

STORYWRITER: A Multi-Agent Framework for Long Story Generation

Haotian Xia*
Tsinghua University
Beijing, China
xiaht24@mails.tsinghua.edu.cn

Hao Peng*
Tsinghua University
Beijing, China
peng-h24@mails.tsinghua.edu.cn

Yunjia Qi
Tsinghua University
Beijing, China
qyj23@mails.tsinghua.edu.cn

Bin Xu†
Tsinghua University
Beijing, China
xubin@tsinghua.edu.cn

Juanzi Li
Tsinghua University
Beijing, China
lijuanzi@tsinghua.edu.cn

Lei Hou
Tsinghua University
Beijing, China
houlei@tsinghua.edu.cn

Xiaozhi Wang
Tsinghua University
Beijing, China
wangxz20@mails.tsinghua.edu.cn

Abstract

Long story generation remains a challenge for existing large language models (LLMs), primarily due to two main factors: (1) discourse coherence, which requires plot consistency, logical coherence, and completeness in the long-form generation, and (2) narrative complexity, which requires an interwoven and engaging narrative. In this paper, we present STORYWRITER, a modular and open-source multi-agent framework for controllable and scalable long story generation. We conduct both human and automated evaluation, and STORYWRITER significantly outperforms existing story generation baselines in both story quality and length. Furthermore, we use STORYWRITER to generate a dataset, which contains about 6,000 high-quality long stories, with an average length of 8,000 words. We train the model Llama3.1-8B and GLM4-9B using supervised fine-tuning on LONGSTORY and develop STORYWRITER_{LLAMA} and STORYWRITER_{GLM}, which demonstrates advanced performance in long story generation. All code, models, and data are made publicly available¹ to encourage further development.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; **Natural language processing**.

Keywords

Story generation, Large Language Model, Multi agent

ACM Reference Format:

Haotian Xia, Hao Peng, Yunjia Qi, Bin Xu, Juanzi Li, Lei Hou, and Xiaozhi Wang. 2025. STORYWRITER: A Multi-Agent Framework for Long Story Generation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, November 10–14, 2025, Seoul, Republic of Korea. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3746252.3761616>

1 Introduction

Story generation aims to automatically produce coherent, organized, and engaging narratives [25]. Typically, story generation involves using a premise, often a brief beginning or theme, as input to create a complete narrative [1]. Since the emergence of large language models (LLMs; 20), the quality of generated stories using LLMs has steadily improved [28]. However, generating long stories, particularly those exceeding 1,000 words, remains a significant challenge for LLMs [16].

The main challenges of long story generation are from two aspects: (1) discourse coherence, which requires plot consistency, logical coherence, and completeness in long-form generation. Existing LLMs still face challenges in generating fluent long texts [14]. In long story generation, LLMs need to retain long-distance key information, such as events, characters, and their relationships, to ensure plot consistency across the narrative. (2) narrative complexity, which requires interwoven, engaging, and diverse story content. While human-written stories typically exhibit these characteristics, LLM-generated narratives are often homogeneous, lacking in diversity and plot development [21, 22].

To address the above challenges, we propose STORYWRITER, a multi-agent framework for long story generation, which consists of three main modules: (1) **outline agent**, which generates event-based outlines. Generating outlines is a typical procedure in story generation, previous studies adopt LLMs to directly generate outlines [22, 26, 29], which may be insufficiently specific and diverse. Inspired by conventional event knowledge [24], we adopt an agent to generate a detailed event graph, where each node represents an event, and edges represent relationships between

*Equal contribution.

†Corresponding author.

¹<https://github.com/Summer17xia/StoryWriter>



This work is licensed under a Creative Commons Attribution 4.0 International License. *CIKM '25, Seoul, Republic of Korea*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2040-6/2025/11

