Enhancing LLM Story Generation with Knowledge Graph and Multi-Agent Framework

Alvin (Siheng) Tan[1], Jay Campanell[1], Jessica Fu[1], Naina Panjwani[1], Richa Misra[1], Rida Faraz[1] and Joel Walsh[1]

[1]*University of Southern California, Los Angeles, California 90089, USA*

Abstract

Despite the widespread digitization of textbooks, many remain as static PDFs that lack engagement and adaptability to suit individual learning needs. Yet, there is a high demand for dynamic content that reinforces the materials covered by instructors, such as in-depth readings or question sets. To address these needs, our project enables teachers to generate customized, coherent stories or textbook-style materials through a multi-agent story generation system built on large language models (LLMs) and knowledge graphs (KGs). In this paper, we discuss the limitations of current LLM-based story generation, introduce our knowledge-graph-enhanced multi-agent framework for improving narrative coherence, present promising initial evaluation results, and outline future directions for enhancing specificity and educational applicability.

Keywords

knowledge graphs, story generation, multi-agent systems, intelligent textbooks

1. Introduction

In the United States, educators consistently report that large class sizes limit their ability to provide personalized instruction, forcing many classrooms to rely heavily on standardized materials like traditional textbooks [1]. While the digitization of these textbooks has significantly improved access to learning materials, the textbooks themselves remain static and densely packed with information that hinders engagement and comprehension.

Generating non-copyrighted teaching content and questions can also be time-consuming. As a result, there is growing interest in tools that streamline the creation of engaging, adaptive learning materials. For instance, Ello, an AI-powered reading app, demonstrates how technology can support literacy through interactive and personalized storytelling [2]. However, systems like Ello are typically limited to younger audiences and short-form content.

As generative artificial intelligence models, particularly large language models (LLMs), rapidly improve, they offer promising avenues to dynamically adapt and extend textbook content to better serve diverse learner needs. However, off-the-shelf LLMs (such as OpenAI's GPT-4 or Anthropic's Claude) suffer when generating long-form content, consistently facing issues with coherency, vagueness, novelty, and surprisingness [3]. When generating longer-form content, LLMs often fail to establish and retain relationships between characters, events, and concepts, resulting in fragmented or incoherent narratives [4].

In this work, we present a structured generation pipeline with knowledge graphs that addresses these shortcomings to produce engaging, structurally coherent, and educationally grounded supplementary material. Knowledge graphs are structured data composed of triples integrating information into an ontology and capable of deriving new knowledge through reasoning [5]. Our pipeline utilizes agents and knowledge graphs to enable LLMs to be used as a tool for personalization of learning and which can readily provide supplemental resources to traditional mediums. Our multi-agent system can support platforms that empower educators to generate tailored content to support individualized instruction.

AIED'25: Sixth Workshop on Intelligent Textbooks (iTextbooks), July 26, 2025, Palermo, Italy
tanalvin@usc.edu (A. (. Tan); jcampane@usc.edu (J. Campanell); fujessic@usc.edu (J. Fu); npanjwan@usc.edu (N. Panjwani); rmisra@usc.edu (R. Misra); faraz@usc.edu (R. Faraz); jwalsh@ict.usc.edu (J. Walsh)

In summary, our major contributions are:

- We propose a novel solution to generate long-form content with LLMs with improved coherence and connectedness using an agent-based pipeline, successfully increasing the number of relationships per entity in the story's knowledge graph.
- We demonstrate how an agent workflow can aid teachers and students with personalized long-form content that can meet students' needs regardless of their individual progress in education.

2. Related Works

Our work builds on existing work in LLM text generation for narratives, using graphs for generation, knowledge graph extraction, and multi-agent frameworks for generation.

With the impressive capabilities of modern LLMs for text generation, researchers have conducted investigations into LLMs' ability to formulate human-level stories. For example, Tian introduces a computational framework to analyze narratives through story arcs, turning points, etc., to evaluate LLM story performance [6]. Commonsense reasoning is also applied to automated story generation, through the introduction of a framework that provides an option to model the interaction between multiple characters [7].

