Remember Me: Enhancing Long-term Character Persistence in LLM-Generated Stories using Knowledge Graphs

Alvin Tan[1], **Jay Campanell**[1], **Richa Misra**[1], **Naina Panjwani**[1], **Jessica Fu**[1], **Rida Faraz**[1], **Joel Walsh**[2]
[1]Department of Computer Science, University of Southern California, LA, CA 90089, USA
[2]Department of Computer Science, Occidental College, LA, CA 90041, USA
{tanalvin, jcampane, rmisra, npanjwan, fujessic, faraz}@usc.edu
jwalsh2@oxy.edu

Abstract

Despite advances in the instruction-following capabilities of LLMs, long-form content generated by state-of-the-art models still lacks coherence and consistency. Previous work in Question Answering and factual text generation has leveraged knowledge graphs (KG)' representation capabilities, but there have been limited efforts to incorporate KGs in creative tasks. To improve LLMs' zero-shot performance in narrative generation, we propose a multi-agent story generation pipeline with knowledge graphs as long-term context storage to improve character and relationship consistency in long fiction. To this end, we designed and tested a pipeline that can generate long narratives chapter by chapter with LLM agents responsible for generation planning, knowledge graph extraction, and next chapter generation. We also created a comprehensive list of starting prompts to evaluate the pipeline's performance across genres. In order to evaluate the pipeline quantitatively, we devised a novel retention-based story consistency metric due to a gap in effective quantitative metrics for unreferenced text generation. The character retention metrics and relationship retention metrics reflect a narrative's long-term consistency across chapters. Through a statistical analysis of stories generated with the multi-agent KG pipeline and those generated with naive prompting, we found significant improvement in retention and cohesion in stories generated with KG guidance.

Introduction

As large language models (LLMs) become well-trained experts in question-answering and code-writing, people are exploring applications of LLMs in the creativity space. For example, research has been done on enhancing human-AI co-writing experience (Ocampo, Bown, and Grace 2025). Content creators are using LLMs to assist them in ideation and scriptwriting to boost efficiency. In addition, LLMs are commonly used in the education sphere to create personalized learning experiences for students. For instance, Ello is a company that applies LLM for personalized content creation and reading assistance for children (Ello 2025). As more and more people are trying to leverage the computational creativity of LLMs in fields like entertainment, literature, and education, there is a growing demand for systems that can generate dynamic, coherent, and pedagogically grounded narratives.

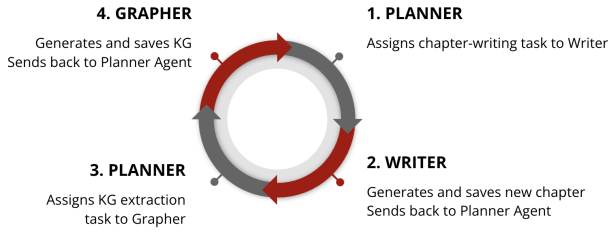

Figure 1: Agentic Structured Generation Workflow

Recent advances in large language models (LLMs) have demonstrated strong capabilities in text generation. However, off-the-shelf LLMs (such as OpenAI's ChatGPT or Anthropic's Claude) struggle with long-form generation, frequently exhibiting issues with narrative drift, character inconsistency, relationship loss, and logical breakdowns across chapters. These limitations are particularly problematic for personalized-learning experiences, where cohesion, continuity, and conceptual consistency are essential for comprehension and learning (Ismayilzada, Stevenson, and van der Plas 2024). Prior work has shown that LLM-generated stories often lack sustained relationships between entities and fail to preserve implicit connections over time, resulting in fragmented narratives (Yi et al. 2025).

Knowledge graphs (KGs) offer a promising structural representation for addressing these challenges. By explicitly modeling entities and their relationships, KGs provide a mechanism for tracking narrative state and enforcing consistency. Previous research has explored the use of graphs for narrative reconstruction and text generation, as well as automated KG extraction from text (Ehrlinger and Wöß 2016). Separately, multi-agent frameworks have emerged as effective tools for decomposing complex generation tasks into coordinated subtasks. However, existing work has largely treated knowledge graph extraction, story generation, and evaluation as isolated components. To our knowledge, there is no prior work that systematically measures how an agentic pipeline that iteratively builds and conditions on narrative knowledge graphs impacts long-form story cohesion and en-

tity retention.

In this work, we present a multi-agent, graph-structured generation framework for long-form story creation using LLMs. Built using LangGraph, a framework specialized towards LLM multi-agent applications, our system decomposes the generation into specialized agents—a planner, writer, and grapher—that collaboratively generate stories chapter by chapter while maintaining a shared, evolving narrative knowledge graph (LangChain 2026). Unlike prior approaches that rely solely on post-hoc graph extraction or KG-only conditioning, our framework enables the writer agent to directly condition on both previously generated text and the current knowledge graph state, allowing for explicit preservation of characters, relationships, and narrative objectives over time.

