NOVELDREAMER: HARNESSING LLMS FOR COHERENT AND ENGAGING LONG-FORM STORYTELLING

Arian Emami
emamiarian8@gmail.com

ABSTRACT

Recent advancements in large language models (LLMs) have demonstrated significant potential in text generation, but challenges remain, particularly in crafting long-form narratives that maintain consistency in topic and style, engage readers, and preserve coherence. NovelDreamer addresses these issues through a multifaceted approach. It employs a retrieval-augmented generation (RAG) method, utilizing samples from related works sourced from Wikiquote to enhance stylistic and thematic consistency. To ensure narrative engagement, NovelDreamer integrates established story structures, such as the Hero's Journey and Freytag's Pyramid, guiding the LLM in crafting compelling and well-structured stories. Additionally, by pre-planning the story into chapters and acts, NovelDreamer facilitates the creation of coherent and captivating long-form narratives. The code for NovelDreamer is publicly available at this URL.

***K*eywords** Large Language Models · Story Generation · Narrative Coherence · Retrieval-Augmented Generation · Story Structures · Thematic Consistency · Long-Form Storytelling

1 Introduction

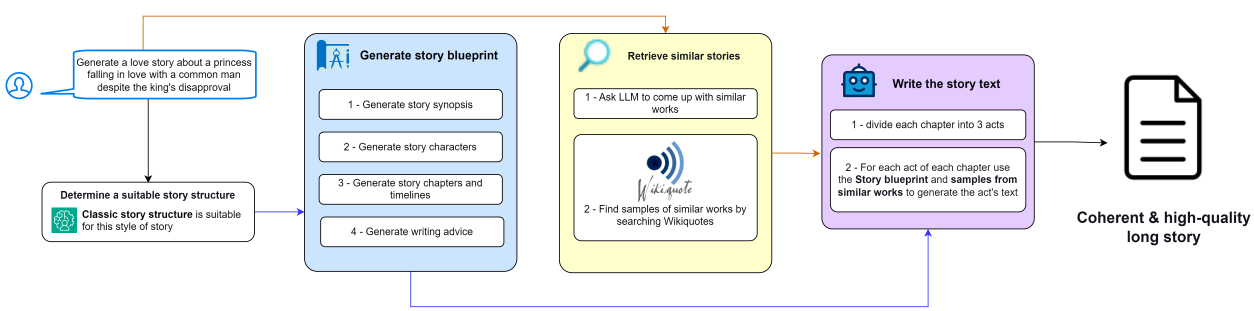

Figure 1: The workflow of NovelDreamer.

The field of Natural Language Processing (NLP) has witnessed remarkable advancements in recent years, with automated story generation emerging as a particularly challenging and creative task [1]. By learning from human-written narratives, automated storytellers aim to produce engaging stories for various applications, including entertainment, education, and social bonding. The advent of deep learning techniques has led to significant progress in data-driven approaches to automated story generation [2, 3, 4]. With the rapid development of large language models (LLMs), generated stories have significantly improved in length, complexity, and fluency.

Despite these advancements, existing work on LLM-based computational storytelling faces several challenges. While current research primarily focuses on optimizing automated story generation systems from various angles, such as long-form generation [5, 6] and controllable generation [7, 8], there remain significant hurdles in producing consistently engaging and coherent narratives.

In this paper, we address three critical areas where previous methods using LLMs struggle with story generation [1]:

- **Topic/style/genre matching:** Existing approaches often fail to accurately capture the desired style or genre of a story. We address this issue by implementing a Retrieval-Augmented Generation (RAG) approach, providing samples from popular similar works extracted from Wikiquotes. This enhancement significantly improves the LLM's adherence to the intended style and genre.
- **Interestingness:** Maintaining reader engagement throughout the narrative is a persistent challenge. Our method allows the LLM to choose from established story structures such as the Hero's Journey and Freytag's Pyramid. These structures serve as guides for the LLM to plan ahead and maintain story engagement across its entire length.
- **Coherence:** LLMs often struggle with maintaining narrative coherence over extended story lengths. By planning the story ahead into several chapters and acts, we closely guide the LLM in creating coherent stories and prevent drifting into nonsensical narratives.

