Kharagpur Data Science Hackathon(KDSH)2026

Story Consistency Verification Using Retrieval-Augmented Reasoning

Team Name: INNOVATRIX

Track: A

Task: Narrative Consistency Verification

Team Leader: P.Sai Varshini

Team Members: KS.Varshini,B.Srijani

1. INTRODUCTION:

Ensuring factual consistency between a summary or backstory and a long narrative document is a challenging problem for modern language models. Long contexts, ambiguous language, and implicit storytelling often lead to incorrect or hallucinated reasoning. Track A of KDSH 2026 focuses on identifying whether a given backstory is consistent with its corresponding novel.

In this work, we propose a retrieval-augmented reasoning framework that decomposes the verification problem into interpretable steps. Our approach emphasizes scalability, robustness to noisy text, and conservative decision-making to minimize false positives.

2.Overall Approach:

The proposed system follows a multi-stage pipeline

1 .Data Preprocessing:

The provided datasets contain long textual fields and are stored in tab-separated format. We load the data using a robust parsing strategy that handles malformed rows and preserves textual integrity.

Each test instance consists of:

- A unique story identifier
- A backstory describing key narrative facts
- A corresponding novel text

2 .Claim Extraction:

Rather than reasoning over the entire backstory at once, we decompose it into atomic factual claims. These claims typically capture:

- Character attributes
- Events
- Relationships
- Temporal facts

This decomposition simplifies downstream reasoning and allows independent verification of each claim.

3 .Evidence Retrieval:

Because novels are long and exceed the context window of most language models, we employ a retrieval-based strategy:

- Novel texts are split into overlapping chunks of fixed length
- For each claim, the most relevant chunks are retrieved using semantic similarity
- Only retrieved chunks are used for verification

This approach ensures that the reasoning model operates on focused and relevant evidence.

4 .Claim Verification and Aggregation

Each claim is evaluated against its retrieved evidence and classified as:

- SUPPORT – evidence aligns with the claim
- CONTRADICT – evidence clearly contradicts the claim
- UNKNOWN – insufficient or ambiguous evidence

A story is labeled inconsistent only if multiple independent contradictions are detected. Otherwise, it is labeled consistent. This aggregation strategy reduces sensitivity to isolated errors and narrative ambiguity.

3.Handling Long Context

Long narrative documents pose significant challenges due to memory constraints and noise. Our approach addresses this through structured context management:

1 .Text Chunking

Novels are segmented into chunks of approximately 800–1200 tokens. This size preserves semantic coherence while remaining computationally efficient.

2 .Top-K Retrieval

For each claim, only the top-k most relevant chunks are selected. This prevents irrelevant information from influencing the reasoning process.

3 .Advantages

- Prevents context overflow
- Improves factual grounding
- Reduces hallucination
- Enables scalable processing of long documents

4. Distinguishing Causal Signals from Noise

Narrative texts often contain ambiguity, metaphor, and implicit references. A naive contradiction detector would produce many false positives. We address this challenge through the following design choices:

1 .Multi-Contradiction Threshold

A single contradictory claim is insufficient to mark a story as inconsistent. We require multiple independent contradictions to increase confidence.

2 .Evidence-Based Decisions

Contradictions must be explicitly supported by retrieved text. Claims without clear evidence are conservatively classified as unknown.

3 .Conservative Bias

When evidence is weak or ambiguous, the system defaults to labeling the story as consistent. This reduces erroneous contradiction detection.

5. Results Generation

The system produces a results.csv file containing:

- story_id
- prediction (1 = consistent, 0 = inconsistent)
- rationale (short explanation based on detected contradictions)

The entire pipeline is fully automated and reproducible, requiring only a single command to generate predictions end-to-end.

6. Limitations and Failure Cases

Despite its effectiveness, the approach has several limitations:

1 .Implicit Knowledge

Some contradictions rely on unstated narrative assumptions or world knowledge that is not explicitly present in the text.

2 .Character Aliasing

Characters may be referred to by multiple names or pronouns, which can reduce retrieval accuracy.

3 .Literary Ambiguity

Metaphorical language, unreliable narration, and stylistic variation may lead to uncertain classifications.

4. Retrieval Errors

If relevant evidence is not retrieved among the top-k chunks, the system may incorrectly label a claim as unknown or consistent.

7. conclusion

We presented a retrieval-augmented reasoning framework for story consistency verification that effectively handles long contexts and narrative noise. By decomposing backstories into claims, retrieving focused evidence, and applying conservative aggregation logic, our approach balances accuracy and robustness.

The system is interpretable, scalable, and fully reproducible, making it suitable for real-world narrative reasoning tasks and competitive evaluation settings such as KDSH 2026.