<https://doi.org/10.1145/3746252.3761616>

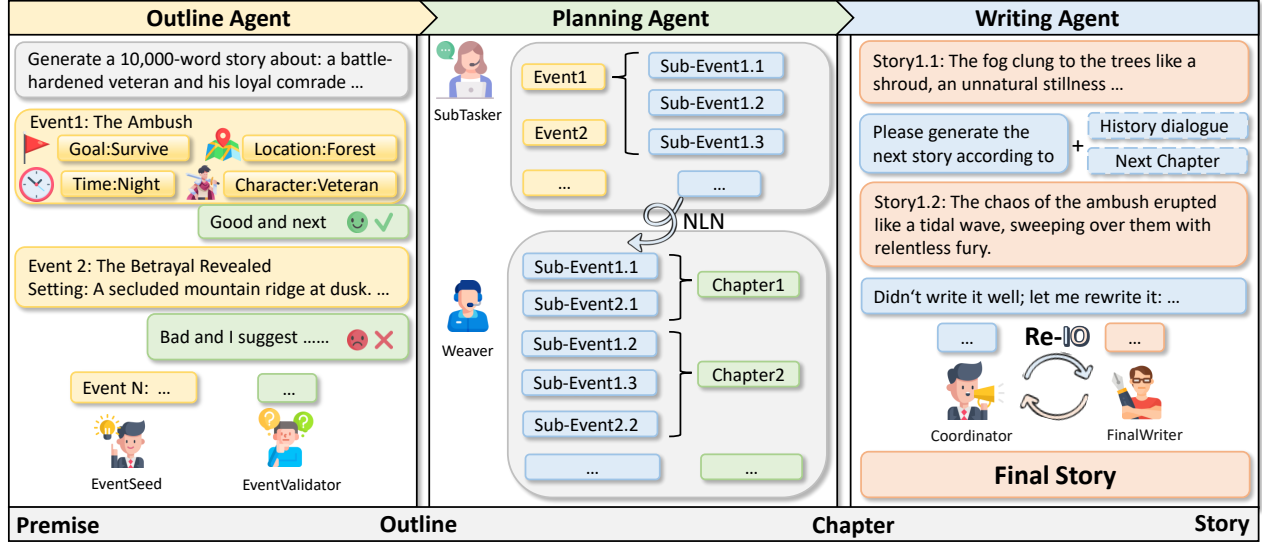


Figure 1: An overview of the three-stage story generation framework. The process consists of (from left to right): (1) event-based outline generation by the Outline Agent, (2) chapter construction using Non-Linear-Narration (NLN) by the Planning Agent, and (3) final story synthesis via ReIO (Re-write Input and Output) by the Writing Agent. Each stage employs a distinct methodology to progressively refine the narrative from high-level structure to detailed, coherent story text.

events, such as causal relationships [23]. Each event is associated with several characters [24]. We then adopt an agent to validate the consistency of each event and produce the final outline. (2) **planning agent**, which generates detailed sub-events and globally plans which events should appear in each chapter to maintain an interwoven and engaging story. Specifically, we first use LLMs to generate sub-events for each event to provide richer event information. Human writing is non-linear, with events and characters often linked in diverse ways across different chapters [2, 17]. We also employ an LLM to globally plan which events and characters should appear in each chapter, ensuring consistency and enabling the reappearance of key elements across chapters. (3) **writing agent**, which generates and refines specific story content based on the historical context. Long story generation involves long-range dependencies and directly feeding the entire history to the LLM may result in missing key information [13], we adopt an agent named Coordinator to dynamically compress the previous writing history based on the current event. The goal of compression is to retain only relevant events and create a compact and effective writing history for generating a more coherent story. We then input this history with an event requiring expansion to the final writer to generate a sub-story, and then refine it using the Coordinator.

We conduct extensive experiments to validate the effectiveness of STORYWRITER. We adopt GPT-4o-mini [18] as the backbone to implement STORYWRITER. We conduct evaluation on the widely used MoPS dataset [15]. We also investigate several strong baselines, including DOC [30], Agents' Room [12], and GPT-4o-mini [18]. We adopt both human evaluation and GPT-4o-based automated evaluation across 6 commonly used dimensions[6], including relevance, coherence, empathy, surprise, creativity, and complexity. STORYWRITER significantly outperforms other models, demonstrating its

effectiveness. Finally, we adopt STORYWRITER to generate a training dataset, LONGSTORY, which contains about 6,000 stories with an average length of 8,000 words. We fine-tune the Llama3.1-8B Instruct model [7] using supervised fine-tuning on LONGSTORY to develop STORYWRITER_{LLAMA}. We evaluate the trained model using LongWriter-Ruler and LongBench-Write [4], and find that STORYWRITER_{LLAMA} significantly outperforms Llama3.1-8B Instruct on story exceeding 2,000 words, and even surpasses GPT-4o [19]. This demonstrates the effectiveness of LONGSTORY.

