



# **KDS HACKATHON 2026**

## **REPORT**

# Technical Report: Backstory Consistency Verification System

## 1. Overall Approach

### Core Objective

The **Backstory Consistency Verification System** is a retrieval-based Natural Language Processing (NLP) framework designed to automatically assess whether a given backstory claim is **semantically consistent or contradictory** with respect to reference novel texts. The system works by retrieving relevant passages from primary literary sources and comparing them against the input claim using semantic similarity and reasoning techniques.

It produces a **binary classification output**—**Consistency (1)** or **Contradiction (0)**—along with **verbatim textual evidence** extracted from the source material to justify the decision.

This system follows **Track A: System Reasoning with NLP methods and Generative AI (GenAI)** to perform semantic understanding, contextual reasoning, and evidence-based verification.

### Problem Statement and Methodology

The system addresses the following problem: **Given a backstory claim and a primary novel text, determine whether the claim is factually consistent with the content of the novel.** To solve this, the system adopts a **multi-stage retrieval-and-reasoning pipeline** designed to handle long-form literary texts while ensuring accurate semantic comparison and explainability.

The approach consists of **four core components**:

#### 1. Text Chunking:

The complete novel text is segmented into **overlapping, fixed-length word-based chunks**. This strategy mitigates document-length limitations imposed by embedding models while preserving **contextual continuity across chunk boundaries**, ensuring that important narrative details are not lost during segmentation.

#### 2. Semantic Embedding:

Both the generated text chunks and the input backstory claim are transformed into **high-dimensional semantic vector representations** using a pre-trained **Sentence-Transformer model (all-MiniLM-L6-v2)**. These embeddings capture

contextual meaning beyond surface-level lexical similarity, enabling robust semantic comparison in embedding space.

### 3. Evidence Retrieval:

Using **cosine similarity scoring**, the system computes similarity scores between the claim embedding and each chunk embedding. It then retrieves the **top-k most semantically relevant chunks** from the novel, prioritizing passages that exhibit the highest semantic alignment with the backstory claim. These retrieved excerpts serve as potential supporting or refuting evidence.

### 4. Consistency Decision:

A **threshold-based verification mechanism** evaluates the maximum similarity score obtained from the retrieved chunks against a **tuned similarity threshold**. If the score exceeds the threshold, the claim is classified as **consistent (1)**; otherwise, it is labeled as **contradictory (0)**. The retrieved top-k chunks are presented alongside the decision, providing **transparent and explainable evidence** for the system's verdict.

## Design Rationale

The threshold-based decision rule provides interpretability and controllable precision-recall trade-offs through empirical optimization. Evidence retrieval ensures that predictions are supported by verbatim text, enabling human validation and system transparency.

## 2. Handling Long Context

### Strategy for Managing Extended Text

Literary novels are often far longer than what neural models can process in a single pass due to token and memory limitations. To handle this challenge, the system is designed to efficiently work with extended texts without losing important narrative details.

### Chunking Strategy

To make long novels manageable, the text is divided into **fixed-size, overlapping word-based chunks**. The overlap between consecutive chunks ensures that important contextual information—especially details that span across paragraph or sentence boundaries—is preserved.

- **Benefit:** Enables processing of arbitrarily long texts while maintaining local semantic context.

## Retrieval-Based Prioritization

Rather than processing the entire novel end-to-end, the system employs a retrieval step that dynamically selects only the top-k most relevant chunks relative to each backstory claim.

- **Top-k Retrieval:** By sorting chunks by cosine similarity and selecting only the highest-ranked excerpts, the system concentrates analysis on informationally relevant portions.
- **Efficiency:** Reduces computational overhead and memory consumption while focusing decision-making on salient evidence.

## Two-Stage Pipeline

1. **Coarse Filtering:** Retrieve top-k chunks using efficient cosine similarity computation across all chunks.
2. **Fine-Grained Decision:** Apply threshold-based logic on top-k results to determine final consistency verdict.

Overall, this approach enables the system to handle long novels **without truncation or loss of critical information**, while maintaining both **accuracy and computational efficiency**.

## 3. Distinguishing Causal Signals from Noise

### Signal Identification Methodology

Meaningful patterns—genuine consistency or contradiction signals—are distinguished from noise through semantic similarity scoring and evidence-based validation.

#### Semantic Embedding as Signal Amplification

- **Semantic Space Alignment:** Sentence-Transformer models encode both the backstory claim and novel passages into the same semantic space. In this space, texts with similar meanings are positioned closer together—even if they use different words.
- **Noise Reduction:** Superficial term matching is inherently avoided; only semantically meaningful alignments produce high similarity scores. Synonymous expressions and paraphrases naturally register as signals rather than noise.

#### Threshold-Based Filtering

- **Similarity Thresholding:** The system applies a learned threshold on maximum cosine similarity scores retrieved from the top-k chunks. Similarities below this threshold are classified as insufficient evidence (contradiction), while those above it indicate consistency.