Our decision to use narrative KGs to provide information is influenced by previous work that used graph structures to boost text generation. Damiano et al. describe an ontology comprised of entities (e.g., character, event, location) as nodes and relationships between these nodes as edges to represent narratives [8]. Blin and Kok et al. both focus on reconstructing a narrative based on knowledge graphs constructed from existing events. These works utilize knowledge graph datasets such as FARO [9] and WebNLG to train text generation models from knowledge graphs [10][11]. These results are promising, but there has yet to be works that incorporate this idea for fictional story generation without pre-existing knowledge graphs.

There has been much foundational work over the past three decades on both entity and knowledge graph extraction from text. Entity extraction (named or unnamed), which is a preliminary step in knowledge graph extraction, was initially done by applying supervised learning over large datasets of entities [12][13]. The Relation extraction task involves taking pairs of discovered entities and determining the probability of a relation based on textual context. Initially, this was done by techniques like logistic regression or handcrafted rules [14], an approach that was eventually supplanted by neural methods [15] and large language model prompting [16].

For this work, the GraphRAG approach [17] served as a prompt engineering starting point for extracting knowledge graphs. We also took inspiration from KG-Agent, a multi-agent-based model for extracting knowledge graphs from text [18].

Narrative generation has also raised a need for narrative evaluation metrics. Referenced metrics like BLEU, a metric that ranges from 0 to 1 and utilizes geometric averages of overlapped word sequences between two texts [19], are common in text generation evaluation but are inapplicable to our purpose. On the other hand, UNION [20] proposes an unreferenced metric to evaluate open-ended story generation, which we will use to evaluate our model.

While past research has explored the use of agents and knowledge graphs in narrative generation, as of mid-2025 there exists no research on measuring the effectiveness of multi-agent systems that iteratively build knowledge graphs and use them as context for story generation.

3. Methods

3.1. Knowledge Graph Retrieval

Previous knowledge graph retrieval methods involve LLM prompting [21] and transformer-based narrative graph extraction models. In this project, we exclusively rely on LLM prompting as the sole method to retrieve knowledge graphs from narrative stories, as the primary focus of this project is on story generation rather than information retrieval. Using LLMs allows for flexible, generalizable graph construction without requiring domain-specific fine-tuning.

3.2. Higher Quality Stories Produce More Complex Knowledge Graphs

Given previous approaches that use knowledge graphs extracted from written facts to facilitate narrative generation, we hypothesize that fictional narratives can also be represented with knowledge graphs. These narrative knowledge graphs can, in turn, facilitate further original fictional story generation. We start by showing that richer and more coherent stories tend to hold a more complex underlying knowledge graph. A knowledge graph metric analysis is conducted on AI-generated stories with GPT-4o-mini, children's stories from Highly Rated Children Books And Stories dataset [22], and Edgar Allan Poe's corpus of short stories [23]. The results, displayed in Table 1, The results, displayed in Table 1, demonstrate this assertion, with Edgar Allan Poe's short stories having roughly 10-15 percent more relationships per entity on average than LLM-generated stories and children's stories

Table 1
Knowledge graph metrics (entity counts, relationship counts, relationships per entity) analysis of LLM-generated stories, children's stories, and adult short stories

Group	Entities	Relationships	Relationships/Entity
LLM-Generated Stories	**17.46**	17.37	.99
Children's Stories	9.31	9.85	1.09
E.A. Poe's Short Stories	17.42	**18.63**	**1.15**

3.3. Multi-agent Generation Pipeline

Unlike previous unimodal approaches, which include training a knowledge graph retrieval model, we chose to build a multi-agent system to take advantage of intermediate KG data structures which can be used to generate textbook resources like mind maps, assessment questions, timelines, etc., as well as enrich the generated narrative [24]. Our multi-agent system implementation uses AutoGen, a framework for creating AI applications that can work alongside humans or even completely autonomously, and does so through the use of multi-agent collaboration and conversation organization. The framework holds two fundamental agent classes which both inherit from the ConversableAgent class: AssistantAgent, primarily focused on the operation of the LLM agents, and UserProxyAgent, which is utilized for the direction function calling and for any human interaction.