Beyond generation, we introduce a more targeted evaluation methodology that moves beyond surface-level text similarity metrics. We propose entity- and relationship-centric metrics grounded in named entity recognition (NER) to measure character retention, relationship persistence, and narrative stability across chapters (Pakhale 2023). These metrics allow us to directly quantify the benefits of knowledge-graph-guided generation in ways that align with human judgments of cohesion, defined as a quality that measures how well characters, relationships, and plot connect within stories by Graesser et al. (2004) and logical consistency.

Our contributions are threefold:

1. We introduce a LangGraph-based multi-agent framework that integrates iterative knowledge graph construction directly into long-form LLM story generation.[1]
2. We demonstrate that conditioning generation on evolving narrative knowledge graphs significantly improves character retention, relationship persistence, and structural cohesion across chapters.
3. We propose and validate new evaluation metrics for long-form narrative generation that explicitly measure entity and relationship continuity, complementing existing unreferenced story quality metrics.

Related Work

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

With the impressive capabilities of modern LLMs for text generation, researchers have conducted investigations into the ability of LLMs 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 generation performance (Tian et al. 2024). Common sense 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 (Peng et al. 2023).

[1]Our code is available at: `https://github.com/usc-caisplusplus/kg-story-persistence`

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 composed of entities (e.g., character, event, location) as nodes and relationships between these nodes as edges to represent narratives (Damiano and Lieto 2013). Blin (2022) and Kok et al. (2024) both focus on reconstructing a narrative based on knowledge graphs constructed from existing events. These works utilize knowledge graph datasets such as FARO (Rebboud, Lisena, and Troncy 2022) and WebNLG to train text generation models from knowledge graphs (Blin 2022; de Kok et al. 2024). These results are promising, but there has yet to be work that incorporates this idea for fictional story generation without pre-existing knowledge graphs.

Particularly in research presented on computational creativity, structured, knowledge-driven frameworks have been used to approach narrative generation. Early systems depend on symbolic representations and unique narrative planning to guarantee causal cohesion and a global plot structure (Riedl and Young 2010). Other foundational systems, like Mexica, emphasized creativity as a process driven by internal narrative knowledge rather than surface-level text generation by modeling stories with explicit representations of characters, emotions, and events (Pérez y Pérez and Sharples 2001; Gervás 2009). More computational creativity work has increasingly explored hybrid approaches which are neural symbolic approaches that combine language models with explicit knowledge structures to guide creative output (Veale, Cardoso, and y Pérez 2019). This line of research provides conceptual grounding for our use of narrative knowledge graphs as an explicit representation to guide LLM-based story generation, positioning our approach as a continuation of long-standing computational creativity perspectives adapted to modern generative models.

Many studies conducted within the computational creativity community have examined how explicit narrative theories and structures can direct computational story generation systems. Pickering and Jordanous demonstrate how high-level narratological concepts can methodically shape plot developments rather than leaving them to unrestricted text generation. Narrative theory is used to engineer unexpectedness in a story generator (Pickering and Jordanous 2017). In order to show how structured representations of actions and intentions can support cohesive plots like ”kill the dragon, rescue the princess,” Laclaustra et al. (2014) frame story generation as a plan-based, multi-agent process in which characters pursue goals within an explicitly modeled narrative world. Moreover, in order to choose and polish narrative content while keeping an eye on the underlying story structure, Méndez and Gervás (2023) incorporate ChatGPT into a story-sifting pipeline using a sizable language model. This line of research provides conceptual grounding for our use of narrative knowledge graphs as an explicit representation to guide LLM-based story generation, positioning our approach as a continuation of long-standing computational creativity perspectives adapted to modern generative models.

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 (Grishman and Sundheim 1996; Sang and De Meulder 2003). The Relation extraction task involves taking pairs of discovered entities and determining the probability of a relation based on the textual context. Initially, this was done by techniques like logistic regression or handcrafted rules (Yates et al. 2007), an approach that was eventually supplanted by neural methods (Zeng et al. 2014) and large language model prompting (Cabral, Claro, and Souza 2024). For this work, the GraphRAG approach (Han et al. 2025) 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 (Jiang et al. 2024).

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 (Papineni et al. 2002), are common in text generation evaluation but are inapplicable to our purpose. On the other hand, UNION proposes an unreferenced metric to evaluate open-ended story generation (Guan and Huang 2020).

Work has been released detailing different pipelines for story generation with knowledge graph enhancement. In Pan et al., the authors propose a pipeline that relies on counts of previous context and scenes to determine whether or not to move forward with the generation and make use of a small sample size of human annotators (Pan et al. 2025). Building on prior research in knowledge graph systems, our work introduces an agentic and iterative framework that dynamically extracts from an evolving knowledge graph during generation. This approach enables structured long-term entity and relationship tracking in LLMs that offers new empirical insight.

