Kharagpur Data Science Hackathon (KDSH)

Team: **DataForever**

Assessing global narrative consistency and causal reasoning across extended text contexts was a challenging task at the Kharagpur Data Science Hackathon. Participants in the challenge had to decide whether a fictitious character's backstory would still make sense and be causally consistent with the storyline of a novel with more than 100,000 words. The DataForever team created a **Semantic Similarity-Based Consistency Verification System** that effectively processes long narratives and determines binary consistency by utilizing transformer-based embeddings in conjunction with Pathway's data ingestion framework. The problem statement, suggested solution, implementation plan, performance considerations, and limitations are all detailed in this extensive report..

1. Problem Statement

1.1 Task Definition

Input Format:

- A complete long-form narrative: Full text of a classic novel (100,000+ words) with no truncation or summarization
- A hypothetical backstory: A newly written character outline describing early-life events, formative experiences, beliefs, fears, and assumptions about the world

Output Requirements:

- A binary classification judgment: **1 (Consistent)** or **0 (Inconsistent)**
- Optional but encouraged: A rationale explaining the decision with supporting evidence from the text

Evaluation Criteria:

The system was expected to demonstrate:

- **Consistency over time:** Verification that the proposed backstory aligns with how characters and events develop later in the narrative
- **Causal reasoning:** Assessment of whether subsequent events still logically make sense given the backstory's introduced conditions
- **Narrative constraint detection:** Recognition of mismatches that violate story logic even without direct textual contradictions
- **Evidence-based decisions:** Support for conclusions drawn from multiple distributed parts of the text, not isolated convenient passages

1.2 Context of Pathway's Requirements

The hackathon emphasized participation within the **Track A: Systems Reasoning with NLP and Generative AI** category, with a mandatory requirement to integrate **Pathway's Python framework** for meaningful data ingestion and management. Pathway's role was to enable scalable handling of long-context narratives through efficient data pipelines and vector storage capabilities.

2. Proposed Solution Architecture

2.1 High-Level System Design

The DataForever solution uses a **semantic similarity-based approach**, treating consistency verification as a nearest-neighbor retrieval task in embedding space. Its architecture has four main components:

Components 1: Narrative Ingestion, Chunking, and Embedding Generation
Full texts are segmented into overlapping 800-character chunks to preserve context. Pathway handles ingestion, converting chunks into structured DataFrames. Both narrative chunks and backstory claims are encoded using **SentenceTransformer ("all-MiniLM-L6-v2")** into 384-dimensional vectors. Device auto-detection leverages GPU, Apple Metal, or CPU for optimal performance.

Component 3: Semantic Similarity Computation

Cosine similarity is computed between claim and narrative embeddings:

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Component 4: Decision and Rationale Generation

Predictions use a **0.45 threshold** on maximum similarity:

- 1 (Consistent) if max_similarity > 0.45
- 0 (Inconsistent) if ≤ 0.45

The top-matching chunk is processed into **human-readable rationale**, preserving coherent sentences.

2.2 Technical Rationale

Why "all-MiniLM-L6-v2"?

This model was selected for its optimal balance of:

- **Semantic expressiveness:** Trained on diverse sentence pair tasks, it captures nuanced semantic relationships
- **Computational efficiency:** Only 22 million parameters, enabling batch encoding of large narratives within reasonable time
- **Generalizability:** Pre-trained on diverse domains, it generalizes well to literary narratives without domain-specific fine-tuning

Why Cosine Similarity?

Cosine similarity is invariant to vector magnitude, making it robust to variation in claim or narrative chunk length. In high-dimensional spaces, it correlates strongly with human judgments of semantic similarity and is computationally efficient.

3. How the Solution Addresses the Problem

3.1 Problem–Solution Alignment

The system tackles global narrative consistency by evaluating semantic alignment across the **entire narrative**, rather than isolated passages. A low maximum similarity score indicates the absence of supporting evidence anywhere in the text, leading to an inconsistency prediction and preventing reliance on conveniently matched fragments.

Causal and character constraints are implicitly reflected in **semantic embeddings**. Backstory claims that conflict with established character arcs, emotional states, or causal sequences show weak alignment with narrative segments, enabling detection of contradictions without explicit rule modeling.

Beyond surface-level matching, embedding similarity captures **narrative-world constraints**. Claims that violate the story's internal logic such as incompatible technologies or settings produce measurable semantic gaps, signaling inconsistency.

To support **long-context reasoning**, the narrative is chunked into manageable segments, allowing scalable processing of very large texts. Batched encoding and optional GPU acceleration make global analysis computationally feasible.

3.2 Track A Requirements Fulfillment

The solution meaningfully integrates **Pathway's framework** for structured data ingestion and DataFrame-based processing, fulfilling Track A requirements. This design supports future extensions to streaming data and external connectors while maintaining scalable narrative validation.