- **Empirical Optimization:** The threshold is tuned on labeled training data (train.csv) by sweeping across candidate values and selecting the threshold that maximizes accuracy on known labels.

## Top-k Evidence Validation

- **Multi-Evidence Aggregation:** By retrieving top-k chunks (rather than a single best match), the system reduces susceptibility to outlier noise. Consensus among multiple high-scoring matches strengthens confidence in the decision.
- **Explainability:** Returned verbatim excerpts provide human-verifiable evidence, enabling manual inspection and filtering of false positives caused by semantic artifacts.

## Heuristic Filters

The system inherently filters irrelevant data through:

- **Cosine Similarity Metric:** Normalized vectors and cosine distance naturally downweight less relevant content.
- **Book-Specific Matching:** Claims are matched against the correct novel source (via book\_name column), preventing cross-novel contamination.

# 4. Key Limitations and Failure Cases

## Known Constraints and Assumptions

### Semantic Embedding Limitations

- **Out-of-Domain Concepts:** If a backstory claim references entities, events, or concepts not present or only obliquely mentioned in the novel, the embedding model may fail to identify genuine signals due to vocabulary or conceptual gaps.
- **Implicit vs. Explicit Information:** The system cannot reliably infer implicit plot details or unstated context. Claims about events not directly narrated in the novel may be classified as contradictions despite being logically consistent with the narrative.

### Threshold-Based Decision Rule

- **Single Threshold Rigidity:** A fixed threshold applied uniformly to all claims may be suboptimal for claims with varying semantic similarity distributions. Claims requiring nuanced judgment (e.g., temporal ambiguities, partial truths) may be misclassified.
- **Threshold Overfitting:** Tuning thresholds on training data risks overfitting to artifacts in the labeled dataset, reducing generalization to novel claims.

## Chunking-Induced Information Loss

- **Boundary Fragmentation:** Critical information spanning chunk boundaries may be split across separate chunks, reducing within-chunk semantic coherence and potentially lowering retrieval effectiveness.
- **Fixed Chunk Size:** A predetermined chunk size does not adapt to varying information density across the novel, potentially creating inappropriately sized segments.

## Model and Data Dependencies

- **Embedding Model Bias:** Sentence-Transformers (`all-MiniLM-L6-v2`) has inherent biases toward certain domains and writing styles. Performance may degrade on literary texts outside the model's training distribution.
- **Limited Training Data:** Accuracy depends on the size and quality of labeled training data (`train.csv`). Sparse or noisy labels directly undermine threshold tuning quality.
- **Cross-Novel Generalization:** The system requires labeled examples for each novel to reliably tune novel-specific thresholds. Applying thresholds trained on one novel to an untrained novel may yield poor performance.

## Failure Scenarios

### Scenario 1: Semantic Drift

Claims that use different vocabulary or phrasing than the novel, even when semantically consistent, may produce low similarity scores and be misclassified as contradictions.

### Scenario 2: Negation and Logical Operators

Claims containing explicit negations (e.g., "X never happens") or complex logical structures may not be reliably distinguished from affirmations, leading to incorrect consistency decisions.

### Scenario 3: Temporal and Quantitative Precision

Claims specifying exact dates, counts, or measurements must be matched precisely; small discrepancies (e.g., claiming an event occurred on Day 5 when the novel says Day 6) are treated as contradictions, even if functionally minor.

### Scenario 4: Scalability Constraints

- **Computational Overhead:** Embedding and similarity computations scale linearly with the number of chunks and number of claims. Very large novels or batch processing may incur significant latency.
- **Memory Requirements:** Storing embeddings for all chunks of a large novel requires substantial memory; high-dimensional embeddings (384-dim for `all-MiniLM-L6-v2`) multiply this cost.

### Scenario 5: Limited Explanation for Complex Decisions

While the system returns top-k evidence, it does not provide reasoning for why a specific threshold was applied or why the decision was borderline. This limits human understanding of edge cases.

## Scalability and Accuracy Constraints

- **Novel Size:** Extremely long novels (>500K words) may face computational constraints or reduced performance due to chunking artifacts.
- **Claim Ambiguity:** Inherently ambiguous claims without clear factual grounding cannot be reliably classified; the system cannot resolve human-level interpretive uncertainty.
- **Domain Shift:** Application to genres or writing styles significantly different from the training corpus may degrade accuracy substantially.

## Conclusion

The Backstory Consistency Verification System provides a scalable, interpretable approach to automated claim verification using semantic embeddings and evidence retrieval. While the system effectively handles long contexts and filters semantic signals, it is constrained by embedding model limitations, threshold rigidity, and dependence on labeled training data. Critical consideration of these limitations is essential when deploying the system on novel datasets or domains.