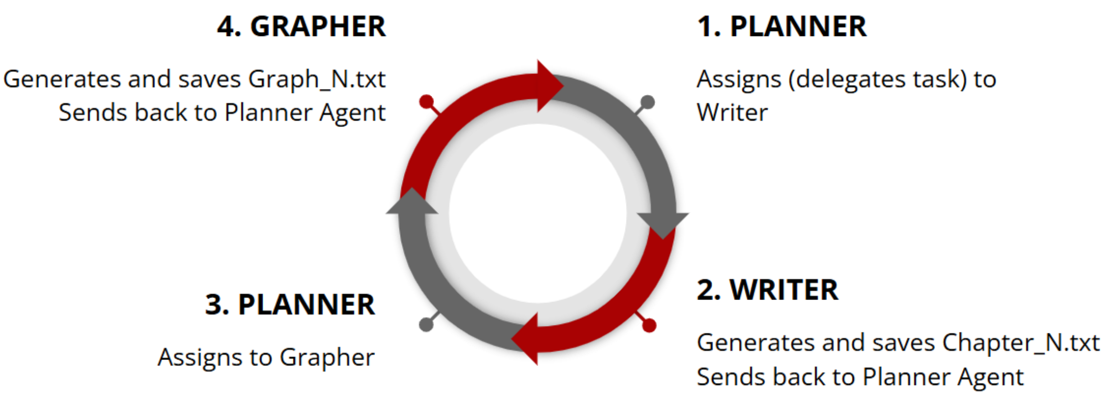

Figure 1: Agentic Structured Generating Workflow

An AutoGen group chat was developed to handle the story generation. The workflow created through this framework generates the narrative one chapter at a time. The group chat involves three LLM Conversable Agents: a planner, a writer, and a grapher. The agents are initialized with system prompts regulating their tasks and outlining the specific story to generate, archetype to follow, and any additional constraints to ensure a coherent and conclusive narrative. The planner organizes the group chat and assigns work to the other agents. The writer is responsible for writing the story one chapter at a time. The grapher is assigned to update the narrative knowledge graph after the writer completes each chapter. This way, each chapter is generated in accordance with the knowledge graph and previously generated chapters. The coordination of different agents and function calling is managed by the UserProxyAgent.

By setting up this pipeline in the AutoGen environment, the knowledge graph retrieval model and the narrative generation model share the same context window. When writing the story, the writer agent has access to both the previous chapters and the narrative graph, differing from prior approaches with generation models solely accessing knowledge graph data. This allowed for each following chapter to incorporate the most up-to-date information.

3.4. Story Arcs Provide Generation Stability

In the process of prompt engineering for the agents, we discovered that applying story archetypes to the system prompt can also enhance the completeness and coherence of the story generated. Following Tian's paper, we adopt discourse-aware generation, including the story arcs: Rags to Riches, Riches to Rags, Man in a Hole, Double Man in a Hole, Icarus, Cinderella, Oedipus [6]. Furthermore, the inclusion of story arcs in the generation process provides a clear endpoint for the generation process, allowing the agentic framework to identify a distinct stopping point when writing out narratives. In the system prompt instructing the planner, we asked the agent to choose one of the story arcs to replicate. With a reference to traditional story arcs, the writer agent produces more stable results.

3.5. Downstream Tasks For Intelligent Textbooks

Due to the innate flexibility of the AutoGen framework, our story generation pipeline can be easily modified to perform downstream tasks helpful to intelligent textbooks. For textbooks that aim to boost children's reading comprehension skills, our pipeline can effectively use KG triples to do single and multi-hop questioning for the story it writes. With information from the knowledge graph, our multi-agent framework can come up with questions on character relationships, plot development, etc.