2 STORYWRITER

STORYWRITER is a modular, multi-agent framework designed to facilitate long-form story generation, shown in Figure 1. Inspired by human writing workflows, we decompose the generation process into three coordinated stages, each handled by specialized agents: the *Outline Agent*, the *Planning Agent*, and the *Writing Agent*. Each module is implemented as a callable LLM interface with task-specific prompting and memory compression logic. The design prioritizes scalability, controllability, and reusability across different LLM backbones.

Outline Agent. The Outline Agent is responsible for constructing an event-centric skeleton of the story. Given a user-provided premise, it generates a sequence of structured event tuples, each annotated with temporal, causal, and character attributes. A consistency checker module filters logically incoherent or redundant events. This structured outline provides a semantically rich input for downstream planning.

Planning Agent. The Planning Agent expands high-level events into sub-events and assigns them to story chapters using a non-linear narration (NLN) strategy. Inspired by narratological theories

(e.g., analepsis and prolepsis), the planner allows out-of-order but causally consistent event allocation, increasing narrative engagement [10, 11]. This agent operates globally across the event graph and chapter layout to maintain inter-event coherence and thematic recurrence.

Writing Agent. The Writing Agent generates detailed story text based on the planned sub-events. To mitigate the limitations of long-context attention in LLMs, it uses a *ReIO* (Rewrite Input and Output) mechanism. A coordination module compresses story history by extracting relevant contextual spans before generation. After drafting, the output is optionally rewritten to ensure consistency with the original outline and style uniformity.

All agents are orchestrated via the AutoGen [27] framework. Agents are stateless and modular, supporting asynchronous execution, replacement, or fine-tuning. The architecture enables STORYWRITER to scale across model sizes and use cases, while maintaining controllable and interpretable outputs.

3 Experiments

3.1 Experimental Setup

Evaluation Datasets. We use the dataset MoPS [15]. They provide the MoPS code suite, along with 7.6k generated premises and 1,000 extended stories. Compared to premises generated by conventional methods and those collected from literary forums like WRITING-PROMPTS [8], the stories generated by MoPS exhibit higher quality and greater information density.

Evaluation Setup. We adopt the evaluation framework proposed by HANNA [5], a benchmark for story assessment, with minor adaptations to certain dimension definitions. This framework specifies six orthogonal criteria—Relevance, Coherence, Empathy, Surprise, Creativity, and Complexity—each grounded in social science literature. To comprehensively assess the generated stories, we employ both human and automated evaluation. For human evaluation, anonymized outputs are distributed to graduate students in an English program (all with TOEFL scores of 108 or above), who rate the stories on a five-point Likert scale across the six dimensions. For automated evaluation, we utilize GPT-4o [19], which assigns integer scores from 1 to 5 for each dimension. This dual evaluation protocol ensures a robust and multifaceted assessment of narrative quality.

Baselines. We compare stories generated by two methods DOC [30] and Agents’ Room [12]:

(1) **DOC.** A method designed to enhance text quality by generating more comprehensive outlines. (2) **Agents’ Room.** Agents’ Room is a multi-agent framework for story generation. This approach introduces an orchestrator responsible for determining when to invoke the writer agent and the planner agent, thereby ensuring coordinated execution among agents. (3) **GPT-4o mini.** We directly input the premise into GPT-4o-mini to generate the story.

3.2 Experimental Results

Main Results. All the experimental results are presented in Table 1. We observe the following: (1) In general, our story generation

framework STORYWRITER significantly outperforms the baselines in both human and automated evaluations, demonstrating its effectiveness. (2) STORYWRITER significantly surpasses previous baselines in terms of length while maintaining high generation quality, indicating its effectiveness in generating longer stories. (3) Across different specific evaluation dimensions, our method outperforms DOC and GPT-4o-mini in relevance and coherence, slightly falling behind Agents’ Room. This may be due to that STORYWRITER generates longer stories, and coherence inevitably decreases with increased length [4]. However, in terms of content diversity and creativity, our model significantly outperforms all baselines, validating the effectiveness of our approach and demonstrating that it can generate higher-quality, creative content, which is the ultimate goal of story generation.