Methodology

Knowledge graph retrieval

Prior work on narrative knowledge graph extraction has explored transformer-based models and LLM prompting methods. In this work, we rely exclusively on LLM-based prompting to construct knowledge graphs from narrative text, prioritizing flexibility and generalization over domain-specific extraction accuracy. This choice allows our system to operate across open-ended fictional narratives without requiring labeled data or fine-tuning.

Knowledge graph construction is performed incrementally during generation. After each chapter is produced, entities and relationships are extracted and used to update a shared narrative knowledge graph representing characters, locations, events, and their relations. The entity extraction process targets these specific literary components of the story: character, event, object, location, organization and theme. The relationship extraction process discovers and stores short-form text representations of the relationships between extracted entities, saved as edges in the knowledge graph. This evolving graph is maintained as part of the system state and made available to subsequent generation steps.

By explicitly tracking narrative structure through an evolving knowledge graph, the system preserves contextual information across chapters and enables direct measurement of character and relationship retention in long-form story generation.

Multi-agent generation pipeline

We implement our multi-agent pipeline using LangGraph, a graph-based framework for orchestrating multi-agent LLM workflows with explicit state propagation and control flow (LangChain 2026). Rather than relying on a single generation step or post-hoc processing, our approach decomposes story generation into coordinated agents that operate over a shared state, enabling explicit control over narrative structure and memory.

The pipeline consists of three specialized agents: a writer, a grapher, and a planner. Story generation proceeds iteratively on a chapter-by-chapter basis. The writer agent generates a new chapter conditioned on both the preceding chapters and a structured knowledge graph context derived from the current graph store. After each chapter is produced, the grapher agent extracts entities and relationships from the new text and inserts them into an evolving knowledge graph structure. The planner then increments the chapter counter and resets agent completion flags to prepare the next iteration.

LangGraph represents this workflow as a directed graph with three nodes: writer, grapher, and planner. After the writer node executes, a router function inspects the shared story state to determine whether to terminate the story at this current chapter.

By persisting entities and relationships in a knowledge graph structure across iterations and surfacing chapter-of-introduction, milestone chapters, and recent chapters as structured context blocks in the writer's prompt, the pipeline enables consistent character tracking, relationship preservation, and long-term narrative cohesion across chapters, addressing limitations of unimodal LLM prompting for long-form text generation.

Story Arcs

We noticed that generations with the same prompt set often lead to similar results. To increase variance in the content generated to better understand the pipeline's performance, we incorporated story arcs in addition to a curated list of starting prompts. 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 (Tian et al. 2024). 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, the agent is prompted to choose one of the story arcs to replicate. With a reference to traditional story arcs, the pipeline is able to generate more diverse results based on the prompt.

Results

Traditionally, researchers use translation metrics like BLEU or METEOR to evaluate LLM text-generation performance (Papineni et al. 2002; Lavie and Agarwal 2007). However, these referenced methods fall short when evaluating unreferenced story generation, which lacks n-gram overlap. UNION, a model trained to distinguish between human-written stories and negative examples, has been devised to produce quantitative scores to evaluate unreferenced story generation. The authors automatically perturb human-written stories using four techniques (repetition, substitution, reordering, and negation alteration) to simulate common errors observed in natural language models, such as repeated plots, incoherent content, conflicting logic, and flipped semantics. At inference time, UNION takes a generated story as input and outputs the probability that it is human-written, serving as a reference-free quality score. Nevertheless, since UNION is based on BERT encoders and trained on short story datasets like ROC Stories, the model is unable to serve our goal of evaluating multi-chapter long stories with more than 4000 words on average. (Guan and Huang 2020)

We have also experimented with LLM-as-judge for evaluation. However, LLMs are generally inaccurate in giving numerical ratings that are reflective of story quality. Also, as reported by Wang et al. (2025), LLMs tend to have favoritism when judging LLM-generated stories, producing unwanted bias.

Retention Analysis

While existing measures fail to evaluate unreferenced multi-chapter story generation comprehensively, we seek to evaluate the pipeline by focusing on the effect of character and relationship retention. Since one of the major purposes of our pipeline is to enhance long-term cohesion throughout the generated story, the measurement of how characters and relationships are preserved across chapters is reflective of the pipeline's performance. We conduct retention calculations for characters and relationships in each chapter in three ways: chapter 1 retention, rolling retention, and cumulative retention. These metrics are calculated on a chapter-by-chapter basis. The characters and relationships are extracted with Name Entity Recognition (NER) instead of LLMs. Characters are extracted with NER directly, chapter by chapter. Relationships are extracted based on character co-occurrence in text windows and verb-based dependency recognition.