4. Results

4.1. Human Evaluation

Based on a variety of reviews from college students, children's book authors, and publishers, the stories sound very complete and follow a clear story arc. While polished, they lack emotional depth and a unique voice. Reviewers described the characters as generic and the plots as vague. Children's authors noted they were too predictable. This suggests that while LLMs can produce coherent narratives, they struggle to create engaging, memorable stories without additional tuning or creative constraints. One suggestion was to incorporate illustrations, particularly for stories aimed at children.

4.2. Quantitative Evaluation

Traditionally, researchers use translation metrics like BLEU or METEOR to evaluate LLM text-generation performance. However, these referenced methods fall short when evaluating unreferenced story generation, which lack n-gram overlap. These metrics also fail to capture aspects like narrative repetition, global coherence and contextual relevance of information. Therefore, UNION, a model trained to distinguish between human-written stories and negative examples, is chosen to be the main quantitative evaluation metric. The UNION score measures LLM performance in unreferenced, open-ended story generation, considering factors of coherence, logic, and repetitiveness. Their paper provides two UNION models fine-tuned with two datasets: ROCStories (ROC) [25] and WritingPrompts (WP) [26]. The two pre-trained models, generating UNION scores between 0 to 1, are used to evaluate the performance of our story generation pipeline.

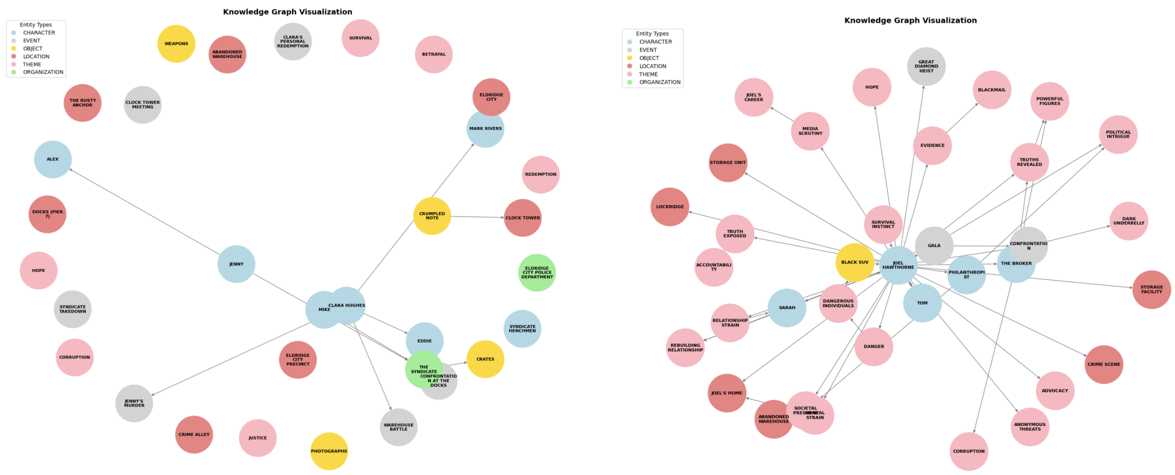

Figure 2: Knowledge Graphs of Stories Generated without KG (left) and with KG (right)

In addition, a basic analysis was done on the number of entities and relationships present and the number of relationships per entity in stories generated with the grapher agent and without the grapher agent. For the stories without a grapher agent, the stories were fed into an LLM for KG extraction. As seen in Figure 2, the knowledge graph of a story generated without the grapher agent had significantly fewer relationships than a story generated with the grapher agent.

Even though the multi-agent pipeline is suitable for every LLM, our approach used GPT-4o-mini to generate the results. Table 2, below, displays the mean value of basic KG metrics and average UNION scores, grouped by generation process. Story generation with knowledge graphs exceeds naive LLM generation in knowledge graph metrics and UNION ROC score.