4 Constructing LONGSTORY

In this section, we use STORYWRITER to generate a high-quality long story dataset LONGSTORY. We train the model Llama3.1-8B and GLM4-9B using supervised fine-tuning on LONGSTORY and develop advanced storytelling LLM STORYWRITER_{LLAMA} and STORYWRITER_{GLM}. Our dataset shows significant improvement in doing sft on multiple downstream models.

LONGSTORY Construction. We construct a high-quality dataset with 5,500 long-form stories, LONGSTORY, using STORYWRITER. Specifically, we first collect 6,000 story premises from the training set of MoPS [15] and use STORYWRITER to generate a long story for each premise. We then perform careful data cleaning to remove stories that are too short, do not meet format requirements, or exhibit low quality. Specifically, we merge multiple chapters of stories to mitigate the risk of overfitting to specific text structures during SFT training. As a result, we curate a final dataset comprising 5,500 long stories, LONGSTORY, with an average length of about 8,000 words.

Experimental Setup. We adopt the same evaluation dataset MoPS in § 3.1. Due to the high cost of the manual evaluation, we only employ automated evaluation, which is also widely used in previous work [4, 9]. In addition to evaluating the content quality from 6 dimensions mentioned in § 3.1, we also report the length score used by the LongBench-Write evaluation method [4]. This method controls the length of text generated by LLMs by setting different output length constraints, which not only assesses the model’s ability to generate long texts but also evaluates its adherence to word count constraints. The length score computes the degree of alignment between the actual response length and the required length in the instruction, which can be computed as follows:

$$S_l = \begin{cases} 100 \cdot \max\left(0, 1 - \frac{(l'/l-1)}{3}\right) & \text{if } l' > l, \\ 100 \cdot \max\left(0, 1 - \frac{(l/l'-1)}{2}\right) & \text{if } l' \leq l. \end{cases} \quad (1)$$

l' denotes the actual response length and l denotes the required length. Specifically, we adopt the same settings as LongBench-Write: for each instruction in the MoPS test set, we add an output length constraint from {500, 1,000, 2,000, 4,000, 10,000}, and then generate response for each length constraint and compute the final scores. We bucket the results based on lengths and report the average of the following metrics within each bucket: S_q , which evaluates

Table 1: Experimental results of human and automatic scoring (on a scale from 1 to 5). RE, CH, EM, SU, CR, and CX represent relevance, coherence, empathy, surprise, creativity, and complexity, respectively. Bold indicates the best result according to human evaluation, and underline indicates the best result according to automatic evaluation.

Model		Average	RE	CH	EM	SU	CR	CX	Average Length
DOC	Human-Eval	3.7	4.2	4.3	3.2	3.4	3.7	3.2	2,373
	Auto-Eval	3.9	4.1	4.3	4.0	3.5	3.8	3.5	
Agents' Room	Human-Eval	3.8	4.5	4.4	3.3	3.2	3.7	4.0	3,134
	Auto-Eval	3.9	3.5	4.5	4.0	3.7	3.9	3.7	
GPT-4o mini	Human-Eval	3.6	4.0	3.8	3.3	3.4	3.6	3.7	1,078
	Auto-Eval	3.9	4.0	<u>4.7</u>	4.1	3.5	3.7	3.4	
STORYWRITER	Human-Eval	4.2	4.4	4.3	3.8	3.6	4.3	4.8	8,081
	Auto-Eval	<u>4.2</u>	<u>4.1</u>	4.4	<u>4.4</u>	<u>3.7</u>	<u>4.2</u>	<u>4.6</u>	

Table 2: Experimental results (%) of STORYWRITER_{LLAMA}, STORYWRITER_{GLM} and the baselines. S_q represents the average score of the 6 dimensions, as described in § 3.1. S_l is the length score, calculated using Equation 1. $\bar{S}$ is computed as $(S_q + 20 \times S_l)/2$, following the approach used by Bai et al. [4]. The highest scores are shown in bold.

