ttda704 at SemEval-2026 Task 4: Modeling Narrative Structures via Pseudonymization and Multi-View Sentence Alignment

Tai Tran Tan[1,2]*, **An Dinh Thien**[1,2]*
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{22521287, 22520010}@gm.uit.edu.vn

Abstract

We present our approach to *SemEval 2026 Task 4: Narrative Story Similarity and Narrative Representation Learning*. Our solution uses contrastive learning with fine-tuned sentence transformers to capture narrative similarity across abstract themes, course of action, and outcomes. We develop two pipelines: (Track A) a single-view method that encodes full narratives with smart layer freezing to reduce overfitting, and (Track B) a multi-view method that models theme, plot, and outcome with view-specific projection heads and self-supervised alignment. Both pipelines build on sentence-transformers models and are trained with contrastive loss on synthetic data. The code is available at the following GitHub repository: https://github.com/dinhthienan33/SemEval2026-Task4-ttda704.

1 Introduction

Understanding and comparing complex narratives is a fundamental yet persistent challenge in Natural Language Processing. The significance of this problem is highlighted by consecutive SemEval competitions dedicated to narrative analysis ranging from measuring the multidimensional similarity of multilingual news stories (Chen et al., 2022) to characterizing and extracting framing narratives from online media (Piskorski et al., 2025). Fictional story similarity demands robust representation learning techniques that extend far beyond simple lexical overlap or factual event extraction. To effectively map the architecture of a story, computational models must isolate and encode deep latent dimensions, such as abstract themes, the progression of the plot, and final resolutions (Hatzel and Biemann, 2024a).

Historically, handling severe narrative ambiguity has proven extremely difficult. Even advanced generative models frequently falter when faced with scenarios possessing multiple valid interpretations, highlighting the necessity for rigid structural scaffolding to guide model reasoning (Wang et al., 2024). Furthermore, attempting to resolve these uncertainties at inference time using computationally heavy mechanisms such as generating multiple reasoning paths via self-consistency voting (Wang et al., 2023) introduces massive latency. This makes LLM-based approaches highly impractical for scalable story retrieval, embedding extraction, or large-scale comparative tasks.

To overcome these limitations, our work shifts away from relying on expensive generative inference at prediction time and instead focuses on efficient narrative representation learning. Inspired by cognitive dual-process theories, which suggest that human cognition adapts its processing depth based on the complexity of the information (Walter, 2011), we use LLMs only in a limited offline role for synthetic data generation and narrative structure extraction, while keeping the final similarity system itself embedding-based and efficient at inference time.

The main contributions of our work are:

- LLM-augmented Narrative Representation. A deterministic framework that leverages LLMs for high-quality data augmentation and structural extraction, while utilizing fine-tuned sentence transformers for final similarity inference.
- Multi-view structural modeling. A contrastive learning approach that decomposes stories into theme, course of action, and outcome via view-specific projection heads with self-supervised alignment.
- Robust contrastive training. Triplet-margin optimization with smart layer freezing to improve generalization while preserving narrative nuance.

*Equal contributions.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2800–2809

2 Related Work

2.1 Narrative Representation and Story Embeddings

While traditional Semantic Textual Similarity (STS) focuses on lexical and sentence-level overlap, narrative similarity requires modeling higher-order structures. This follows the shift in the field towards "Story Embeddings," as formalized by Hatzel et al. (Hatzel and Biemann, 2024a), who argue that determining story similarity requires moving beyond keyword matching to capture the underlying narrative graph. Previous approaches have attempted to model stories hierarchically (Lee et al., 2020), often focusing on the sequential progression of events. More specific mechanisms, such as re-contextualization through attention (Wilner et al., 2021), have been proposed to better encode the flow of a narrative.

2.2 Sentence Transformers and Backbone Architectures

To operationalize narrative similarity efficiently, we build upon the Sentence-BERT framework (Reimers and Gurevych, 2019), which enables fast and scalable dense text comparison. While foundational contrastive learning approaches (Gao et al., 2021) excel at general semantic matching, off-the-shelf models are insufficient for fictional texts. Framing narrative comparison as a dense retrieval task requires models to encode deep structural elements rather than mere surface-level semantics (Hatzel and Biemann, 2024a; Younus and Qureshi, 2025). To address this gap and align the representation space with narrative theory, we systematically fine-tune MPNet backbone on the task-provided synthetic narrative triples.

2.3 Multi-View and Data-Efficient Contrastive Learning

The separation of narrative similarity into theme, plot, and outcome naturally frames our task as a multi-view learning problem. As demonstrated by Zhang et al. (Zhang et al., 2022), leveraging multiple perspectives of a document significantly improves dense representation compared to standard single-vector compression. We implement this through a shared transformer backbone coupled with aspect-specific projection heads, effectively decoupling complex narrative signals into orthogonal embedding spaces. Furthermore, the limited availability of human-annotated narrative triples necessitates data-efficient training strategies. To overcome this scarcity and maximize model generalization, we incorporate synthetic data augmentation (Jiang et al., 2022).

