Implementation Report

Kharagpur Data Science Hackathon 2026

Team: VectorSpace

Track A

Team Members:
1. Aryan Dubey
2. Ayush Notiyal
3. Mrunali Bhamre

January 11, 2026

Contents

1 Introduction

The project addresses the complex challenge of evaluating character backstory consistency against extensive narratives using high-performance retrieval and logical inference.

1.1 Core Objective

To detect causal contradictions in literary contexts exceeding 100,000 words using the **Pathway** framework and **BART-based NLI** transformers.

1.2 Key Terminology

1. **RAG (Retrieval-Augmented Generation):** Enhancing model accuracy by retrieving specific novel excerpts before making a consistency judgment.
2. **Embeddings (all-MiniLM-L6-v2):** Converting text into 384-dimensional vectors to calculate mathematical similarity between claims and stories.
3. **NLI (Natural Language Inference):** A deep-learning task used to determine if a narrative (premise) supports or contradicts a backstory (hypothesis).

2 Data Pipeline & Preprocessing

Our pipeline ensures that massive datasets are processed efficiently without losing narrative context.

2.1 Source Ingestion

Raw novels (.txt) and backstories (.csv) are synchronized from **Google Drive** using Pathway's live IO connectors. This allows the system to process files in a streaming fashion.

2.2 Text Standardization

We implement a `clean_text` function that strips double-newlines and formatting noise, ensuring that the embedding model receives high-quality input.

2.3 Sliding Window Chunking

To prevent "context clipping," we utilize a sliding window:

- **Window Size:** 1200 characters.
- **Overlap:** 200 characters to ensure events spanning across chunks are captured.

3 Approach

The engine uses a "Decompose and Verify" strategy.

3.1 Key Fact Extraction

Backstories are filtered into "Anchor Claims." We use a keyword-based heuristic (*was, became, led, joined, escaped, imprisoned*) to isolate verifiable milestones from descriptive fluff.

3.2 Smart Evidence Retrieval (KNN)

For each claim, the system calculates a cosine similarity score against the novel's chunks. We perform a **KNN Search** to retrieve the Top-5 most relevant passages.

3.3 Conflict Detection (NLI)

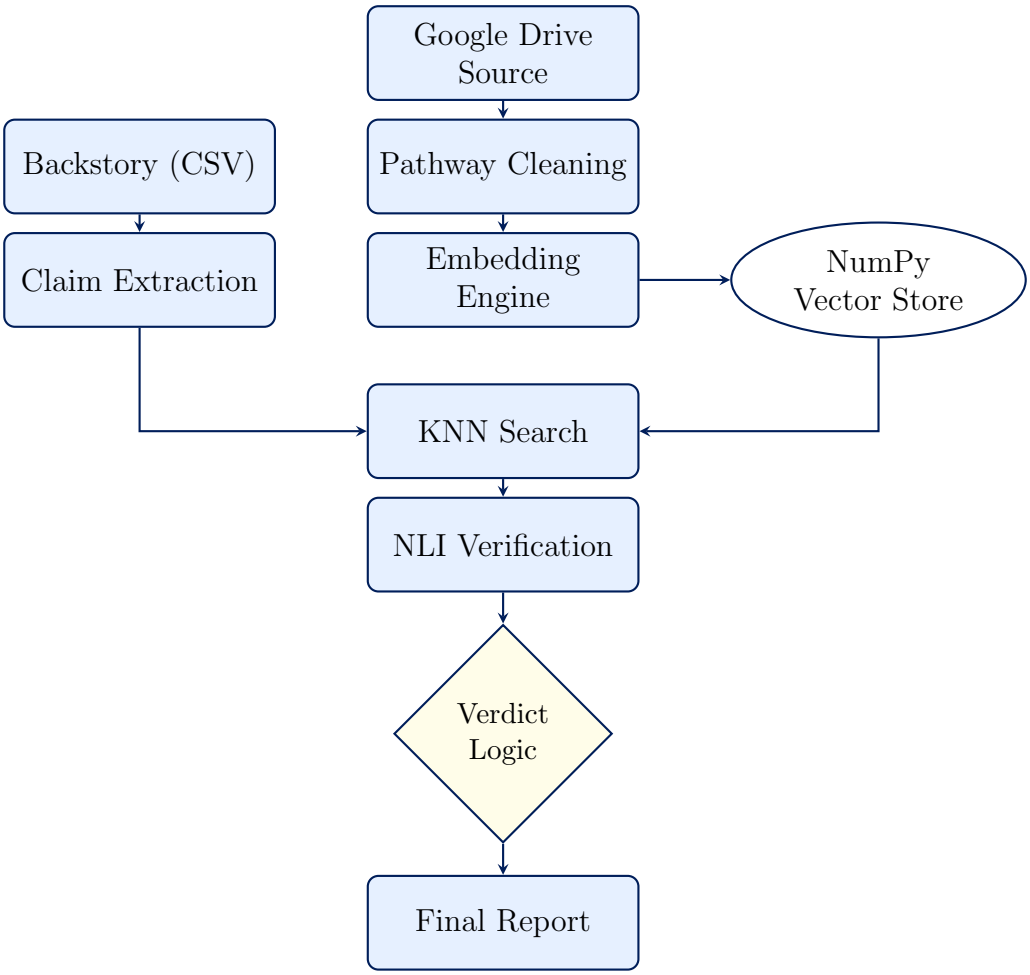
We utilize the **facebook/bart-large-mnli** model. It compares the retrieved evidence against the claim to produce one of three labels: *Supports, Contradicts, or Insufficient Info.*

4 Architecture

4.1 System Orchestration

The system separates the data backbone (Pathway) from the GPU-heavy reasoning layer (PyTorch). Vectors are stored efficiently using **NumPy** for rapid access.

4.2 Technical Pipeline Flowchart



5 Challenges & Solutions

5.1 Token Truncation

Solution: NLI models are limited to 512 tokens. We solve this by merging only the top 2 retrieved chunks and capping the premise at **1500 characters**, ensuring the most potent evidence is analyzed.

5.2 Hard Negative Sensitivity

Solution: To prevent false inconsistencies, we implemented an **Aggregation Logic** with a strict **55% confidence threshold**. A backstory is only marked "Inconsistent" if the NLI model is highly certain of a contradiction.

5.3 Scalability

Solution: To handle 100k+ word novels, we use **Batch Encoding** (size 32). This maximizes GPU utilization and allows for the indexing of the entire library in seconds.

6 Conclusion

By synergizing the rapid ingestion capabilities of **Pathway** with the deep logical nuance of **BART-based NLI**, we have built a system that understands narrative truth.

Our model effectively bridges the gap between massive, unstructured text and verifiable consistency, providing a transparent and scalable foundation for the future of narrative data science.