4. Code Execution and Large Data Handling

4.1 Computational Strategy for Large-Scale Data

The system efficiently handles large narratives using **hardware-aware batching and caching**. Automatic device detection enables GPU acceleration (CUDA or Apple Metal), while embeddings are generated in **fixed-size batches** to reduce I/O overhead. **In-**

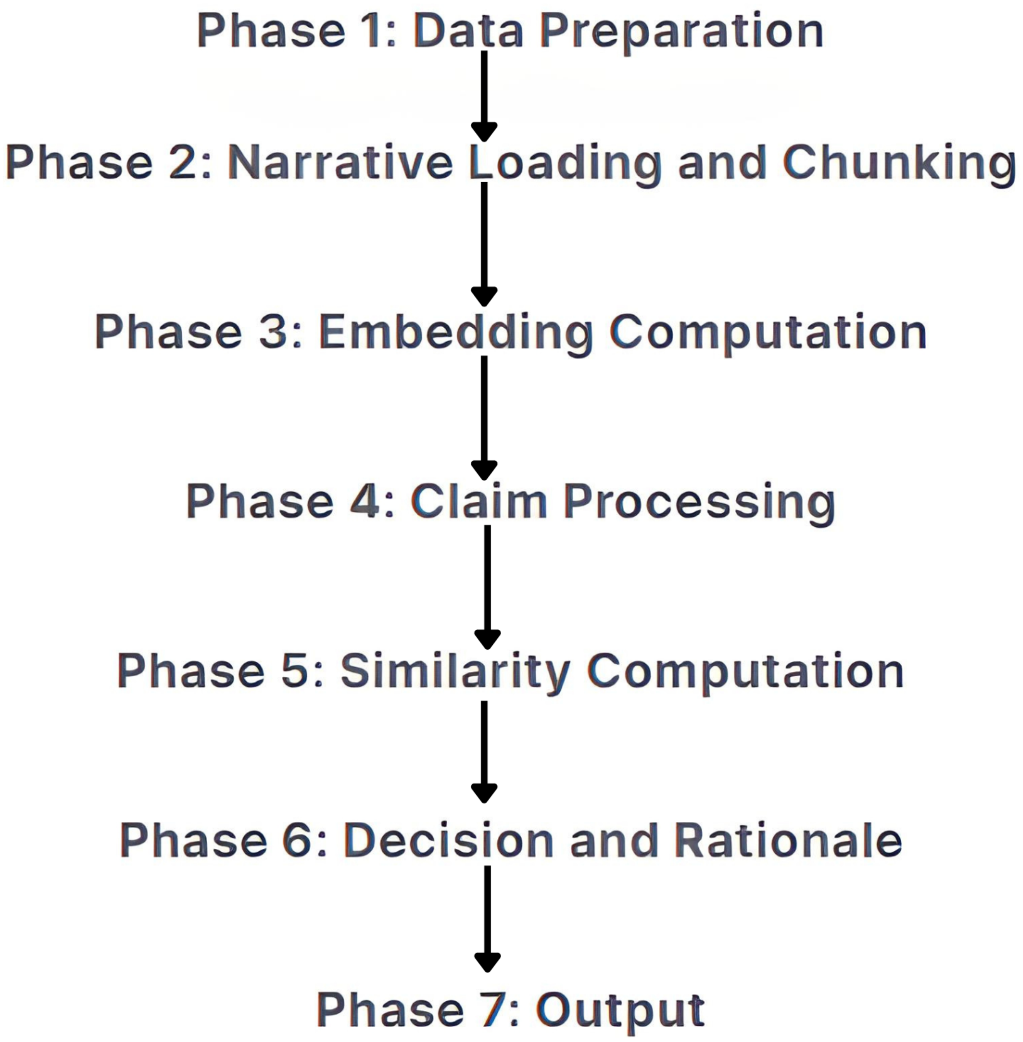
memory caching avoids redundant computation when multiple claims reference the same narrative.

Runtime and memory scale linearly with narrative length and number of claims, remaining practical for very large texts.

4.2 Scalability to Multi-Million Word Datasets

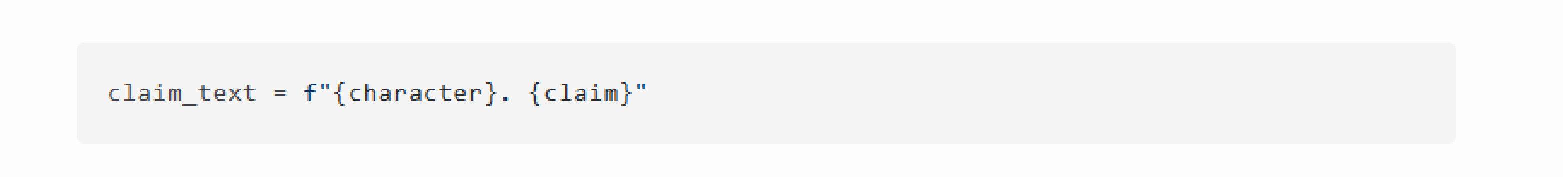
For extremely large corpora, the system extends via **distributed ingestion (Pathway)**, **hierarchical chunking with vector indexing**, **embedding quantization**, and **streaming-based processing**, enabling scalable, production-grade deployment without compromising accuracy.