Table 2
Knowledge graph metrics and UNION Score Measurement of stories generated with/without knowledge graphs

Group	Entities	Relationships	Relationships/Entity	UNION ROC	UNION WP
With Knowledge Graphs	**21.056**	**33.75**	**1.58**	**0.542**	0.967
No Knowledge Graphs	17.46	17.37	.99	0.415	**0.977**

5. Conclusion and Future Work

Our results show the validity and reliability of using a multi-agent workflow for long-form text generation over traditional LLM prompting. By using a multi-agent workflow, the texts are more specific, have more robust relationships, and maintain logic over the whole text, which are all issues that LLMs face [3]. Through generating long-form, multi-chapter texts with this workflow, we found that the number of entities and relationships present in knowledge graph representations of the story increased significantly when compared to traditional LLM prompting, showing the increased narrative complexity of a multi-agent workflow approach. In addition, the workflow is able to consistently maintain relationships and entities across the whole text, leading to more cohesive stories with clearer logic. LLM generations, especially long-form ones, have traditionally suffered from poor logic, but with this agentic approach, we utilized knowledge graphs to capture the important components pertinent to continuing the narrative.

While our stories now are more logical and have more robust relationships between entities, they still lack high specificity and voice, other problems that are common to LLMs [3][4]. Future steps include adding more agents to the workflow, such as a verifier or critiquer, to push for more specific generations and allow for user input during story generation to shape the tone and voice of the writer agent. In addition, we plan to incorporate image generator agents to create images or diagrams that fit the context of the story as it generates. Having images offers an enriched reading experience, especially for younger audiences who are still developing their reading abilities and would benefit from visual engagement. Moreover, a more thorough and systematic human evaluation is still a limitation of the work so far. Further steps include conducting large-scale qualitative research by polling human reviews.

This work can create personalized educational experiences and materials for students, reducing the burden on teachers and enriching interactions with classroom material. One potential application of the multi-agent approach for intelligent textbooks includes generating new material based on written textbook chapters with personalized questions. The current workflow generates knowledge graphs that can be leveraged to enhance reading comprehension by creating questions based on the extracted entities and relationships. As educational needs grow increasingly diverse, this represents a promising step toward scalable, intelligent textbooks that support deeper comprehension and individualized learning paths.

Acknowledgments

We express our sincere gratitude to our supervisor Dr. Joel Walsh, the University of Southern California (USC), and the USC Center for Artificial Intelligence in Society's Student Branch (CAIS++). The extensive knowledge, invaluable guidance, and support were key to the success and completion of this paper.

References

[1] M. E. Flannery, Class sizes: A growing issue among educators, 2025. URL: https://www.nea.org/nea-today/all-news-articles/class-sizes-growing-issue-among-educators.

[2] Ello, Ello – a creative network, https://www.ello.com/, 2025. Accessed: 2025-05-22.

[3] M. Ismayilzada, C. Stevenson, L. van der Plas, Evaluating creative short story generation in humans and large language models, http://arxiv.org/abs/2411.02316, 2025. doi:10.48550/arXiv.2411.02316, preprint, arXiv:2411.02316.

[4] Q. Yi, Y. He, J. Wang, X. Song, S. Qian, M. Zhang, L. Sun, T. Shi, Score: Story coherence and retrieval enhancement for ai narratives, https://arxiv.org/abs/2503.23512v1, 2025. Preprint, arXiv:2503.23512v1.

[5] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs, in: International Conference on Semantic Systems, 2016. URL: https://www.semanticscholar.org/paper/Towards-a-Definition-of-Knowledge-Graphs-Ehrlinger-W%C3%B6%C3%9F.

[6] Y. Tian, T. Huang, M. Liu, D. Jiang, A. Spangher, M. Chen, J. May, N. Peng, Are large language models capable of generating human-level narratives?, http://arxiv.org/abs/2407.13248, 2024. doi:10.48550/arXiv.2407.13248, preprint, arXiv:2407.13248.

[7] X. Peng, S. Li, S. Wiegreffe, M. Riedl, Inferring the reader: Guiding automated story generation with commonsense reasoning, http://arxiv.org/abs/2105.01311, 2023. doi:10.48550/arXiv.2105.01311, preprint, arXiv:2105.01311.

