Systems Reasoning with NLP for Narrative Consistency

Kharagpur Data Science Hackathon 2026 – Track A

Author: Aditya Swain
Team: Solo Participant

Abstract

Reasoning over long-form narratives remains challenging for modern NLP systems, particularly when the task requires distinguishing causal consistency from superficial textual overlap. This work presents a structured, reasoning-driven pipeline for evaluating whether character backstories are consistent with canonical novel texts. Instead of relying on end-to-end generation or opaque embeddings, the system decomposes the problem into explicit claims, evidence retrieval, and rule-based contradiction analysis. Long-context handling is achieved via selective passage retrieval from full novels ingested using the Pathway Python framework. The approach emphasizes interpretability, robustness, and evidence-grounded reasoning, aligning closely with the goals of Track A.

---

1. Introduction

Narrative consistency checking is a fundamental yet difficult task in natural language understanding. Unlike short-text classification problems, narrative reasoning requires maintaining coherence across hundreds of thousands of tokens while accounting for causality, character development, and implicit motivations.

In this hackathon task, the goal is to determine whether a structured character backstory is *consistent* or *contradictory* with the events and facts described in a long-form novel. This problem goes beyond keyword matching: a backstory may sound plausible yet contradict the original narrative in subtle but critical ways.

Rather than applying a monolithic neural model, this submission adopts a systems-reasoning perspective. The key idea is to explicitly model **what is being claimed**, **what evidence exists**, and **how contradictions arise**, thereby producing decisions that are explainable and robust.

2. Problem Formulation

Each data point consists of:

- A **book name**
- A **character name**
- A **structured backstory**
- A **binary label**: *consistent* or *contradict*

The challenge arises from:

- Extremely long source documents (entire novels)
- Implicit narrative dependencies
- Backstories that mix factual, psychological, and moral claims

The system must reason over the *entire narrative context* while remaining computationally tractable and resistant to noise.

---

3. Overall System Architecture

The proposed system is organized into four major stages:

1. **Claim Extraction**
2. **Long-Context Narrative Ingestion**
3. **Evidence Retrieval**
4. **Contradiction Scoring and Aggregation**

This modular design allows each component to be independently reasoned about and improved.

4. Claim Extraction

Rather than treating backstories as monolithic text blocks, each backstory is decomposed into a small set of **explicit claims**. These claims fall into several semantic categories:

- **Events** (e.g., traumatic childhood incidents)
- **Beliefs** (e.g., moral or ideological positions)
- **Capabilities** (e.g., skills, intelligence, physical ability)
- **Motivations** (e.g., duty, revenge, self-preservation)

Claims are represented as structured dictionaries containing:

- Claim type
- Natural language statement
- Optional metadata such as certainty or scope

This explicit representation enables targeted evidence retrieval and avoids diffuse reasoning over entire backstories.

---

5. Long-Context Handling with Pathway

The novels used in this task exceed several hundred thousand characters, making naïve full-context processing impractical.

To address this, the system uses the **Pathway Python framework** as a long-context ingestion and management layer. Full novels are ingested into Pathway-managed tables, establishing a stable document store for downstream reasoning.

While Pathway supports advanced streaming and vectorized operations, in this system it is deliberately used in a controlled manner:

- As a document ingestion layer
- As an authoritative store for narrative data
- As a clean abstraction boundary between raw text and reasoning logic

This design ensures compatibility with long documents while keeping the reasoning pipeline interpretable and reproducible.

6. Evidence Retrieval Strategy

For each claim, the system retrieves multiple potentially relevant passages from the novel. Retrieval is based on:

- Keyword overlap between claim statements and novel text
- Contextual proximity within narrative segments

Rather than relying on a single passage, the system gathers **multiple pieces of evidence** per claim. This reduces the risk of false conclusions based on isolated or ambiguous excerpts.

Importantly, retrieval is conservative: it favors recall over precision, allowing the contradiction logic to operate on a broader evidence base.

---

7. Distinguishing Reasoning Signals from Noise

One of the core challenges is separating meaningful contradictions from narrative noise. The system employs several safeguards:

1. **Backstory Support Check**
   Claims not even supported by the backstory text itself are penalized early.
2. **Multi-Passage Evaluation**
   Claims are evaluated against multiple passages, preventing single-sentence overfitting.
3. **Aggregate Decision Logic**
   Final classification is based on the total number of contradictory claims rather than isolated mismatches.

This layered filtering ensures that decisions arise from consistent patterns rather than surface-level coincidences.

8. Contradiction Scoring

Each claim–passage pair is evaluated using deterministic, interpretable rules. These rules assess whether a passage:

- Supports the claim
- Is neutral
- Explicitly contradicts the claim

Contradiction signals include:

- Direct factual negation
- Incompatible timelines
- Conflicting motivations or character traits

A claim is considered contradicted only if the aggregate evidence score falls below a threshold. A character is labeled *contradict* only if multiple claims fail consistency checks.

---

9. Results and Observations

The system demonstrates stable performance on the training data and produces balanced predictions on the test set. While the approach does not aim to maximize raw accuracy, it exhibits:

- Predictable behavior
- Robustness to noisy text
- Strong alignment with narrative logic

The outputs are reproducible end-to-end using the provided codebase, fulfilling the hackathon’s reproducibility requirements.

10. Limitations and Failure Cases

Despite its strengths, the system has several limitations:

- Claim extraction is manual and may omit nuanced narrative details.
- Rule-based contradiction logic lacks deep semantic understanding.
- No vector embeddings or neural rerankers are used.
- Subtle irony or unreliable narration may evade detection.

These trade-offs are intentional, prioritizing interpretability and correctness over aggressive modeling.

---

11. Conclusion

This submission presents a structured, reasoning-driven approach to narrative consistency checking. By decomposing the task into explicit claims, controlled evidence retrieval, and rule-based contradiction analysis, the system avoids many pitfalls of end-to-end generation.

The design aligns closely with the objectives of Track A: correctness, robustness, long-context handling, and thoughtful NLP system engineering. While simple by design, the approach demonstrates that careful reasoning pipelines remain highly effective for complex narrative tasks.