Chapter 1 retention Chapter 1 retention reflects the percentage of entities from the first chapter that are still present in the current chapter, checking the pipeline's performance in remembering long-term relationships and callback characters.

$$\text{chapter 1 retention} = \frac{current \cap chapter_1}{|chapter_1|}$$

Rolling Retention Rolling retention represents the percentage of the previous chapter's entities that are still present in the selected chapter, evaluating the story's chapter-to-chapter continuity.

$$\text{rolling retention} = \frac{current \cap previous}{|previous|}$$

Cumulative Retention Cumulative retention shows the percentage of all historical entities currently active in the selected chapter, reflecting the overall engagement of the new chapter with the established storyline.

$$\text{cumulative retention} = \frac{current \cap all\,previous}{|all\,previous \cup current|}$$

Experiment Setup

To evaluate the effect of the pipeline on narrative retention, a series of Student's T-tests was conducted on the retention metrics, comparing two groups of LLM-generated stories (specifically with the claude-haiku-4.5 model: those produced with the multi-agent KG pipeline (50 stories, 270 chapters total) and those produced with naive prompting (50 stories, 231 chapters total). Ten dependent variables were examined across two categories: character retention (total characters, new characters, Chapter 1 retention, rolling retention, and cumulative retention) and relationship retention (total relationships, new relationships, Chapter 1 retention, rolling retention, and cumulative retention). Character retention results are presented in Table 1, and relationship retention results are presented in Table 2.

Experiment Results

Empirically, we show that stories generated with our knowledge-graph-guided pipeline exhibit higher character and relationship retention than those generated without graph integration. Our NER-based evaluation reveals consistent gains in narrative stability across both character and relationship dimensions. From Table 1 and Table 2, five of ten metrics showed statistically significant differences between KG-guided and stories generated from naive prompting. From Table 1, KG-guided stories retained significantly more chapter 1 core characters across the following chapters (mean = 0.689) compared to No-KG stories (mean = 0.622). Cumulative character retention was also significantly higher in the KG condition (mean = 0.500 vs. mean = 0.457), together with rolling character retention (mean = 0.770 vs. mean = 0.731). Also, the KG-guided pipeline tends to introduce fewer new characters in each chapter (mean = 1.244 vs. mean = 1.454), indicating a focus towards building deeper character development instead of introducing new but flat characters. For relationships, KG-guided stories produced significantly more total relationships per chapter (mean = 3.948 vs. mean = 3.208) and retained a significantly greater proportion of chapter 1 relationships across subsequent chapters (mean = 0.294 vs. mean = 0.195). Notably, no significant differences exist for the number of new relationships per chapter, or for rolling or cumulative relationship retention, indicating that KG guidance primarily strengthens the persistence of early-established narrative relationships while lacking emphasis on immediate relationship retention. Nevertheless, it performs better than base-

Table 1: Character Retention

Metric	KG			No-KG			t-test		
	M	SD	n	M	SD	n	t	p	Cohen's d
Total Characters	5.344	2.233	270	5.329	2.061	231	0.08	0.936	0.007
New Characters	1.244	1.324	270	1.454	1.422	231	-1.71	0.088	-0.153
Chapter 1 Retention**	0.689	0.240	270	0.622	0.264	231	2.97	0.003	0.266
Rolling Retention*	0.770	0.202	270	0.731	0.216	231	2.11	0.035	0.189
Cumulative Retention**	0.500	0.188	270	0.457	0.181	231	2.64	0.008	0.237

Note: M = Mean, SD = Standard Deviation, n = Sample Size (chapters). $^{*}p < 0.05$, $^{**}p < 0.01$.

Table 2: Relationship Retention

Metric	KG			No-KG			t-test		
	M	SD	n	M	SD	n	t	p	Cohen's d
Total Relationships*	3.948	3.938	270	3.208	3.954	231	2.09	0.037	0.188
New Relationships	2.030	2.265	270	1.957	3.090	231	0.31	0.753	0.028
Chapter 1 Retention**	0.294	0.383	233	0.195	0.332	204	2.89	0.004	0.277
Rolling Retention	0.395	0.372	221	0.350	0.378	170	1.16	0.245	0.119
Cumulative Retention	0.170	0.194	268	0.139	0.213	224	1.72	0.086	0.156

Note: M = Mean, SD = Standard Deviation, n = Sample Size (chapters). $^{*}p < 0.05$, $^{**}p < 0.01$.

line LLM in preserving characters both long-term and short-term. Taken together, these results provide concrete and interpretable evidence that agentic KG conditioning leads to stronger long-term narrative consistency in LLM-generated text, particularly for characters and relationships introduced early on.

