The University of Hong Kong

Department of Computer Science

CAES9542

Final Year Project Progress Report 1

Structured story planning via fine-grained breakdown of facts on the outline

This is just an incomplete report to show the progress of the work

Zhiheng Lyu 3035772432

Due Date: Oct. 25, 2023

Submitted: Oct. 25, 2023

Abstract

This report presents a progress update on a final year project focused on enhancing the controllability and interpretability of Large Language Models (LLMs) in story outline planning. Addressing the challenges of aligning generated content with user-defined outlines, our project introduces a system that encourages active user participation in narrative construction. Through a combination of prompt learning, fine-tuning, and the utilization of smaller language models for specific NLP tasks, we aim to offer a nuanced, specialized solution for story outline generation. Our methodology incorporates a granular breakdown of narrative elements and a novel algorithmic approach, blending the expressiveness of natural language with the precision of formal logic. By meticulously managing facts within valid timeline intervals and leveraging diverse model scales, we strive to enhance narrative coherence and minimize runtime overheads. This report encapsulates the development journey, highlighting significant progress made toward creating a collaborative, coherent, and logically structured narrative generation system.

Contents

List of Figures

1 Introduction

This progress report documents the development of a LLM-based novel **story planning system**, outlining objectives, methodology, and progress to date. Section 1 introduces the project, detailing its motivation and implementation strategy, including the use of both standardized and non-standardized NLP tasks. Related work in NLP, Large Language Models, and story generation is reviewed in Section 2. The current state of the project, including data structure design and system architecture, is presented in Section 3. Future work and evaluation strategies are discussed in Section 4, with a summary of key achievements in Section 5. Supporting materials are provided in the Appendix.

1.1 Project Motivation and Objective

In the dynamic field of computational linguistics, Large Language Models (LLMs) have shown unparalleled prowess in text generation, producing outputs nearly indistinguishable from human writing. Despite these advancements, the challenge persists in aligning the generated content precisely with user-defined story outlines and intentions. Our project seeks to bridge this gap, focusing specifically on story outline planning. The primary objectives are:

1. **Enhanced Controllability for Story Outlines:** The goal is to provide users with the ability to shape and guide the development of story outlines effectively. Through user-friendly tools and interfaces, we aim to enable active user participation in crafting the narrative structure, ensuring the LLM adheres closely to the desired storyline and thematic elements.

2. **Improved Interpretability of Narrative Elements:** We strive to ensure that each segment of the generated story outline is not just coherent but also logically connected to the overall narrative. The project is committed to enhancing the model's ability to generate story outlines that are not only internally consistent but also align seamlessly with user expectations and the narrative's contextual flow.

By concentrating on these areas, our project transitions from generic text generation to a more specialized domain of story outline planning. We aim to create a system where the LLM becomes a collaborative tool in the creative process, helping users to construct story outlines that are coherent, contextually relevant, and true to their envisioned narrative trajectory.

1.2 Project Methodology

In our work, we blend advantaged techniques with adaptive computing to solve a wide range of language tasks. We use large language models for complex problems and smaller models for common ones, choosing Python for its ease of use. The annotation and testing by human preference is also used as a fair approach for our project.

Implementation of Non-standardized Tasks In tackling creative and complex tasks such as outline planning [1] and factual decomposition [2], we leverage prompt learning on large language models fine-tuned via RLHF. This approach allows us to navigate unique challenges by crafting precise prompts, aiming for high-performance outcomes.

Implementation of Standardized NLP Tasks For standardized tasks like similarity retrieval [3] and contradiction classification [4], we utilize smaller language models from HuggingFace [5], balancing performance and efficiency. Our methodology involves pre-training and fine-tuning using LLMs data, achieving resource savings without compromising accuracy.

Programming Language Selection Given the performance considerations of our project's logical layer, our evaluation led to Python as the ideal choice. Its simplicity, flexibility, and vast ecosystem align with our project's needs, ensuring easy integration and adaptability.

Annotation and Testing For high-quality results in non-standardized tasks, we employ the ”Potato Annotation” platform [6], combining automation and manual expertise. Our expert annotators, compensated at $10 per hour, are instrumental in validating and refining our methodologies to align with our goals.