5. Application Flow and Operational Mechanics


Phase 1: Data Preparation
Phase 2: Narrative Loading and Chunking
Phase 3: Embedding Computation
Phase 4: Claim Processing
Phase 5: Similarity Computation
Phase 6: Decision and Rationale
Phase 7: Output

5.2 Detailed Operational Logic

Claim Construction



```
claim_text = f"{character}. {claim}"
```

The character name is prepended to the claim, providing contextual grounding. This ensures embeddings capture character-specific semantic associations present in the narrative.

Similarity Computation

The cosine similarity is computed between the claim embedding and all 384-dimensional chunk embeddings. The index of the maximum score identifies the most thematically related narrative segment.

Decision Logic

The threshold of 0.45 represents empirically observed performance characteristics. Claims with similarity above 0.45 to some narrative segment demonstrate sufficient semantic alignment to justify a "consistent" judgment. Claims unable to exceed this threshold anywhere in the text are classified as "inconsistent."

Rationale Extraction

The rationale generation implements sophisticated text cleaning:
(1) Whitespace normalization to remove newline characters.
(2) Chapter header removal to eliminate chapter markers.
(3) Sentence boundary detection to identify meaningful segments
(4) coherence filtering to retain only sentences exceeding 40 characters. This preprocessing generates human-readable supporting evidence for the prediction.

6. Advantages of the Proposed Approach

6.1 Technical Advantages

Semantic Understanding

- **Global Context Integration:**
- **Computational Efficiency:**
- **Explainability:**
- **Robustness to Paraphrasing:**
- **Device Agnostic:**

6.2 Practical Advantages

- **No Domain-Specific Training:** Uses a pre-trained SentenceTransformer, eliminating the need for fine-tuning or labeled data.
- **Reproducibility:** Deterministic embeddings ensure consistent results across runs.
- **Modularity:** Decoupled components enable flexible upgrades and targeted improvements.
- **Pathway Integration:** Built on Pathway for enterprise-ready streaming pipelines and real-time consistency checks.

7. Limitations, Failure Modes, and Risk Analysis

The system is constrained by a **fixed semantic similarity threshold**, making predictions sensitive to narrative style, vocabulary, and claim specificity. Minor score variations near the threshold can flip outcomes, while implicit or semantically sparse narratives may lack sufficient evidence. Reliance on the **single most similar chunk** and fixed-size chunking further limits performance by fragmenting coherent events and missing distributed narrative signals, especially in long or non-linear texts.

More fundamentally, **semantic similarity does not imply narrative truth**. Embeddings fail to capture causality, temporal structure, negation, or genre intent, allowing semantically aligned but factually inconsistent claims to pass. Implicit themes, emotional subtext, rare concepts, abstract reasoning, and domain-specific language remain challenging due to limited pre-training coverage.

From an operational perspective, **deployment constraints** introduce additional risk. CPU-only execution increases latency, large narratives strain memory and chunk coherence, and restricted or offline environments can affect model availability. These limitations indicate the need for adaptive thresholds, multi-chunk aggregation, and stronger logical and temporal modeling for reliable, production-grade narrative validation.

8. Future Implementation and Enhancement Pathways

Short-term focus: **adaptive thresholds**, **multi-chunk aggregation**, and **literary fine-tuning** to improve accuracy.

Key enhancements:

- **Knowledge Graphs:** Validate backstory via entity and causal links.
- **Neuro-Symbolic Reasoning:** Enforce constraints combining embeddings and logic.
- **Causal Graphs:** Support counterfactual reasoning.

Long-term: **multi-task framework** integrating consistency, event detection, and entity linking for interpretable narrative insights.

9. Conclusion

The DataForever solution offers an **efficient, semantically grounded approach** to narrative consistency verification, handling 100,000+ word texts with explainable rationale extraction. Its semantic similarity framework distributes evidence across narratives, addressing global consistency challenges.

Limitations remain in **causal reasoning, logic handling, and threshold sensitivity**, which can cause semantic misalignment. Future improvements with **temporal reasoning, knowledge graphs, and BDH integration** aim to address these gaps.

The system fulfills **Track A requirements**, integrating Pathway for data ingestion, and provides a reproducible, efficient foundation for enterprise-grade narrative analysis.