Discussion

This work demonstrates that integrating knowledge graphs into a multi-agent generation pipeline substantially improves the cohesion, consistency, and structural integrity of long-form stories generated by LLMs. By explicitly tracking entities and relationships through an evolving narrative knowledge graph, our framework addresses persistent limitations of traditional LLM prompting, including character loss, relationship drift, and logical inconsistency between chapters. Using KG as an intermediate shared data structure across agents also preserves an explicit trace of the logic behind the generated content, with the chapter-by-chapter evolving KGs as a creativity footprint.

Case Study

For illustration, we can take two examples of stories generated with the same starting prompt:

> "During a chess tournament, the reigning champion collapses dead mid-game, poisoned by a rare toxin."

The KG-guided story (Story 35) ended up with 10 chapters, while the Non-KG Generated story (Story 30) ended with 6 chapters. The ending knowledge graph representations of both stories are present in Figure 2. The KG guided story demonstrates stronger character and relationship consistency across its ten chapters, primarily because its protagonist, Sarah Mitchell, is given sufficient space for each psychological escalation to feel earned. Her core identity as a forensic analyst who left the job but can't stop thinking like a detective is established in the opening chapter and never falters. The narrative has prepared each risk that she takes: her decision to withhold Marcus's letter from Rousseau in Chapter 3 is a logical consequence of the detachment she expressed in Chapter 1; her impetuous walk into Henri's office in Chapter 8 is the logical conclusion of a stubbornness that has been tested and reaffirmed in several previous chapters. The most carefully sustained dynamic in the story is her relationship with Inspector Rousseau. He's skeptical but reverent since their initial encounter. He always pushes back on her methods. His surveillance of her in Chapter 9 is a real payoff because his protectiveness has been quietly planted since Chapter 4. The villain Henri Devereaux is treated with remarkable coherence also: his Chapter 8 confession monologue is consistent with the phrasing of his Chapter 7 phone call, and his farewell letter in Chapter 10 is in character: proud, unsentimental, never truly remorseful.

The story generated without KG guidance has a single protagonist, Marcus Ashford, who is built around an equally compelling premise: a man whose career was deliberately destroyed by a system he never understood. But the story has continuity issues that break that premise. The first few chapters show Marcus as a man who has spent thirty-six years avoiding confrontation, deliberately living in obscurity, and refusing to re-enter the chess world. But by Chapter 4, after one long talk with David Rothschild, he's all in on a high-stakes international conspiracy bust. The internal resistance that should define him collapses in roughly ten pages without meaningful dramatization of the struggle. This is the story's central consistency failure: a character

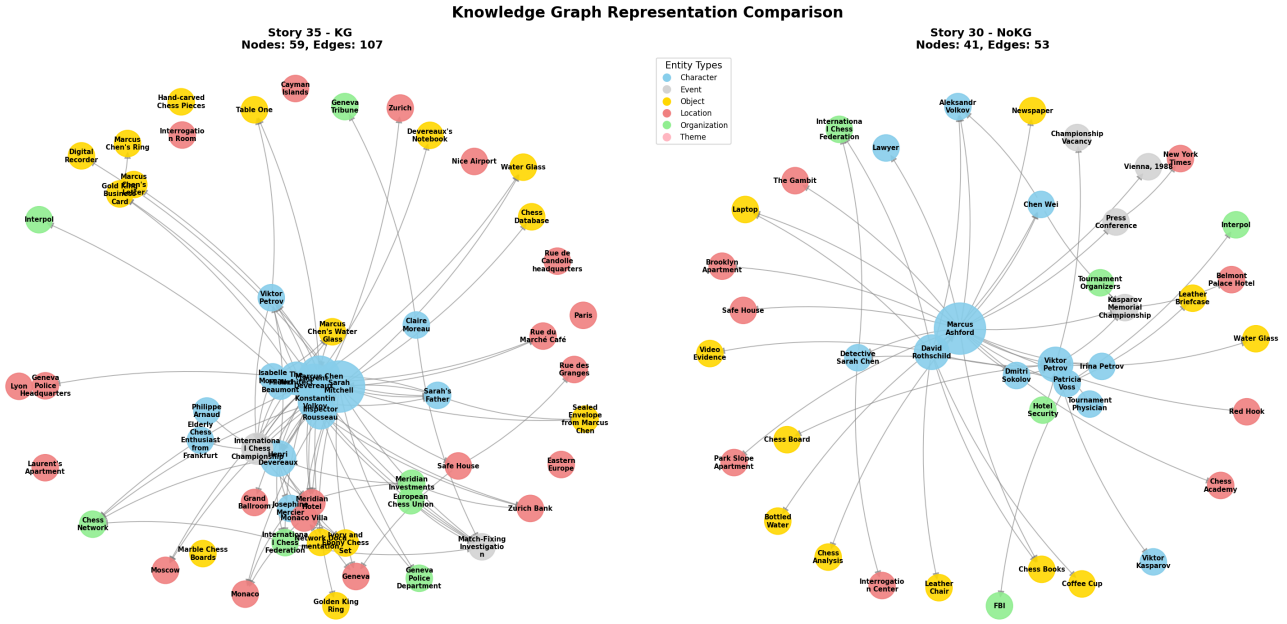

Figure 2: Knowledge Graph of Story Generated with Multi-Agent Pipeline (left) and with Vanilla Prompting (right)