2 Related Work

In the field of NLP, the evolution of Large Language Models (LLMs) like ChatGPT has significantly advanced human-computer interaction, despite challenges in long-form text generation. Research has focused on enhancing narrative creation, with tools like DOC improving the structure and flow of extended texts, yet issues with consistency and coherence persist, which is our research focus about.

2.1 Natural Language Processing (NLP) and Large Language Models(LLMs)

Natural Language Processing (NLP) is a prominent subfield of artificial intelligence dedicated to teaching computers to interpret and generate human language. Over the years, NLP has witnessed remarkable advancements, enabling a wide array of applications ranging from machine translation and sentiment analysis to the development of intelligent chatbots. Within this domain, Large Language Models (LLMs) like ChatGPT[7] have emerged as groundbreaking innovations, signifying a major shift in the NLP landscape. These models are engineered to digest and process enormous datasets, paving the way for enhanced human-computer interactions. Despite their prowess, LLMs face inherent challenges, especially when tasked with long-form text generation. This includes difficulties in consistently retrieving relevant information over extended passages and planning the structure of the text to maintain coherence and logical flow.

2.2 Long-Form Text Planning and Story Generation

Story generation, as an application of long-form text planning and generation, holds a paramount position in the realm of NLP. The task of generating captivating narratives is complex, often requiring meticulous planning and structured approaches rather than ad-hoc improvisations. A well-crafted story demands not only consistency in its theme but also a coherent structure that ties together its various elements seamlessly.

To aid in this intricate process, several research initiatives have been developed, with tools like DOC standing out. DOC is designed to simplify the challenges of long-form text generation by leveraging a detailed outline approach. This methodology ensures a degree of controllability over the narrative's progression. However, while it offers a structured framework, DOC [1] is not without its limitations. Its dependence on direct sampling, without adequate mechanisms to guide the generation process, can sometimes result in contradiction and inconsistencies in the narrative. This can manifest as repetitive scenarios or disjointed story arcs, which detract from the overall cohesiveness and quality of the generated content.

3 Currrent Procedure

This section presents a comprehensive overview of the current procedures and engineering choices employed in the development of our story planning system. It delves into the intricate data structure design, detailing the relationships and interactions between story elements such as outline, plot, facts, and world status. Additionally, it outlines the modular architecture of the system, discussing the functionalities and implementation strategies of each module, from outline generation and axiomatic fact decomposition to contradiction detection and plot correction. The engineering decisions made throughout the development process are highlighted, showcasing the balance between efficiency, accuracy, and narrative coherence in our approach.

3.1 Data Structure Design of Story Outline, Plot, Facts and World Status

To meticulously plan the story outline, an in-depth analysis of the structure and interrelations of various narrative elements is imperative. This section delves into the intricacies of data structure design, exploring the delineation and connections between story outline, plot, facts, and world status to ensure a coherent and engaging storytelling experience.

3.1.1 Event

In narrative theory, an 'event' is a crucial unit that encapsulates changes in the world as portrayed in the story, affecting characters, settings, or the overall situation. These events are organized in a causal and chronological manner, ensuring a seamless narrative flow. Our algo-