Our approach builds upon recent advancements in LLM capabilities while addressing their limitations in the context of story generation. By combining traditional narrative theories with modern language generation techniques, we present a novel method that significantly improves the quality and consistency of automatically generated stories.

The main contributions of our work are as follows:

- We propose a RAG-based technique that enhances the LLM's ability to match desired topics, styles, and genres in story generation.
- We introduce a method for incorporating established story structures into the LLM's planning process, improving overall narrative interestingness and engagement.
- We develop a chapter-based planning approach that enhances story coherence and prevents narrative drift in long-form story generation.

2 Related Work

2.1 Automated Story Generation

The field of automated story generation has evolved significantly over the years, with approaches ranging from symbolic planning to neural language modeling and, most recently, large language models [9].

2.1.1 Symbolic Planning Approaches

Early work on story generation relied heavily on symbolic planning techniques. These systems required substantial knowledge engineering of logical constraints, which limited their generality. While effective in certain contexts, they often struggled to generate plots or stories in natural language, making them less suitable for general-purpose storytelling [10, 11, 12, 13, 14].

2.1.2 Neural Language Modeling Approaches

The advent of neural language modeling approaches [2, 3, 4] marked a significant shift in the field. These methods circumvented the need for manual knowledge engineering and tended to produce relatively fluent, varied, and naturalistic language. Research in this area has focused on various aspects of story generation:

- **Controllability:** A significant amount of work has been dedicated to improving the controllability of story generators [7, 8].
- **Story Quality:** Researchers have aimed to enhance different dimensions of story goodness, such as common-sense reasoning [15] and temporal and causal relationships [15, 16].

2.1.3 Large Language Models

The introduction of large, pre-trained language models such as GPT-3, ChatGPT, and GPT-4 has further advanced the field. These models are capable of generating longer, more fluent story sequences, with some approaches extending generation to many thousands of words [6, 17]. However, despite their impressive capabilities, LLMs have been unreliable when it comes to generating novel, suspenseful stories. This limitation is partly due to the complex cognitive nature of suspense, which does not naturally emerge from the latent state representations of transformer models.

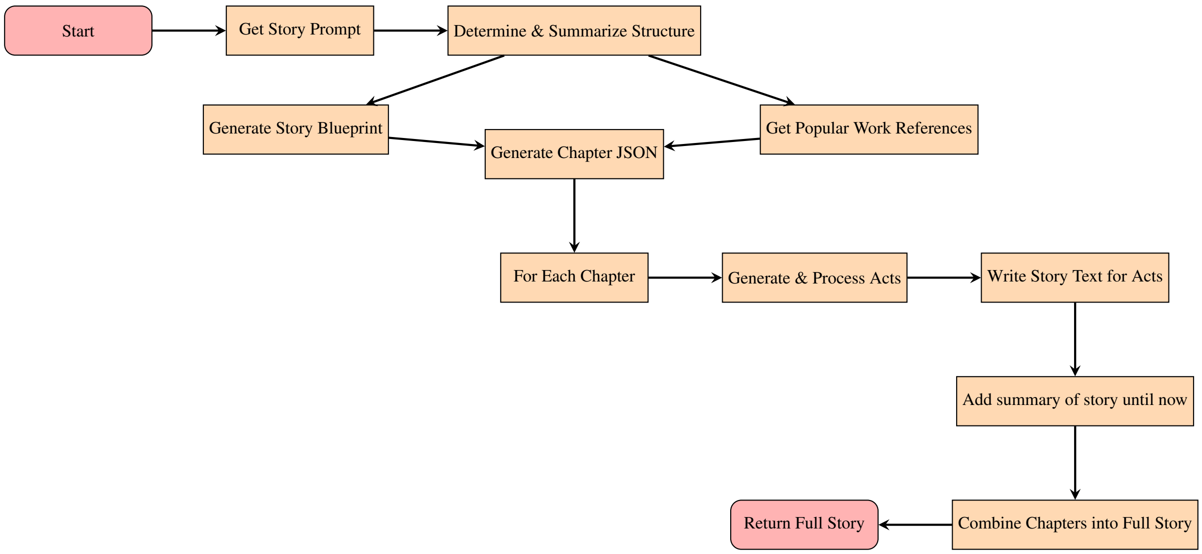

Figure 2: Flowchart of the story generation process.