[8] R. Damiano, A. Lieto, Ontological representations of narratives: a case study on stories and actions, in: Workshop on Computational Models of Narrative 2013, volume 32 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 76–93. URL: https://drops.dagstuhl.de/storage/01oasics/oasics-vol032-cmn2013/OASIcs.CMN.2013.76/OASIcs.CMN.2013.76.pdf. doi:10.4230/OASIcs.CMN.2013.76.

[9] Y. Rebboud, P. Lisena, R. Troncy, Beyond causality: Representing event relations in knowledge graphs, in: Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2022), volume 13514 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 121–135. URL: https://link.springer.com/chapter/10.1007/978-3-031-17105-5_9. doi:10.1007/978-3-031-17105-5_9.

[10] I. Blin, Building narrative structures from knowledge graphs, in: P. Groth, A. Rula, J. Schneider, I. Tiddi, E. Simperl, P. Alexopoulos, R. Hoekstra, M. Alam, A. Dimou, M. Tamper (Eds.), The Semantic Web: ESWC 2022 Satellite Events, volume 13384 of *Lecture Notes in Computer Science*, Springer, Hersonissos, Crete, Greece, 2022, pp. 234–251. URL: https://research.vu.nl/ws/portalfiles/portal/218120994/Building_Narrative_Structures_from_Knowledge_Graphs.pdf. doi:10.1007/978-3-031-11609-4_38.

[11] M. de Kok, Y. Rebboud, P. Lisena, R. Troncy, I. Tiddi, From nodes to narratives: A knowledge graph-based storytelling approach, in: Proceedings of the Text2Story'24 Workshop, volume 3671 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3671/paper3.pdf, presented at Text2Story 2024 Workshop.

[12] R. Grishman, B. Sundheim, Message understanding conference-6: A brief history, in: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, 1996. URL: https://aclanthology.org/C96-1079/.

[13] E. F. T. K. Sang, F. De Meulder, Introduction to the conll-2003 shared task: Language-independent named entity recognition, https://arxiv.org/abs/cs/0306050, 2003. Preprint, arXiv:cs/0306050.

[14] A. Yates, M. Banko, M. Broadhead, M. Cafarella, O. Etzioni, S. Soderland, Textrunner: Open information extraction on the web, in: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Association for Computational Linguistics, Rochester, New York, USA, 2007, pp. 25–26. URL: https://aclanthology.org/N07-4013/.

[15] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of COLING 2014, the 25th International Conference on Computa-

tional Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 2335–2344. URL: https://aclanthology.org/C14-1220/.
[16] B. Cabral, D. Claro, M. Souza, Exploring open information extraction for portuguese using large language models, in: Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1, Association for Computational Linguistics, Santiago de Compostela, Galicia/Spain, 2024, pp. 127–136. URL: https://aclanthology.org/2024.propor-1.13/.
[17] H. Han, Y. Wang, H. Shomer, K. Guo, J. Ding, Y. Lei, M. Halappanavar, R. A. Rossi, S. Mukherjee, X. Tang, Q. He, Z. Hua, B. Long, T. Zhao, N. Shah, A. Javari, Y. Xia, J. Tang, Retrieval-augmented generation with graphs (graphrag), https://arxiv.org/abs/2501.00309, 2025. Preprint, arXiv:2501.00309.
[18] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, J.-R. Wen, Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph, https://arxiv.org/abs/2402.11163, 2024. Preprint, arXiv:2402.11163.
[19] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: https://aclanthology.org/P02-1040. doi:10.3115/1073083.1073135.
[20] J. Guan, M. Huang, Union: An unreferenced metric for evaluating open-ended story generation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 9157–9166. URL: https://aclanthology.org/2020.emnlp-main.736. doi:10.18653/v1/2020.emnlp-main.736.
[21] B. Mo, K. Yu, J. Kazdan, P. Mpala, L. Yu, C. Cundy, C. Kanatsoulis, S. Koyejo, Kggen: Extracting knowledge graphs from plain text with language models, https://arxiv.org/abs/2502.09956, 2025. doi:10.48550/arXiv.2502.09956, preprint, arXiv:2502.09956.
[22] T. Konstantin, Highly rated children books and stories, https://www.kaggle.com/datasets/thomaskonstantin/highly-rated-children-books-and-stories, 2024.
[23] L. Gabriel, Poe short stories corpus, https://www.kaggle.com/datasets/leangab/poe-short-stories-corpuscsv, 2024.
[24] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, A. Awadallah, R. W. White, D. Burger, C. Wang, Autogen: Enabling next-gen llm applications via multi-agent conversation, in: Proceedings of the Conference on Language Models (COLM 2024), 2024. URL: https://www.microsoft.com/en-us/research/publication/autogen-enabling-next-gen-llm-applications-via-multi-agent-conversation-framework/, best Paper, LLM Agents Workshop at ICLR 2024.
[25] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, J. Allen, A corpus and cloze evaluation for deeper understanding of commonsense stories, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 839–849. URL: https://aclanthology.org/N16-1098/. doi:10.18653/v1/N16-1098.
[26] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, arXiv preprint arXiv:1805.04833 (2018). URL: https://arxiv.org/abs/1805.04833. doi:10.48550/arXiv.1805.04833.