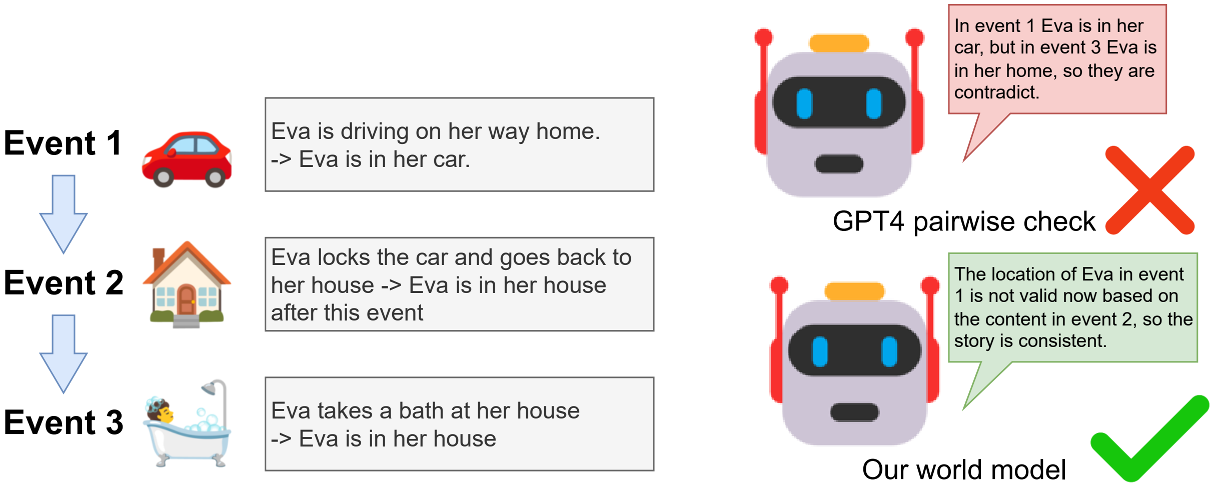
Event 1
Event 2
Event 3
Eva is driving on her way home. -> Eva is in her car.
Eva locks the car and goes back to her house -> Eva is in her house after this event
Eva takes a bath at her house -> Eva is in her house
In event 1 Eva is in her car, but in event 3 Eva is in her home, so they are contradict.
GPT4 pairwise check
The location of Eva in event 1 is not valid now based on the content in event 2, so the story is consistent.
Our world model

Figure 1: Pairwise Contradiction v.s. Fact-based Contradiction Detection

rithm specifically addresses conflicts within pairs of events derived from a hierarchical outline, dissecting each event into subevents during the generation phase to analyze and resolve these discrepancies.

3.1.2 Pre-fact, Post-fact, and Static-fact

Guided by methodologies in fact verification [**?**], we deconstruct events into atomic facts for pairwise comparison. Recognizing that events symbolize shifts in world status, we categorize these facts into three types: ’Pre-facts’ exist before the event, ’Post-facts’ prevail afterwards, and ’Static-facts’ are perpetually valid. This decomposition allows for a granular and precise analysis of narrative structures.

3.1.3 Narrative World Status

’World Status’ in a narrative refers to the set of concurrent truths at a specific moment, highlighting that facts may have temporal validity due to event-induced changes. Thus, the World Status at any point is the ’Maximum Fact Set,’ containing all non-conflicting facts, with precedence

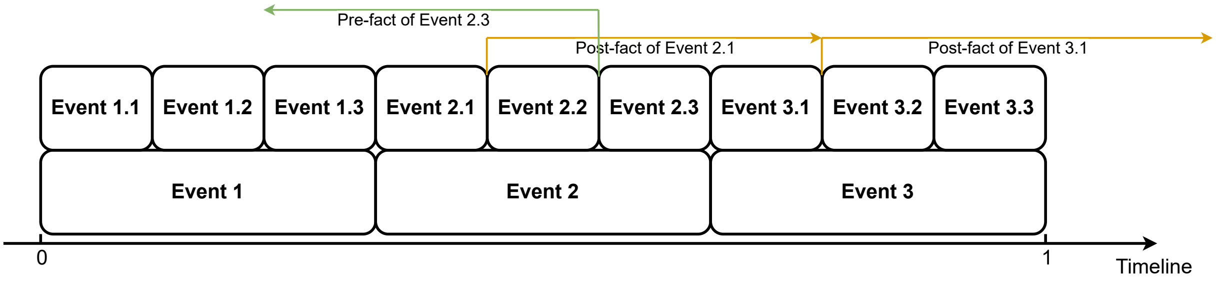

Figure 2: The Story Structure on timeline

given to the most recent facts in cases of contradiction. This framework ensures consistency and coherence in tracking the evolution of the narrative world.

3.1.4 Timeline and Valid Interval

As shown in 2, We model the temporal structure of narratives using continuous time intervals for events, ensuring that subevents are nested within parent events and adjacent events do not overlap. The entire story is set within the normalized timeline of $[0,1]$. When adding k subevents within the interval $[l,r]$, we evenly divide the interval, adjusting slightly by ε to avoid shared boundaries.

Facts are associated with valid time intervals, with pre-facts defaulting to $(-\infty,l]$ and post-facts to $[r,\infty)$ within an event's interval $[l,r]$. Static-facts combine both intervals. To address contradictions, we update the intervals of conflicting facts, giving precedence to the most recent information. A fact is identified as contradicting when its interval overlaps with both a pre-fact and a post-fact, signaling a need for adjustment.