whose entire psychological identity is built around avoidance should require far more pressure and time to reverse course than one conversation allows. The KG-guided story avoids this problem because Sarah resists, partially retreats, and re-commits across several chapters before each major escalation, making her transformation feel cumulative rather than sudden. The supporting characters in the No-KG story compound this problem in ways that the KG-guided story largely avoids. David Rothschild is introduced in Chapter 3 as a man paralyzed by thirty-six years of secret knowledge, too afraid to act alone. But Chapter 4 reveals he has already been actively coordinating with an Interpol undercover agent and maintaining a safe house network. This directly contradicts the passivity established just one chapter earlier. In addition, Sarah Chen is introduced in Chapter 2 as an influential figure who frames Marcus as a suspect, yet she disappears entirely from the narrative after that chapter with no resolution. The KG-guided story has its own version of this problem. Laurent Devereaux is introduced as a terrified ally, then dies off-page, but the story at least references his fate and gives it emotional weight at a funeral scene. The No-KG story simply stops acknowledging Detective Chen's existence. The one genuine consistency success in this story is Chen Wei, whose strange calm during Viktor's death in Chapter 1 is retroactively explained by his Chapter 4 reveal as an undercover agent. But this single well-executed thread cannot compensate for the broader pattern of characters introduced and then abandoned or contradicted. Taken together, the KG-guided generation workflow honors its characters across time, creating fluent character development and problem resolution.

Limitations and Future Work

While the generated stories exhibit stronger logic and continuity, challenges remain. We asked several human evaluators to read stories generated from the pipeline. Human evaluations indicate that the narratives can still lack stylistic specificity, emotional depth, and distinctive authorial voice. This suggests that while structural cohesion can be enforced through graph-based reasoning, higher-level creativity and expressiveness may require additional constraints or specialized agents.

Future work can explore several extensions to this framework. First, we plan to introduce verifier and critic agents to refine specificity, voice, and narrative tension during generation. Second, we aim to expand our evaluation suite by incorporating LLM-as-judge assessments more systematically and developing richer relation-strength and character-interaction metrics. We acknowledge that retention-based analysis focuses primarily on character and relationship cohesion and lacks complexity in measuring creativity at a human level. Third, we will explore conditioning the pipeline on different base models to study how agentic structure interacts with underlying model capabilities. Finally, we envision extending the system beyond narrative generation to educational applications, including intelligent textbooks that automatically generate comprehension questions, timelines, and concept maps from the underlying knowledge graph, as well as multi-modal extensions that incorporate illustrations to support younger readers.

Conclusion

Overall, this work highlights the effectiveness of multi-agent, knowledge-graph-driven generation as a scalable so-

lution for producing coherent long-form content. By combining explicit structural representations with agentic reasoning, our approach represents a promising step toward intelligent creative systems that support deeper text comprehension, personalization, and sustained engagement.

References

Blin, I. 2022. Building narrative structures from knowledge graphs. In Groth, P.; Rula, A.; Schneider, J.; Tiddi, I.; Simperl, E.; Alexopoulos, P.; Hoekstra, R.; Alam, M.; Dimou, A.; and Tamper, M., eds., *The Semantic Web: ESWC 2022 Satellite Events*, volume 13384 of *Lecture Notes in Computer Science*, 234–251. Hersonissos, Crete, Greece: Springer.

Cabral, B.; Claro, D.; and Souza, M. 2024. Exploring open information extraction for portuguese using large language models. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, 127–136. Santiago de Compostela, Galicia/Spain: Association for Computational Linguistics.

Damiano, R., and Lieto, A. 2013. 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)*, 76–93. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

de Kok, M.; Rebboud, Y.; Lisena, P.; Troncy, R.; and Tiddi, I. 2024. 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. Presented at Text2Story 2024 Workshop.

Ehrlinger, L., and Wöß, W. 2016. Towards a definition of knowledge graphs. In *International Conference on Semantic Systems*.

Ello. 2025. Ello – a creative network. `https://www.ello.com/`. Accessed: 2025-05-22.

Gervás, P. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30:49–62.

Graesser, A. C.; McNamara, D. S.; Louwerse, M. M.; and Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers* 36(2):193–202.

Grishman, R., and Sundheim, B. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Guan, J., and Huang, M. 2020. Union: An unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9157–9166. Online: Association for Computational Linguistics.