	Overall			[0, 1k)		[1k, 2k)				[2k, 4k)		[4k, 10k)		[10k, 20k)	
	$\bar{S}$	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q
Llama3.1-8B-Instruct	46.6	34.5	2.9	89.0	4.0	83.7	3.9	0.0	3.5	0.0	2.2	0.0	1.0		
GLM4-9B	47.3	36.6	2.9	93.7	4.2	89.6	4.0	0.0	3.3	0.0	2.0	0.0	1.0		
LongWriter-GLM4-9B	76.3	83.0	3.5	86.9	3.1	93.1	3.2	91.6	4.0	86.9	3.6	56.7	3.4		
LongWriter-Llama3.1-8B	77.9	83.6	3.6	96.9	3.9	96.1	3.5	93.2	4.1	81.9	3.5	51.3	3.2		
Deepseek-Llama-8B	70	73.6	3.3	92.3	3.1	91.9	3.2	88.2	3.6	83.2	3.4	12.3	3.3		
Deepseek-Llama-70B	74.3	79.0	3.5	93.2	3.3	94.5	3.4	87.2	4.0	81.0	3.5	44.1	3.2		
GPT-4o	67.4	52.8	4.1	92.3	4.7	91.7	4.5	62.0	4.3	15.3	3.7	2.7	3.3		
STORYWRITER_{LLAMA}	73.4	75.3	3.5	90.8	3.9	94.1	3.8	77.3	3.5	77.0	3.4	27.7	3.4		
STORYWRITER_{GLM}	83.7	88.5	3.9	99.5	4.4	99.3	4.1	98.0	4.0	88.7	3.5	57.3	3.6		

content quality (the average of the 6 scores from § 3.1), S_l , which evaluates the length score, and $\bar{S}$, which equals $(S_q + 20 \times S_l)/2$. We also report the average overall score across all lengths.

SFT Training. We leverage the Llama 3.1-8B model and GLM-4-9B model as the base model for SFT training. We use the training code proposed by LongAlign [3], as it is specifically designed for long-context training with pre-existing long-context adaptations. We use the premise of each instance in LONGSTORY as the input and the story as the output for supervised fine-tuning to obtain STORYWRITER_{LLAMA} and STORYWRITER_{GLM}. For both of which we set the batch size to 1, learning rate to 2×10^{-5} , training 4 epochs.

Experimental Results. The results of STORYWRITER_{LLAMA} and STORYWRITER_{GLM} trained on LONGSTORY, along with other baselines, are shown in Table 2. We can observe that: (1) For the quality of generated stories (S_q), STORYWRITER_{GLM} significantly outperforms the backbone model, especially in generating stories over 4,000 words. This indicates that STORYWRITER_{GLM} can maintain high quality while generating longer content. (2) For length scoring of the generated stories (S_l), our models also perform much better than Llama3.1-8B-Instruct and GPT-4o. This indicates that our models better adhere to length constraints in story generation. Although the training process does not involve explicit ability enhancement for following length constraints. This suggests that

training with longer responses could enhance the model’s ability to follow length constraints. In conclusion, STORYWRITER_{LLAMA} and STORYWRITER_{GLM} perform better in generating longer stories and adhering to length constraints, demonstrating the effectiveness of our data construction method STORYWRITER and LONGSTORY. As our approach can be extended to the broader field of creative content generation, we encourage the community to utilize it for producing more high-quality data.

5 Conclusion

This paper presents STORYWRITER, a multi-agent approach that generates outlines and long-enough stories automatically. Using STORYWRITER, we generate a large number of diverse and high-quality stories. Human and automatic evaluations demonstrate that STORYWRITER outperforms multiple baselines. Similarly, we create a high-quality dataset LONGSTORY using STORYWRITER. We also perform supervised fine-tuning based on LONGSTORY and provide STORYWRITER_{LLAMA} and STORYWRITER_{GLM}. We believe that STORYWRITER will be helpful for the long story generation task of LLM, and future Auto Story Generation(ASG) tasks can be explored based on these data and STORYWRITER_{LLAMA}. We hope to explore LLM’s generation of long serial novels further, which requires more powerful long-story generation and understanding capabilities from LLMs.

GenAI Usage Disclosure

As mentioned in § 4, we adopted GPT-4o mini to generate the dataset LONGSTORY. We also adopted GPT-4o for automated evaluation as presented in § 3.1. We also used ChatGPT to refine some sentences.