3.2 Engineer Choices for Story Planning System Development

Our system, depicted in Figure 3, comprises four distinct modules. Initially, the Generation module employs Breadth-First Search (BFS) to create an Outline Plot, which is subsequently decomposed into various types of atomic facts. These facts, in conjunction with the current per-

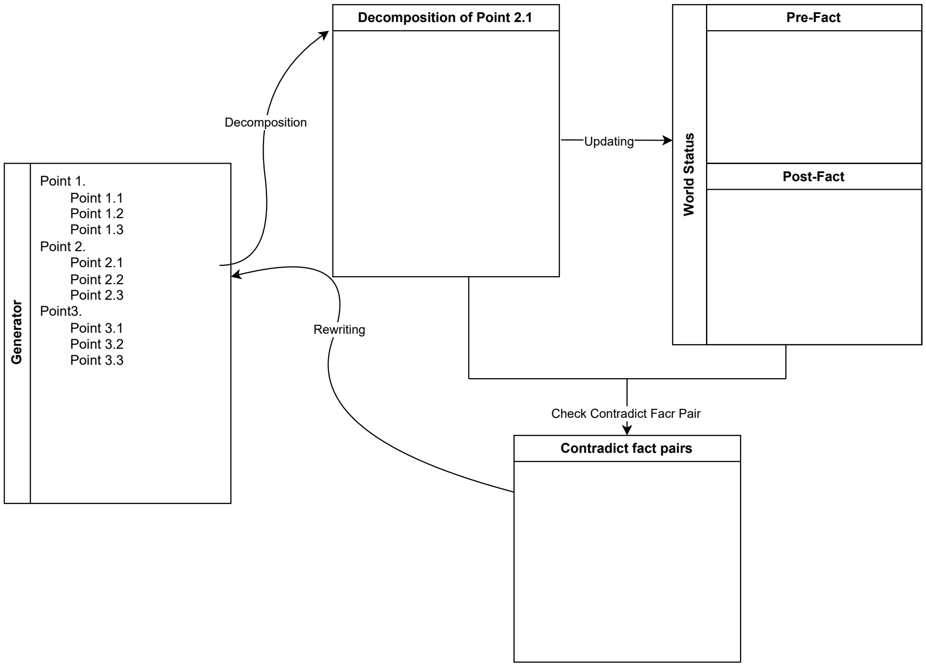

Figure 3: The 4 Module Structure about the Full Pipeline

sistent world state, undergo verification. If they pass the verification, the process moves forward; however, any discrepancies lead to a rewriting of the current plot. This design ensures a close-loop workflow, integrating generation, decomposition, and validation to maintain consistency and accuracy in the narrative structure.

3.2.1 Outline Generation

Our approach to outline generation leverages a tree of events, iteratively splitting each event into several lower-level events. Adopting similar settings to the DOC approach [**?**], our procedure initiates the expansion process from higher-level events, given that they encapsulate more general story information. This breadth-first strategy promotes the generation of higher-level events prior to their lower-level counterparts, and the our method can also easily be adapted to alternative techniques such as depth-first search (DFS).

3.2.2 Axomic Fact Decomposition

We employ Axomic Fact Decomposition to dissect plot points into pre-fact, post-fact, and static-fact components, utilizing the Zero-Shot capabilities of GPT-4 enhanced through Reinforcement Learning from Human Feedback (RLHF). The structured template for this task is detailed in Table 6.2.3. Despite exploring Few-Shot learning, we observed superior performance with Zero-Shot, attributing this to the RLHF process that potentially makes GPT-4 more adept at tasks without explicit examples. This engineering choice is grounded in optimizing for efficiency and effectiveness in the decomposition process.

3.2.3 Contradict Detection

In our contradiction detection pipeline, GPT-4 demonstrates high accuracy in identifying contradictions between fact pairs, even in ambiguous cases. However, the Contradiction Classification stage's O(n^2) complexity creates a bottleneck, contrasting with the linear complexity of other stages.