Han, H.; Wang, Y.; Shomer, H.; Guo, K.; Ding, J.; Lei, Y.; Halappanavar, M.; Rossi, R. A.; Mukherjee, S.; Tang, X.; He, Q.; Hua, Z.; Long, B.; Zhao, T.; Shah, N.; Javari, A.; Xia, Y.; and Tang, J. 2025. Retrieval-augmented generation with graphs (graphrag). `https://arxiv.org/abs/2501.00309`. Preprint, arXiv:2501.00309.

Ismayilzada, M.; Stevenson, C. E.; and van der Plas, L. 2024. Evaluating creative short story generation in humans and large language models. *ArXiv* abs/2411.02316.

Jiang, J.; Zhou, K.; Zhao, W. X.; Song, Y.; Zhu, C.; Zhu, H.; and Wen, J.-R. 2024. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. `https://arxiv.org/abs/2402.11163`. Preprint, arXiv:2402.11163.

Laclaustra, I.; Ledesma, J.; Méndez, G.; and Gervás, P. 2014. Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In *Proceedings of the fifth international conference on computational creativity*.

LangChain. 2026. langchain-ai/langgraph. original-date: 2023-08-09T18:33:12Z.

Lavie, A., and Agarwal, A. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, 228–231. USA: Association for Computational Linguistics.

Méndez, G., and Gervás, P. 2023. Using chatgpt for story sifting in narrative generation. In *Proceedings of the 14th international conference on computational creativity*.

Ocampo, R.; Bown, O.; and Grace, K. 2025. Beyond chat: collaborative editors enhance human involvement and agency when co-writing with large language models. In *Proceedings of the Sixteenth International Conference on Computational Creativity*.

Pakhale, K. 2023. Comprehensive Overview of Named Entity Recognition: Models, Domain-Specific Applications and Challenges. arXiv:2309.14084 [cs].

Pan, Z.; Andronis, A.; Hayek, E.; Wilkinson, O. A. P.; Lasy, I.; Parry, A.; Gadney, G.; Smith, T. J.; and Grierson, M. 2025. Guiding generative storytelling with knowledge graphs. *International Journal of Human–Computer Interaction* 1–23.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Peng, X.; Li, S.; Wiegreffe, S.; and Riedl, M. 2023. Inferring the reader: Guiding automated story generation with commonsense reasoning. `http://arxiv.org/abs/2105.01311`. Preprint, arXiv:2105.01311.

Pickering, T., and Jordanous, A. 2017. Applying narrative theory to aid unexpectedness in a story generation system. In *Proceedings of the eighth international conference on computational creativity*.

Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *J. Exp. Theor. Artif. Intell.* 13:119–139.

Rebboud, Y.; Lisena, P.; and Troncy, R. 2022. 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*, 121–135. Springer.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39:217–268.

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

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

Veale, T.; Cardoso, A.; and y Pérez, R. P. 2019. Systematizing creativity: A computational view. In *Computational Creativity*.

Wang, W.; Gao, M.; Hu, X.; and Wan, X. 2025. Towards a "novel" benchmark: Evaluating literary fiction with large language models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 21648–21673. Vienna, Austria: Association for Computational Linguistics.

Yates, A.; Banko, M.; Broadhead, M.; Cafarella, M.; Etzioni, O.; and Soderland, S. 2007. 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)*, 25–26. Rochester, New York, USA: Association for Computational Linguistics.

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

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2335–2344. Dublin, Ireland: Dublin City University and Association for Computational Linguistics.

Appendix A: LLM Prompts

This appendix documents the exact system and user prompts passed to the language model in both the knowledge-graph-augmented pipeline and the baseline pipeline without knowledge graphs.

A.1 KG Pipeline

A.1.1 Writer Agent – Chapter 1

System prompt

You are a creative story writer. Write the first chapter of a story following the '*<archetype>*' narrative archetype.

Archetype description: *<archetype_description>*

Starting premise: *<starting_prompt>*

Write an engaging first chapter (800--1200 words) that:

- Introduces the main character(s)
- Sets up the initial situation
- Establishes the tone and setting
- Hooks the reader
- Ends with a compelling reason to continue

After writing, call the submit_chapter tool with your chapter text and set is_final_chapter to FALSE (this is just the beginning).

The '*<archetype>*' archetype should guide the overall story structure, so keep it in mind for future chapters.

User prompt

Write Chapter 1 of this story.

A.1.2 Writer Agent – Chapter $N > 1$

System prompt The system prompt for subsequent chapters is assembled dynamically and contains three logical sections: the core instruction block, the story context (last 1–3 saved chapter files), and a knowledge-graph context block.

Core instruction block

You are a creative story writer continuing a story following the '*<archetype>*' narrative archetype.

Archetype description: *<archetype_description>*

This is chapter *<N>*.