2.2 Storytelling with Reinforcement Learning

Reinforcement learning has been applied to content generation in various ways:

- **Fine-tuning:** RL is often used for fine-tuning language models [18, 19].
- **Auxiliary Model Guidance:** Some approaches use RL for auxiliary model guidance in story generation [20, 21].
- **Dynamic Inference-time Option-selection:** Methods involving dynamic inference-time option-selection and/or classification [22, 23, 20] are particularly relevant to our work.

2.3 Controlled Text Generation via Prompting

Recent advancements in language models have increased the popularity of prompting approaches:

- **Manual Prompts:** Some approaches use manually designed prompts [24].
- **Automatic Prompts:** Other methods focus on automatically designing prompts [25, 26].
- **Iterative Prompting:** Chain-of-thought prompting represents an iterative approach to text generation [27].
- **Continuous Soft Prompts:** Some works explore the use of continuous soft prompts for text generation [28, 29].

2.4 Human-in-the-Loop Story Generation

In contrast to fully automated approaches, some research has explored human-in-the-loop methods for generating interesting long stories [30, 31, 32, 33, 34, 35, 36]. While our method is fully automatic, its flexible action space makes it amenable to human collaboration and tuning.

Our work builds upon these various approaches, introducing a novel method that combines the strengths of large language models, reinforcement learning, and controlled text generation. By using an adapted LLM to interpret an internal representation of the current story, employing a highly modular structure, and utilizing a prompting-based approach, our method aims to generate diverse, engaging, and suspenseful stories while addressing the limitations of previous approaches.

3 Methodology

The process of generating a coherent and engaging narrative through an AI-based approach requires a systematic and methodologically sound procedure. As illustrated in Figure 2, this section outlines the main steps used by our story-creating system to produce stories that are not only engaging and logically consistent but also follow common story structures often used in written works. The method is divided into five stages, each designed to optimize specific aspects of story generation, ensuring that the final output is of high literary quality.

3.1 Story Structure Selection

The first stage in our process involves the selection of an appropriate story structure. Narrative structures, such as the Hero's Journey and Freytag's Pyramid, are foundational in maintaining engagement throughout the length of a story. This step is crucial because long-form storytelling, especially when generated by AI, tends to suffer from issues of coherence and sustained reader interest'[1]. The Large Language Model (LLM) is provided with a comprehensive set of common storytelling structures within its prompt to choose from. These structures include:

- **Classic Story Structure**: Suitable for traditional narratives in genres like romance, drama, or adventure.
- **Freytag's Pyramid**: Ideal for tragic tales or stories with a somber tone.
- **The Hero's Journey**: Appropriate for epic tales, fantasy, adventure, and stories of significant transformation.
- **Three Act Structure**: Well-suited for stories with clear conflict and resolution, such as dramas, comedies, and action films.
- **Dan Harmon's Story Circle**: Best for character-driven stories, especially in episodic content.
- **Fichtean Curve**: Perfect for stories with intense drama and suspense, like thrillers.
- **Save the Cat Beat Sheet**: Excellent for structured narratives requiring tight pacing and clear turning points.
- **Seven-Point Story Structure**: Ideal for stories focused on dramatic transformations, particularly in fantasy, sci-fi, or adventure genres.

The LLM is tasked with not only selecting one of these structures but also providing a detailed rationale for its choice. This decision-making process involves:

1. Analyzing the input prompt to identify key thematic elements and narrative scope.
2. Evaluating how each provided story structure aligns with the prompt's requirements.
3. Drawing upon its knowledge base to suggest similar popular works that successfully employed the chosen structure.
4. Articulating a comprehensive justification for the selected structure, considering both the prompt's specifics and the potential for enhancing reader engagement.

This approach leverages the LLM's vast knowledge of literature and allows for a more flexible and context-aware selection process, addressing the challenges of AI-generated long-form narratives by grounding the structural choices in both the specific requirements of the prompt and established storytelling practices.