To optimize our pipeline, we integrated and finetuned a pre-trained Natural Language Inference (NLI) model from Hugging Face (`MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli`), using GPT-4's predictions. This process involved generating 60 outlines from the WritingPrompts dataset, partitioning them into 50 for training and 10 for testing, and working with a training set of 856,748 fact pairs. Utilizing both the original and a preliminarily trained NLI model, we conducted stratified sampling to select 20,000 diverse fact pairs. After deduplication, 18,702 unique pairs remained and were annotated with GPT-4, resulting in an estimated 2.98% rate of contradictions.

For evaluation, we re-ranked the top 10% of fact pairs in the test set and divided them into deciles, sampling 100 data points from each to calculate the positive rate. With a threshold set at 3%, our model achieved a balanced performance, with both precision and recall at 60%.

In this setup, the Hugging Face NLI model acts as a high-recall filter, effectively reducing the workload for GPT-4, which then makes the final decision with high precision. This strategy ensures efficiency without compromising accuracy, allowing for effective contradiction detection even in cases of vague information.

3.2.4 Plot Correction

In this phase, we address contradictions between the current and world facts, employing prompt-based fact injection to revise the current event and mitigate discrepancies, as outlined in Table 6.2.4. The process is designed to be sensitive to contradictions; however, to prevent excessive rewriting, we cap the iterations at five. Beyond this limit, the revision is deemed beneficial in reducing ambiguity, even if some contradictions remain. The new plot is generated as

$$Plot_{new} = f(Plot_{old}, \forall(fact_i, status(fact_i))) \tag{1}$$

4 Future Work

Building upon the promising results attained in empirical analyses, our next steps involve a comparative study to underscore the advantages of our algorithm over prevalent models such as GPT-4 Zeroshot and DOC. We aim to leverage human annotators from Prolific.co for an outsourced, detailed data annotation and evaluation process. This will not only provide a richer understanding of our system's efficacy but also ensure a grounded comparison with existing technologies. Simultaneously, we plan to enhance user interaction by developing a graphical user interface for our system, integrating additional functionalities to facilitate a more seamless and intuitive human-computer interaction. These initiatives are geared towards refining our system, making it more accessible and user-friendly, while continuously pushing the boundaries of narrative generation and coherence.

5 Conclusions

In the realm of hierarchical generation systems like DOC, there have been persistent challenges, notably issues of redundancy, contradictions in facts, and incoherence in the generated text. Addressing these issues head-on, this project embarked on developing a system that seamlessly integrates natural and formal languages, applying innovative algorithms to mitigate these problems. This report has delineated the comprehensive development and testing phases of this novel system.

Our methodology was rooted in a detailed analysis of the structure of story outlines. We pioneered a new algorithmic approach that combines the expressiveness of natural language with the precision of formal logic. In a groundbreaking shift, events were decomposed into more granular facts, each meticulously maintained within valid intervals on a timeline. This approach, supplemented by a mix of symbolic computation and models of varying scales, significantly reduced runtime overheads, yielding promising results.

The empirical evidence gathered through this project indicates a marked improvement in the coherence of story planning. Nevertheless, it is pertinent to note that a thorough evaluation, particularly involving human annotation, is imperative for a more comprehensive assessment of our system's effectiveness. Our innovative decomposition of events and the maintenance of valid intervals stand as testament to the system's potential in enhancing narrative structure and consistency.

As we navigate the complexities of combining natural and programming languages, this project has laid down a robust foundation, opening avenues for more refined and user-centric narrative generation systems. The journey from incoherence and contradiction to a more structured and logical narrative generation is fraught with challenges, but the strides made in this project are a promising step towards overcoming these hurdles.

References

[1] Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 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)*, pages 3378–3465, Toronto, Canada, July 2023. Association for Computational Linguistics.

[2] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit

Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

[3] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[5] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[6] Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. POTATO: The portable text annotation tool. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.

