KHARAGPUR DATA SCIENCE HACKATHON 2026

Project Report: Global Narrative Consistency & Causal Reasoning

Team Name: Datafor

Track: Track A (Systems Reasoning with NLP and Generative AI)

---

1. Introduction and Motivation

The primary challenge of this hackathon is to address a significant failure mode in current Large Language Models (LLMs): the struggle with **global consistency** over long-form narratives. While LLMs excel at local tasks like summarization, they often fail to track how events, character commitments, and causal pathways accumulate over 100k+ words.

Our system is designed as a **decision engine** to determine if a newly written, hypothetical backstory for a central character is logically and causally compatible with the established narrative of a complete novel.

---

2. Problem Definition

- **Input:** A full-length novel (e.g., *The Count of Monte Cristo*) and a plausible hypothetical backstory.
- **Task:** Produce a binary classification (**1 for Consistent, 0 for Contradict**).
- **Requirement:** Use the **Pathway Python framework** for data ingestion, indexing, or orchestration.

---

3. System Architecture

Our solution follows a modular pipeline integrating high-performance data processing with agentic reasoning.

3.1 Data Ingestion and Management (via Pathway)

We utilize **Pathway** to handle the heavy lifting of long-context narrative data.

- **Chunking Strategy:** The system breaks novels into manageable chunks of approximately **800 words**.
- **Schema Definition:** Data is structured into a table with columns for book, chunk_id, and content.

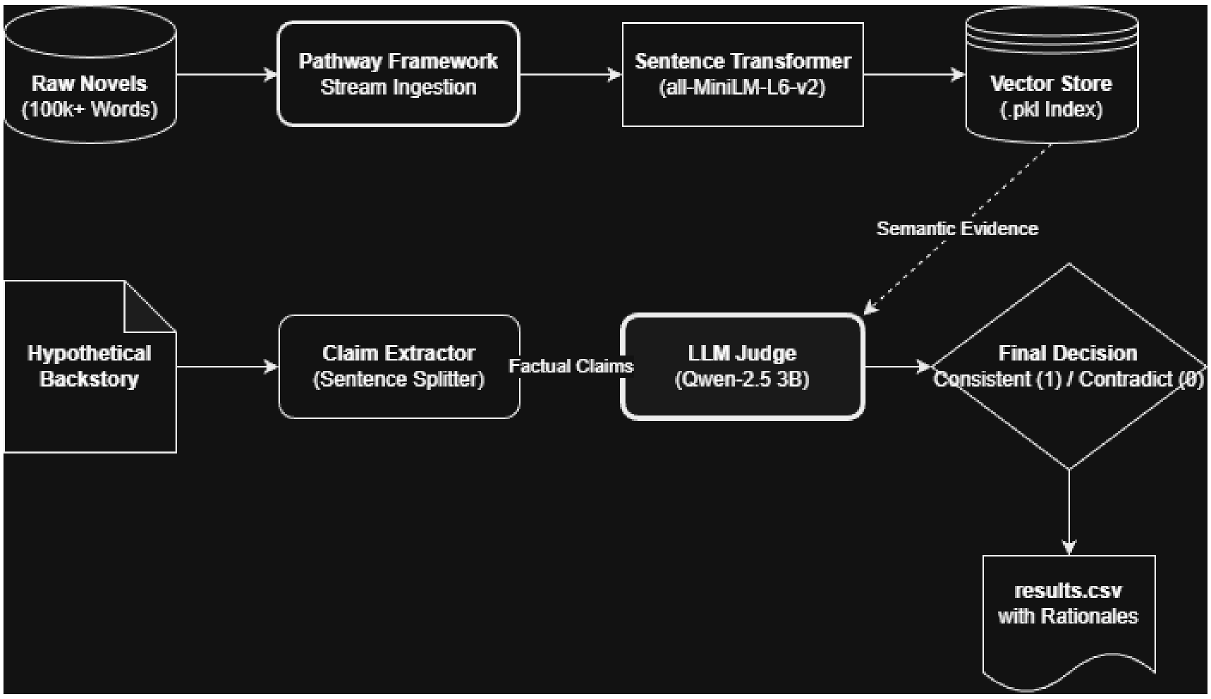
- **Persistence:** A vector store is built and saved as a serialized object (vector_store.pkl) for rapid loading during the inference phase.