Once the most appropriate structure is selected, the model generates a summary of its findings. This summary serves two primary purposes: it provides a clear rationale for the choice of structure and distills the key elements of the chosen structure into a concise format. By summarizing these elements, we significantly reduce the cognitive load on the model in subsequent steps, ensuring that it remains focused on the critical aspects of the narrative without being bogged down by extraneous details. This strategic reduction in token usage prevents the model from drifting away from the narrative's core objectives, thus reducing the chance of errors in story generation.

3.2 Incorporation of Popular Works and Style Adaptation

The second stage focuses on embedding stylistic elements from popular works into the narrative. After the story structure is chosen and summarized, the model identifies popular works that share thematic or structural similarities with the story being generated. This step involves a dual process of extracting references to these works in a structured JSON format and then querying an external resources, namely Wikiquote, to retrieve relevant quotes and stylistic samples.

The inclusion of references from popular works serves a dual purpose. Firstly, it provides the model with high-quality, contextually appropriate examples that can guide the generation of prose, dialogue, and narrative pacing. By integrating these examples, the model can align its output with proven literary styles, thereby improving the fluency and readability of the generated text. Secondly, these references act as a form of "style transfer," subtly steering the model's output towards the tone and voice of well-regarded authors or genres. This approach is grounded in the principle that example-based learning can significantly enhance the quality of AI-generated content, particularly in creative domains like storytelling [37].

The process begins with the model running the story structure through a prompt that outputs the popular works in JSON format. This structured data is then used to search for relevant quotes from these works, which are subsequently formatted and integrated into the generation prompts.

3.3 Story Blueprint Generation

With the structure and stylistic influences in place, the next step involves creating a comprehensive story blueprint. This blueprint is a detailed plan that guides the model through the subsequent phases of narrative generation, ensuring that all story elements are cohesively interwoven. The blueprint generation process is one of the most critical aspects of our methodology, as it establishes the foundation upon which the entire narrative is built.

The initial task of the model is to generate a high-level synopsis, providing a broad overview of the story's setting, key characters, and plot direction. This synopsis acts as a conceptual anchor, helping the model maintain a consistent narrative trajectory as it elaborates on finer details in later stages.

Following the synopsis, the model generates the story's theme and core concept. This step is vital for establishing the underlying messages and moral lessons that the story aims to convey. By defining these elements early on, the model can ensure that character development, plot twists, and thematic arcs align with the overarching narrative goals. This thematic consistency is crucial for maintaining the story's integrity and ensuring that it resonates with readers on a deeper level.

Character profiles are then generated based on the established theme and core concept. These profiles include detailed descriptions of each character's personality, motivations, and relationships. By grounding character creation in the thematic context, the model can produce characters that are not only believable but also integral to the narrative's progression. The LLM is asked to map out the interrelationships between characters to ensure that their interactions contribute meaningfully to the story's development.

With the character profiles in place, the model then outlines the chapters and the story's timeline. This stage involves dividing the narrative into manageable segments, each with a clear focus and purpose. The chapter outline serves as a road-map for the story, detailing key events, character arcs, and plot developments that will occur in each segment. The timeline ensures that these events unfold in a logical and compelling sequence, maintaining the reader's engagement from start to finish while adhering to the chosen story structure.

Finally, the model generates a list of writing advice specific to the story. This advice includes guidelines on how to implement the chosen story structure, how to write effective dialogue, and how to develop characters consistently throughout the narrative. These guidelines serve as a set of heuristics that the model can reference during the actual story writing process, ensuring that the generated text adheres to best practices in storytelling.

To conclude this stage, the blueprint is processed by the model to produce a JSON representation of the chapters, complete with their descriptions. This structured format is used for the next phase of story generation, where the model will need to reference specific chapter details to maintain coherence and continuity across the narrative.

3.4 Chapter and Act Division

The fourth stage addresses one of the most challenging aspects of AI-driven storytelling: maintaining coherence across long-form narratives [1]. Given the limitations of current language models, which struggle with generating long texts while preserving narrative consistency, our method introduces a hierarchical division of the story into chapters and acts. This division not only makes the task more manageable for the model but also aligns with traditional narrative structures, where each act serves a specific purpose within the broader context of the chapter and story.