[7] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

6 Appendix

6.1 A: Psudo Code for Mixed algorithms

Algorithm 1 Filter Condition

```
Require: pre-fact: $P$, valid fact list: $p$
Require: counter of facts: $c$, local nli threshold: $t$
Ensure: whether there we want to check: $flag$
  $flag \leftarrow NLI(P, p) > t$
  if flag then
      $count \leftarrow count + 1$
      if count  mod  k = 0 then
          $t \leftarrow t * 2$
      end if
  end if
```

6.2 B: Prompt used in GPT4

6.2.1 Outline Generation

6.2.2 Contradict Classification

6.2.3 Fact Decomposition

6.2.4 Fact Injection

Algorithm 2 Get valid interval for a pre-fact

```
Require: world status: $W = [\forall i, (p_i, l_i, r_i)]$
Require: pre-fact: $P$, init time: $T$
Ensure: valid interval: $(L, R]$
  $L, R \leftarrow -\inf, T$
  $W \leftarrow Sort(W, r_i > r_j)$
  for $(p, l, r) \in W, r < R$ do
    if Filter(p, P) and Contradict(p, P) then
      $L \leftarrow r$
      break
    end if
  end for
```

{partial_outline}

Can you break down point **{idx}** into up to **{bandwidth}** independent, chronological and similarly-scoped sub-points? Also list the names of characters that appear. Please follow the template below with ”Main Plot” and ”Characters”. Do not answer anything else.

Point **{idx}**.1
Main Plot: [plot event]
Characters: [character names]

Point **{idx}**.2
Main Plot: [plot event]
Characters: [character names]

...

Algorithm 3 Check whether current fact is contradict with world status

```
Require: world status: $W = [\forall i, (p_i, l_i, r_i)]$
Require: pre-fact: $P$, valid interval: $(L, R]$
Ensure: whether there is a contradict: $flag$
  $L, R \leftarrow -\inf, T$
  $W' \leftarrow [\,]$
  $flag \leftarrow False$
  for $(p, l, r) \in W$ do
      if isOverlap(l, r, L, R) then
          $W' \leftarrow W'.add(p, l, r, s = NLI(p, P))$
      end if
  end for
  $W' = Sort(W, s_i > s_j)$
  for $(p, l, r,) \in W, i < k$ do        ▷ k is a constant of maximum value of feature we want to check
      if Contradict(p, P) then
          $flag = True$
      end if
  end for
```

Do the following statements contradict each other? Answer “Yes” or “No”.

{fact1}

{fact2}

Algorithm 4 Update World Status for a pre-fact

```
Require: world status: $W = [\forall i, (p_i, l_i, r_i)]$
Require: pre-fact: $P,$ valid interval: $(L, R]$
Ensure: new world status: $W' = [\forall i, (p_i, l_i, r_i)]$
  $L, R \leftarrow -\inf, T$
  $W' \leftarrow Sort(W, l_i < l_j)$
  for $(p, l, r) \in W', R < randl < R$ do
    if Filter(p, P) and Contradict(p, P) then
      $l \leftarrow R$
    end if
  end for
  $W' \leftarrow W'.add(P, L, R)$
```

Deconstruct the given plot point into atomic facts, considering facts valid until before the plot event (pre-facts), facts valid starting after the plot event (post-facts), and facts that remain valid throughout the event (static facts). For pre-facts, identify the conditions that are present before the event, but change as a result of it. For post-facts, identify the conditions that are valid after the event, which are essentially the transformed versions of the corresponding pre-facts. Static facts are the conditions that remain true throughout the event. Please be sure to present facts as assertive statements, rather than speculative or suggestive ones.

Plot Point: **{plot_point_text}**

Pre-Facts:
[pre-facts]

Post-Facts:
[post-facts]

Static Facts:
[static facts]

Below is a Plot Point which contradicts one or more Existing Facts. Please rewrite the Plot Point to align with all Existing Facts, while keeping as much of the original information as possible and maintaining a clear and concise description.

Plot Point: **{curr_plot}**

Existing Facts:

1. **{status(fact_1)}**, **{fact_1}**
2. **{status(fact_2)}**, **{fact_2}**
3. ...

Below is a Current Plot Point. Please rewrite it to make it more consistent with the given Existing Plot Points, taking into account that the outline is structured as a tree. In this tree-like structure, individual points such as 1.1, 1.2, and 1.3 are child nodes of plot point 1. Retain as much of the original content as possible, and maintain clarity and coherence.

Current Plot Point **{curr_plot_idx}**: **{curr_plot}**

Existing Plot Points:

1. Plot Point **{plot_1_idx}**: **{plot_1}**
2. Plot Point **{plot_2_idx}**: **{plot_2}**
3. ...