang, and Ziwei Liu. 2023. Digital life project: Autonomous 3d characters with social intelligence. *CoRR*, abs/2312.04547.

Jonathan Chang, Jordan Boyd-Graber, and David M Blei. 2009. Connections between the lines: Augmenting social networks with text. In *Proceedings*