Story so far (last few chapters): *<story_so_far>* *<kg_context>*

Your task:

1. Write the next chapter (800--1200 words) that continues from where the previous chapter left off
2. Review the KNOWLEDGE GRAPH above to maintain consistency with established characters, locations, and relationships
3. Monitor the story's progression according to the '*<archetype>*' archetype and its description above
4. DECIDE WHETHER THE STORY HAS REACHED ITS NATURAL CONCLUSION based on:
   - Whether the archetype's narrative structure is complete
   - Whether character arcs are resolved
   - Whether the main conflict has been addressed
   - Whether the story feels satisfying and complete
5. Call submit_chapter with:
   - chapter_text: Your full chapter
   - is_final_chapter: TRUE if this completes the story, FALSE if more chapters are needed

Pacing guidelines:

- Let the story develop naturally -- don't rush to the end
- Ensure proper pacing according to the archetype description above
- The story should feel complete when you mark it as final, not abrupt

Write ONLY narrative content, no meta-commentary.

Knowledge-graph context block (<kg_context>) When entities have been extracted from previous chapters the following structured block is appended to the system prompt immediately after the story text:

KNOWLEDGE GRAPH (use this to maintain consistency):

Introduced in Chapter 1 (foundational): *<character/organisation nodes and relationships from chapter 1>*

Introduced in Chapter *<M>* (milestone): *<nodes and relationships from milestone chapter M>*

New in Chapter *<N-k>*: *<nodes and relationships from recent chapter N-k>*

All Characters & Organisations (cumulative): *<all character and organisation nodes accumulated so far>*

All Character Relationships (cumulative): *<all edges involving at least one character/organisation node>*

Other Entities: *<non-character nodes>*

Other Relationships: *<non-character edges>*

Milestone chapters are every fifth chapter starting at chapter 6 (i.e. 6, 11, 16, ...). Recent chapters are the three chapters immediately preceding chapter N (i.e. $N-3$, $N-2$, $N-1$), excluding chapter 1 and milestone chapters.

User prompt

Write Chapter *<N>* and decide if the story should continue.

A.1.3 Grapher Agent

System prompt The system prompt is assembled from a fixed instruction block and an optional canonical-entity block derived from the current store contents.

Canonical-entity block (if prior entities exist)

EXISTING CANONICAL ENTITIES (you MUST reuse these exact IDs and labels --- do not create duplicates): *<id="..." label="..." type="..." for each stored entity>*

EXISTING RELATIONSHIPS (re-include any that are still active in the story --- do not drop established connections): *<from -> to [label] for each stored relationship>*

Fixed instruction block

You are the KNOWLEDGE-GRAPH AGENT.

Your job is to extract entities and relationships from story chapters while keeping the knowledge graph consistent across chapters. *<canonical_block>*

RECONCILIATION RULES (critical):

- If a name in the chapter refers to the same person/place/thing as an existing canonical entity, use the EXISTING id and label -- do not invent a new one.
- Only create a NEW entity node if it is genuinely not present in the canonical list above.
- Characters who appear under nicknames, shortened names, or slight variations must be mapped to the matching canonical entity.

For nodes, include: id, label, and type (must be one of: character, event, object, location, organization)
For edges, include: from, to, and label fields.

RELATIONSHIP EXTRACTION PRIORITY:

- Character-to-character relationships are the most valuable -- extract these first and as completely as possible
- Then character-to-location and character-to-organisation relationships
- Character-to-event and character-to-object edges are lowest priority

IMPORTANT ID FORMATTING RULES:

- Node IDs must use underscores instead of spaces (e.g. High_School, Detective_Smith)
- Edge from and to fields must exactly match the node IDs you define (same underscore format)

Categories are strictly: character, event, object, location, organization.

User prompt

Extract entities and relationships from this chapter:

<chapter_text>

A.2 Starting Prompts

Each story is initialized with one of the following ten starting premises, selected at random at runtime:

1. A renowned detective receives an anonymous letter claiming that a murder will occur in exactly 48 hours.
2. A small coastal town's lighthouse keeper is found dead, and the only clue is a series of mysterious light patterns.
3. During a chess tournament, the reigning champion collapses dead mid-game, poisoned by a rare toxin.
4. A young apprentice discovers they can see magical creatures that no one else can perceive.
5. In a world where dragons have been extinct for centuries, a farmer finds a living dragon egg in their field.
6. A colony ship arrives at a distant planet only to find it already inhabited by humans who left Earth centuries later.
7. An AI developed to predict crime begins warning about murders that haven't been planned yet.
8. A family moves into their dream home, only to discover the previous owners never actually left.
9. Two rival wedding planners are forced to collaborate on the biggest celebrity wedding of the year.
10. A treasure map tattooed on a dying sailor leads to an island that doesn't appear on any modern charts.