After the chapters have been outlined in the blueprint, the model proceeds to divide each chapter into three distinct acts. This allows the model to focus on a smaller narrative scope at each step, reducing the cognitive load and minimizing the risk of losing track of important plot details. Each act is generated with a clear description and accompanying writing advice, derived from the overall blueprint, ensuring that it adheres to the intended narrative arc.

The model then processes this analysis to extract the act descriptions and writing advice in a structured JSON format. This format is critical for the subsequent stage, where the actual story text for each act will be generated. By dividing the narrative into acts with specific descriptions and guidelines, the model can concentrate on generating coherent, focused segments of text, which are later combined to form the complete chapter.

The division into acts also facilitates the maintenance of narrative tension and pacing, as each act can be tailored to fulfill a specific role within the chapter, whether it be introducing a conflict, developing a character, or resolving a plot point. This structured approach not only enhances the narrative's coherence but also ensures that the story remains engaging, with each act contributing meaningfully to the overall narrative progression.

3.5 Generation of Story Text

The final stage of the process involves the actual generation of the story text. This stage is where the model synthesizes all the information from the previous steps—the story structure, the blueprint, the chapter outlines, and the act descriptions to produce the narrative.

To maintain coherence with previously generated content, the model begins by summarizing the previous chapters, excluding the current chapter's acts, and appending this summary to the prompt. This summarization is crucial because directly providing the model with the raw text of previous chapters can overwhelm the LLM, which has demonstrated limitations in effectively processing and utilizing information from extremely long context prompts [38]. Such an approach increases the risk of the LLM losing track of the task at hand, potentially introducing inconsistencies or irrelevant details into the narrative. By condensing the preceding content into a summary, we mitigate these risks, ensuring that the model retains the essential context while remaining focused and coherent in its storytelling. Additionally, references to popular works and stylistic examples are included in the prompt, reinforcing the narrative's alignment with proven literary styles.

The model is then instructed to generate the story text for each act, one at a time. For each act, the following information is provided to the model:

- The **story blueprint**, which includes the general synopsis, theme, and core concept of the narrative.
- The **original prompt**, serving as the foundational inspiration for the entire story.
- The **chapter description**, outlining the key events and developments that need to occur within the chapter.
- The **act description**, specifying the particular focus and objectives of the current act.
- Any relevant **writing advice** that the model should follow while generating the text.

If the act being generated is not the first in the chapter, the summary of the previous acts is also included, along with the last few lines of text from the preceding act. This ensures that the transition between acts is smooth and that the narrative flow is maintained throughout the chapter.

By carefully managing the context provided to the model at each stage, our method effectively mitigates the challenges associated with long-form text generation, such as the tendency to drift away from the main plot or the introduction of inconsistencies in character behavior or plot developments. The end result is a coherent, engaging, and well-structured narrative that aligns with the chosen story structure and stylistic influences.

3.6 Implementation Details

The entire process of story generation, from preliminary structure selection to the final text output, is implemented using **LLaMA 3.1 8B Instruct** model [39], integrated with custom scripts for generating JSON-formatted data and prompts. The process is automated, with minimal human intervention, ensuring that the storytelling agent can generate narratives at scale while maintaining a high level of quality.

The prompts used at each stage are carefully crafted to guide the model towards producing outputs that align with the intended narrative structure and style. These prompts are iteratively refined based on feedback from initial test runs, ensuring that the final prompts are optimized for generating high-quality stories.

4 Discussion: Multi-step agentic approaches versus scaling up LLMs for automated storytelling

The comparison between multi-step agentic approaches to story generation and the strategy of scaling up large language models (LLMs) or fine-tuning them provides valuable insights into the future of automated narrative generation. While recent advancements in scaling up LLMs, such as GPT-4 [40], have led to impressive achievements in generating human-like text, the performance of these models when tasked with long-form storytelling remains inconsistent. Simply prompting these larger models to generate extended narratives often results in outputs that, while fluent, lack the coherence, thematic consistency, and engagement needed to sustain reader interest over the course of a full-length novel [1].