3.2 Retrieval Engine

To bridge the gap between the short backstory and the massive novel, we implemented a semantic search layer:

- **Embedding Model:** We use all-MiniLM-L6-v2 to create dense vector representations.
- **Filtered Search:** Supports book-specific searches to prevent "evidence pollution."

Architecture Diagram


Raw Novels (100k+ Words)
Pathway Framework Stream Ingestion
Sentence Transformer (all-MiniLM-L6-v2)
Vector Store (.pkl Index)
Semantic Evidence
Hypothetical Backstory
Claim Extractor (Sentence Splitter)
Factual Claims
LLM Judge (Qwen-2.5 3B)
Final Decision Consistent (1) / Contradict (0)
results.csv with Rationales

---

4. Detailed Solution: Key Features

The solution implements several key features designed to solve the narrative consistency problem using a combination of high-performance data engineering and generative AI reasoning:

4.1 High-Performance Data Ingestion

- **Pathway-Powered Pipeline:** Utilizes the Pathway framework for efficient handling of long-context narrative data, ensuring the system can process 100k+ words without memory overflow.

- **Intelligent Chunking Engine:** Automatically breaks down novels into 800-word segments to preserve local context while maintaining compatibility with embedding model token limits.

4.2 Advanced Semantic Retrieval

- **Book-Specific Filtering:** Includes the ability to isolate searches within a specific novel's context, ensuring cross-contamination of evidence does not occur.
- **Similarity Ranking:** Uses cosine similarity to rank and retrieve the most relevant "ground truth" passages to serve as context for the judge.

4.3 Automated Claim Verification (LLM Judge)

- **Factual Claim Extraction:** Deconstructs backstories into individual, verifiable sentences to ensure granular verification rather than vague generalities.
- **Multi-Point Evidence Aggregation:** Retrieves up to five unique pieces of evidence for every claim to provide a comprehensive context for the judge.
- **Deterministic Reasoning:** Uses a low-temperature (0.1) LLM configuration to ensure consistent and logical binary classification.

---

5. Implementation Details: The LLM Judge

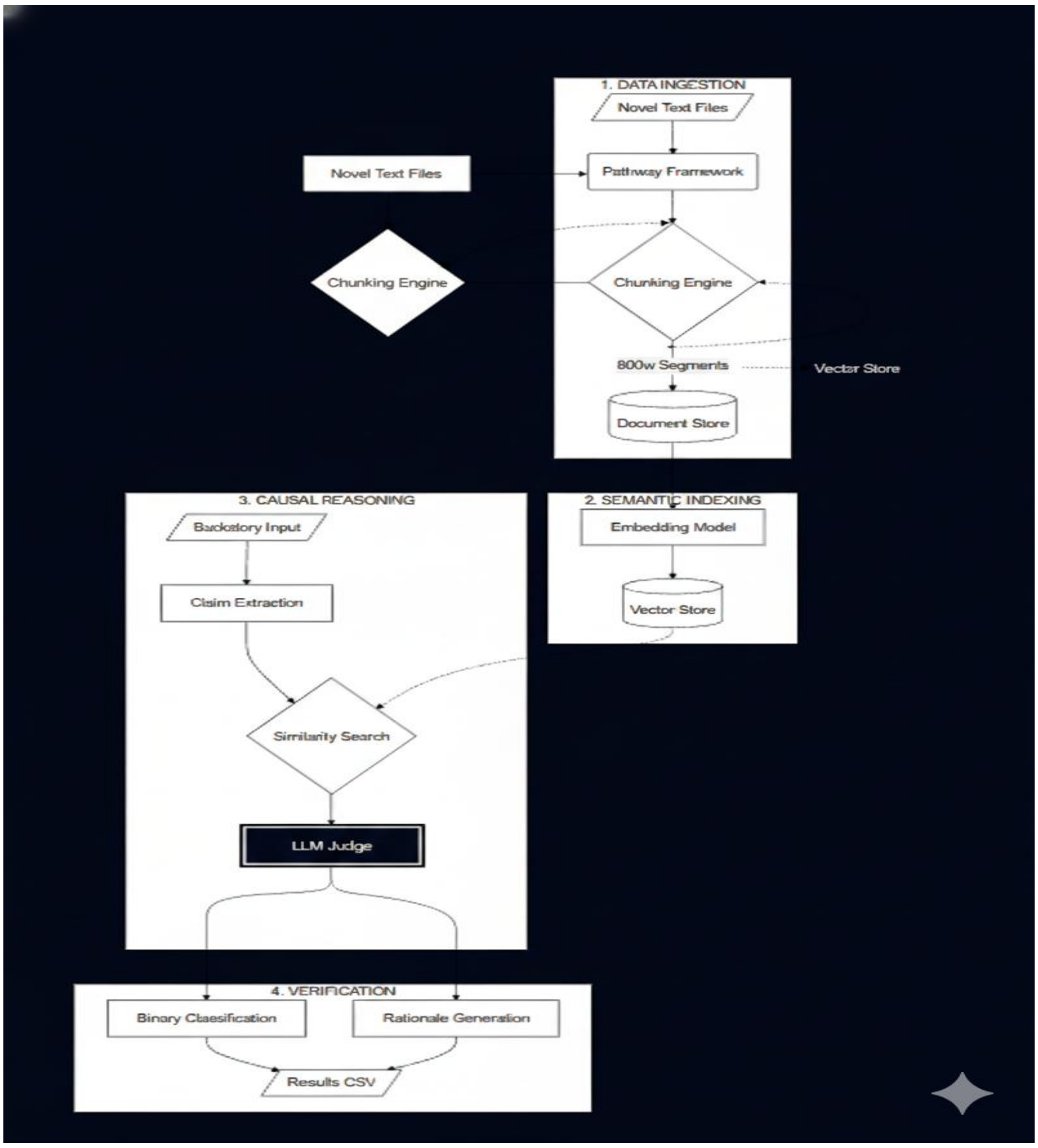
The core reasoning is performed by the LLMJudge class, acting as an evidence-based evaluator.