References

- [1] Arwa I Alhussain and Aqil M Azmi. 2021. Automatic story generation: A survey of approaches. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–38.
- [2] Fatma Alkaaf and Ali Al-Bulushi. 2017. Tell and write, the effect of storytelling strategy for developing story writing skills among grade seven learners. *Open Journal of Modern Linguistics* 7, 2 (2017), 119–141.
- [3] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. Longalign: A recipe for long context alignment of large language models. *arXiv preprint arXiv:2401.18058* (2024).
- [4] Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongWriter: Unleashing 10,000+ Word Generation from Long Context LLMs. *arXiv:2408.07055 [cs.CL]* <https://arxiv.org/abs/2408.07055>
- [5] Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. 2022. Of Human Criteria and Automatic Metrics: A Benchmark of the Evaluation of Story Generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 5794–5836. <https://aclanthology.org/2022.coling-1.509/>
- [6] Cyril Chhun, Fabian M. Suchanek, and Chloé Clavel. 2024. Do Language Models Enjoy Their Own Stories? Prompting Large Language Models for Automatic Story Evaluation. *arXiv:2405.13769 [cs.CL]* <https://arxiv.org/abs/2405.13769>
- [7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [8] Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for Structuring Story Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 2650–2660. doi:10.18653/v1/P19-1254
- [9] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594* (2024).
- [10] David Herman. 2002. *Story Logic: Problems and Possibilities of Narrative*. University of Wisconsin Press, Madison, WI.
- [11] David Herman. 2017. Narratology’s Union with Cognitive Science—A Review of David Herman’s Narrative Theory and the Cognitive Science. *World Literature Studies* 5, 3 (2017), 13–24. doi:10.12677/WLS.2017.53013
- [12] Fantine Huot, Reinald Kim Amplayo, Jennimaria Palomaki, Alice Shoshana Jakobovits, Elizabeth Clark, and Mirella Lapata. 2024. Agents’ Room: Narrative Generation through Multi-step Collaboration. *arXiv:2410.02603 [cs.CL]* <https://arxiv.org/abs/2410.02603>
- [13] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.
- [14] Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. 2024. LongGenBench: Long-context Generation Benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 865–883.
- [15] Yan Ma, Yu Qiao, and Pengfei Liu. 2024. MoPS: Modular Story Premise Synthesis for Open-Ended Automatic Story Generation. *arXiv:2406.05690 [cs.CL]* <https://arxiv.org/abs/2406.05690>
- [16] Aleksandr Migal, Daria Seredina, Ludmila Telmina, Nikita Nazarov, Anastasia Kolmogorova, and Nikolay Mikhaylovskiy. 2024. Overview of Long Story Generation Challenge (LSGC) at INLG 2024. In *Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges*. 47–53.
- [17] John W Oller Jr. 1983. Story writing principles and ESL teaching. *Tesol Quarterly* 17, 1 (1983), 39–53.
- [18] OpenAI. 2024. GPT-4o Mini: Advancing Cost-Efficient Intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/> Accessed: 2025-02-04.
- [19] OpenAI. 2024. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o/> Accessed: 2025-02-04.
- [20] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.
- [21] Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are Large Language Models Capable of Generating Human-Level Narratives?. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 17659–17681.
- [22] Qianyu Wang, Jinwu Hu, Zhengping Li, Yufeng Wang, Yu Hu, Minghui Tan, et al. 2024. Generating Long-form Story Using Dynamic Hierarchical Outlining with Memory-Enhancement. *arXiv preprint arXiv:2412.13575* (2024).
- [23] Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, et al. 2022. MAVEN-ERE: A Unified Large-scale Dataset for Event Coreference, Temporal, Causal, and Subevent Relation Extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 926–941.
- [24] Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, et al. 2023. Maven-arg: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. *arXiv preprint arXiv:2311.09105* (2023).
- [25] Yuxin Wang, Jieru Lin, Zhiwei Yu, Wei Hu, and Börje F Karlsson. 2023. Open-world story generation with structured knowledge enhancement: A comprehensive survey. *Neurocomputing* (2023), 126792.
- [26] Yichen Wang, Kevin Yang, Xiaoming Liu, and Dan Klein. 2023. Improving Pacing in Long-Form Story Planning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 10788–10845.
- [27] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv:2308.08155 [cs.AI]* <https://arxiv.org/abs/2308.08155>
- [28] Kaige Xie and Mark Riedl. 2024. Creating Suspenseful Stories: Iterative Planning with Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2391–2407.
- [29] Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving Long Story Coherence With Detailed Outline Control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3378–3465.
- [30] Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving Long Story Coherence With Detailed Outline Control. *arXiv:2212.10077 [cs.CL]* <https://arxiv.org/abs/2212.10077>