In contrast, the multi-step agentic approach demonstrated in this work offers a promising alternative. By breaking down the storytelling process into structured steps—such as summarizing previous chapters, planning the narrative across acts, and incorporating thematic and stylistic elements from similar works—this method not only generates more coherent and compelling stories but does so using smaller models like **LLaMA 3.1 8B Instruct** [39]. These models, while significantly less resource-intensive than their larger counterparts, benefit from the guidance provided by the agentic framework, allowing them to surpass the performance of larger models that rely solely on raw prompting.

One of the key advantages of the multi-step agentic approach is its ability to generate high-quality long-form narratives with smaller models that are more affordable to operate and can even be run locally. This makes the technology accessible to a broader range of users and reduces the dependency on powerful, expensive infrastructure. Furthermore, smaller models are easier to fine-tune for specific tasks or domains, enhancing their adaptability and potential for customization. For instance, in this project [41], the use of **LLaMA 3.1 8B Instruct** has proven effective in generating compelling stories by leveraging structured prompts and iterative planning, illustrating that smaller models, when properly guided, can match or even exceed the performance of larger models in certain contexts.

Moreover, research into multi-step agentic systems offers the potential to uncover hidden systems and patterns that underlie effective storytelling. By systematically dissecting the narrative construction process, researchers can gain a deeper understanding of the elements that contribute to a good story—insights that may remain elusive when relying solely on the brute force of scaling up LLMs. This deeper understanding can, in turn, inform the development of new models and techniques that further improve the quality of automated storytelling.

5 Limitations

While the multi-step agentic approach employed in **NovelDreamer** demonstrates significant advancements in long-form story generation, it is not without its limitations. One notable issue arises from the system's reliance on popular works as references to improve prose quality. Although this strategy enhances the narrative style and alignment with established literary conventions, it can inadvertently steer the prose toward familiar tones and styles. This tendency may hinder the novelty and uniqueness of the generated stories, leading to outputs that, while well-crafted, may lack the original voice or innovative flair that distinguishes truly novel works.

Additionally, the system is prone to the overuse of certain phrases and expressions, a common issue in large language models. Phrases such as "she felt a sense of relief wash over her" or "sent a shiver down his spine" tend to appear frequently, albeit in slightly varied forms. This repetition can detract from the overall quality of the narrative, making it feel formulaic or repetitive, and potentially reducing reader engagement.

Another limitation pertains to occasional inconsistencies in the JSON output generated by the LLM, which may not always adhere to the predefined schema. Such discrepancies can disrupt the story generation process, leading to errors or exceptions that may hinder the seamless production of narratives. While these occurrences are rare, they represent a significant challenge, particularly in automated systems that rely on structured data formats for processing and generation.

These limitations underscore the need for ongoing refinement of the NovelDreamer system. Future work may focus on enhancing the system's ability to produce more original and varied prose, reducing the repetition of common phrases, and improving the robustness of the JSON output to ensure consistent adherence to schemas. Addressing these challenges will be crucial in advancing the quality and reliability of AI-generated long-form storytelling.

6 Conclusion

In this paper, we have introduced **NovelDreamer**, a multi-step agentic approach designed to enhance the quality of long-form story generation using large language models. By incorporating retrieval-augmented generation and established narrative frameworks, our system addresses key challenges such as thematic consistency, coherence, and narrative engagement, even with smaller models like LLaMA 3.1 8B Instruct [39]. While the system shows promise in producing compelling and coherent narratives, it is not without limitations, including the potential for reduced novelty in prose and the overuse of certain phrases. Additionally, occasional inconsistencies in the generated JSON output highlight the need for further refinement. Nevertheless, **NovelDreamer** represents a significant step forward in the field of automated storytelling, offering a more accessible and efficient alternative to simply scaling up LLMs. Future work will focus on addressing the identified limitations and further exploring the potential of multi-step agentic approaches to uncover deeper insights into the mechanics of effective storytelling.

References

[1] Ivan Yamshchikov and Alexey Tikhonov. What is wrong with language models that can not tell a story? In *Proceedings of the The 5th Workshop on Narrative Understanding*. Association for Computational Linguistics, 2023.

[2] Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[3] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[4] Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy, July 2019. Association for Computational Linguistics.