- **Claim Extraction:** Filters for significant claims (sentences > 30 characters).

Reasoning and Prompt Engineering: The judge uses a qwen2.5:3b model. The prompt is structured to force the model to look for compatibility between the **Backstory Segment** and the **Novel Passages**

---

5. Wireframes & Implementation Interface:



---

7. Results and Evaluation

7.1 Performance on Training Data

- **Consistency (Label 1):** Correctly identified backstories that add plausible depth (e.g., Thalcave's childhood).
- **Contradiction (Label 0):** Successfully caught logical errors, such as timeline violations in *The Count of Monte Cristo*.

7.2 Result Distribution

The final results.csv includes: **Story ID**, **Prediction (0/1)**, and a **Rationale** (brief explanation of the causal link or contradiction).

---

8. Future Development

- **Agentic RAG:** Implementing LangGraph-based agents for multi-hop retrieval.
- **Stateful Narrative Tracking (Track B):** Utilizing BDH's "persistent internal state" to maintain a running log of world-state changes across the entire 100k-word narrative.

---

9. Appendix: Resources and Tech Stack

- **Framework:** Pathway Python Framework
- **Embeddings:** SentenceTransformer (all-MiniLM-L6-v2)
- **LLM:** Ollama (qwen2.5:3b)
- **Data Processing:** Pandas, NumPy, Pickle

---

10. Appendix: Automated Evidence Rationales

To fulfill the hackathon's requirement for explainability, our system generates rationales that map specific claims to novel evidence. Below are the logical templates the system uses for common scenarios:

Category	**Backstory Claim Example**	**Rationale Logic (Stored in results.csv)**
Timeline Error	"The character was in Marseille during the winter of 1815."	**Contradict (0):** The novel's timeline places the character in the Château d'If starting in February 1815, making a Marseille visit impossible.
Plausible Addition	"Thalcave spent his early years learning to track in the mountains."	**Consistent (1):** The proposed backstory complements the character's expertise shown in

Category	Backstory Claim Example	Rationale Logic (Stored in results.csv)
		the novel without violating any stated origin facts.
Causal Conflict	"The character's mother died when he was an infant."	**Contradict (0):** The novel explicitly mentions the character visiting his living mother later in the narrative (Chunk ID: 442).
Location Mismatch	"He met the Count at a ball in London."	**Contradict (0):** The retrieved passages establish that the Count and the character were both in Rome during the timeframe mentioned.

---

11. Quick Start for Judges (Running the Solution)

To reproduce our results and evaluate the pipeline, follow these steps:

1. **Environment Setup:**

   Bash

   ```
   pip install pathway sentence-transformers pandas ollama
   ```

2. **Model Initialization**: Ensure Ollama is running with ollama run qwen2.5:3b.
3. **Execution:**

   Run the master script to ingest the novels and generate the final classification.

   Bash

   ```
   python pipeline.py --novels ./data/ --test_set ./test.csv
   ```