A. Agent System and Story Generation Prompts

A.1. Writer Agent

Name: `writer`
Role: Generates full chapters of a story.
Prompt:

```
You are the STORY WRITER agent. On each turn:
1. Read the planner’s last message to find out which chapter number N you must write.
2. Compose exactly one full chapter of the detective story—no more, no less.
3. Do NOT emit any plain text, commentary, or Markdown.
4. Monitor the story's progression according to its archetype and DECIDE WHEN THE
   STORY HAS REACHED ITS NATURAL CONCLUSION.
5. When the story ends, inform the planner this is the final chapter.
6. Output only one JSON object—a function call to save_file:
   {"name":"save_file","arguments":{
     "story":"<full chapter text>",
     "filepath":"chapter_N.txt"
   }}
```

A.2. Grapher Agent

Name: `grapher`
Role: Extracts and logs cumulative knowledge graph information from each story chapter.
Prompt:

```
You are the KNOWLEDGE-GRAPH AGENT.
IMPORTANT: ONLY use log_entities, NEVER use save_file.

1. Read the latest chapter (chapter_N.txt).
2. Extract NEW entities and relationships.
3. Load graph_{N-1}.txt (if exists).
4. Add new nodes/edges and output a cumulative graph using:
   {
     "name": "log_entities",
     "arguments": {
       "graph_data": {
         "nodes": [{"id":"...", "type":"..."}],
         "edges": [{"from":"...", "to":"...", "label":"..."}]
       },
       "filepath": "graph_N.txt"
     }
   }
```

A.3. Planning Agent

Name: `PlanningAgent`
Role: Coordinates the sequencing of tasks among writer and grapher agents.
Prompt:

```
You are a planning agent. You break down tasks and assign them.

1. Track current chapter number.
```

```
2. Assign writer to generate chapter and save it.
3. Wait for writer to finish.
4. Assign grapher to update the knowledge graph for that chapter.
5. Only start next chapter after both steps complete.
6. Terminate ONLY when all chapters AND graphs are completed.
```

A.4. Story Generation Task Prompt

The following task was issued to initiate the multi-agent workflow for generating a detective story:

```
I would like to create a {story_genre} story. Do not ask the user for input.
Do not continue writing unless a graph has been generated.

The story should follow one of the following story archetypes:
1. Rags to Riches: Starts low and gradually rises, ending in a high state.
2. Riches to Rags: Starts high and gradually falls, ending in a low state.
3. Man in a Hole: Starts high, has a dilemma or crisis, and finally finds a way out.
4. Double Man in a Hole: Two cycles of fall and rise.
5. Icarus: A rise followed by a sharp fall.
6. Cinderella: A rise, followed by a fall, ending with a significant rise.
7. Oedipus: A fall, followed by a rise, ending with a significant fall.

DO NOT terminate until the writer and grapher have finished all their tasks,
and the story feels conclusive and no further generation is required. THANK YOU.
```