[5] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online, November 2020. Association for Computational Linguistics.

[6] Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[7] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[8] Xiangyu Peng, Kaige Xie, Amal Alabdulkarim, Harshith Kayam, Samihan Dani, and Mark Riedl. Guiding neural story generation with reader models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7087–7111, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[9] Kaige Xie and Mark Riedl. Creating suspenseful stories: Iterative planning with large language models, 2024.

[10] James R. Meehan. Using planning structures to generate stories. *American Journal of Computational Linguistics*, pages 78–94, November 1975. Microfiche 33.

[11] Michael Lebowitz. Planning stories. In *Proceedings of the 9th annual conference of the cognitive science society*, volume 1, page 1, 1987.

[12] Marc Cavazza, Olivier Martin, Fred Charles, Steven J Mead, and Xavier Marichal. Interacting with virtual agents in mixed reality interactive storytelling. In *Intelligent Virtual Agents: 4th International Workshop, IVA 2003, Kloster Irsee, Germany, September 15-17, 2003. Proceedings 4*, pages 231–235. Springer, 2003.

[13] Mark O Riedl and R Michael Young. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

[14] Stephen Ware and R Young. Modeling narrative conflict to generate interesting stories. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6, pages 210–215, 2010.

[15] Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108, 2020.

[16] Rujun Han, Hong Chen, Yufei Tian, and Nanyun Peng. Go back in time: Generating flashbacks in stories with event temporal prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1450–1470, Seattle, United States, July 2022. Association for Computational Linguistics.

[17] Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. Doc: Improving long story coherence with detailed outline control. *arXiv preprint arXiv:2212.10077*, 2022.

[18] Jonathan D. Chang, Kiante Brantley, Rajkumar Ramamurthy, Dipendra Misra, and Wen Sun. Learning to generate better than your llm, 2023.

[19] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

[20] Xiangyu Peng, Kaige Xie, Amal Alabdulkarim, Harshith Kayam, Samihan Dani, and Mark O. Riedl. Guiding neural story generation with reader models, 2022.

[21] Louis Castricato, Alexander Havrilla, Shahbuland Matiana, Michael Pieler, Anbang Ye, Ian Yang, Spencer Frazier, and Mark Riedl. Robust preference learning for storytelling via contrastive reinforcement learning, 2022.

[22] Amal Alabdulkarim, Winston Li, Lara J. Martin, and Mark O. Riedl. Goal-directed story generation: Augmenting generative language models with reinforcement learning, 2021.

[23] Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. Controllable neural story plot generation via reward shaping. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, IJCAI-2019. International Joint Conferences on Artificial Intelligence Organization, August 2019.

[24] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[25] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020.

[26] Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. Controllable generation from pre-trained language models via inverse prompting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2450–2460, New York, NY, USA, 2021. Association for Computing Machinery.

[27] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 2022.

[28] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[30] Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 89–97, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[31] Andy Coenen, Luke Davis, Daphne Ippolito, Emily Reif, and Ann Yuan. Wordcraft: a human-ai collaborative editor for story writing, 2021.

[32] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. Talebrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

[33] Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. Co-writing screenplays and theatre scripts with language models: An evaluation by industry professionals, 2022.

[34] Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, Shruti Singh, Brent Harrison, Murtaza Dhuliawala, Pradyumna Tambwekar, Animesh Mehta, Richa Arora, Nathan Dass, et al. Improvisational storytelling agents. In *Workshop on Machine Learning for Creativity and Design (NeurIPS 2017)*, volume 8, 2017.

[35] Timothy S Wang and Andrew S Gordon. Playing story creation games with large language models: Experiments with gpt-3.5. In *International Conference on Interactive Digital Storytelling*, pages 297–305. Springer, 2023.

[36] Zhiyu Lin and Mark O. Riedl. Plug-and-blend: A framework for plug-and-play controllable story generation with sketches. In *Artificial Intelligence and Interactive Digital Entertainment Conference*, 2021.

[37] Liang Wang, Nan Yang, and Furu Wei. Learning to retrieve in-context examples for large language models, 2024.

[38] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning, 2024.

[39] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan,

Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024.

[40] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023.

[41] Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3345–3355, 2024.