3 Task Description

SemEval-2026 Task 4: Narrative Story Similarity and Narrative Representation Learning challenges participants to develop systems capable of identifying similarities between stories based on their underlying narrative structures rather than surface-level lexical overlaps (Hatzel et al., 2026).

3.1 Definition of Narrative Similarity

The task organizers define narrative similarity through **a specific schema consisting of three core aspects**:

1. **Abstract Theme:** This encompasses the defining constellation of problems, central ideas, and core motifs (e.g., a struggle against nature, a coming-of-age journey).

2. **Course of Action:** This describes the sequence of events, conflicts, and turning points. It focuses on the chronological order and the causal links between actions (e.g., a protagonist ignores a warning, suffers a loss, and then recovers).

3. **Outcomes:** This refers to the resolution of the plot, such as the fate of the characters or the moral lesson learned. The guidelines emphasize that similar themes can lead to polar opposite outcomes, which must be distinguished.

3.2 Subtasks

- **Track A (Triple Classification):** The system is presented with an *Anchor* story and two candidates, *Story A* and *Story B*. The objective is to perform a binary classification to determine which candidate is narratively closer to the Anchor based on the definitions above.

- **Track B (Representation Learning):** The system must generate vector embeddings for individual stories such that the cosine similarity between the vectors reflects their narrative affinity.

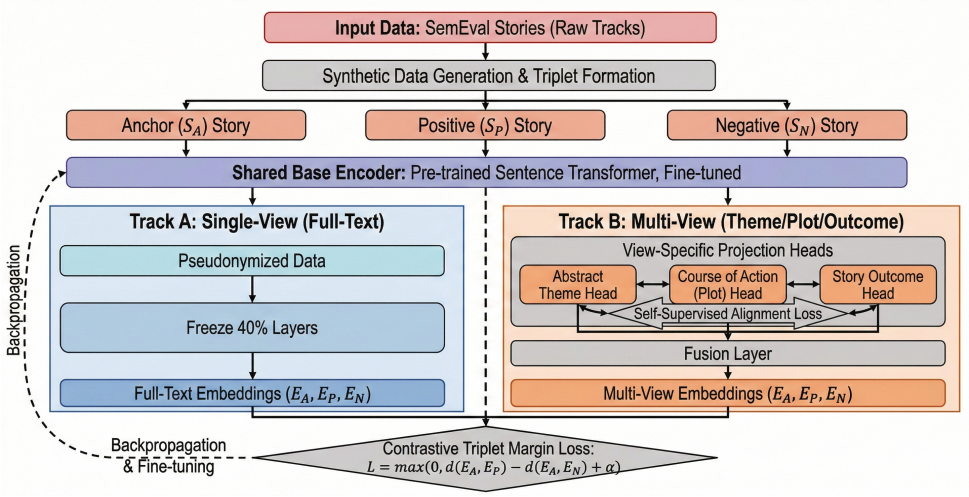

Figure 1: High-level overview of the proposed pipeline.

4 Methodology

4.1 Single-View Contrastive Learning for Track A

Track A requires determining which of two candidate stories is narratively more similar to an anchor story. We employ a single-view contrastive learning approach that learns embeddings directly from full narrative texts, optimized through a pseudonymization step and targeted fine-tuning.

Pseudonymization. To mitigate surface-level noise and decouple story progression from specific naming conventions, we apply entity-level referential normalization with consistent placeholders while preserving within-story coreference. Repeated mentions of the same character are mapped to the same pseudonym throughout the story, whereas distinct entities receive different placeholders. This makes the transformation identity-preserving at the discourse level, even though lexical content is removed. By pseudonymizing entities into generic tokens (e.g., "Character_A", "Location_1"), we reduce the model's reliance on entity overlap—a common source of spurious correlations—and encourage it to focus on narrative structure and semantic progression (Hatzel and Biemann, 2024b) (see Figure 2).

Our scheme is related to, but distinct from, simplified theta-role abstractions in linguistics. Theta roles explicitly encode semantic functions (e.g., agent, patient, experiencer), whereas our preprocessing does not induce a symbolic role inventory; it performs consistent referential normalization and lets the encoder infer event-level relations from context. In implementation, we use a two-stage pipeline with `fastcoref` and `spaCy` (`en_core_web_trf`). We first obtain coreference clusters, infer a cluster type via lightweight voting (`PERSON`→`PER`, `GPE/LOC/FAC`→`LOC`, `ORG/NORP`→`ORG`), and skip clusters without proper-noun evidence to avoid replacing non-entity spans. We then assign deterministic placeholders (e.g., `Character_A`, `Location_1`, `Organization_1`, `Entity_1`) using per-type counters and a global mapping dictionary. Next, we run NER fallback on uncovered spans, prevent overlap with a character-level usage mask, and apply deduplicated replacements from right to left (descending character offsets) to preserve index correctness. For each triplet item, one shared mapping is used across `anchor_text`, `text_a`, `text_b`, and `text` so identical mentions remain consistently pseudonymized within the sample.

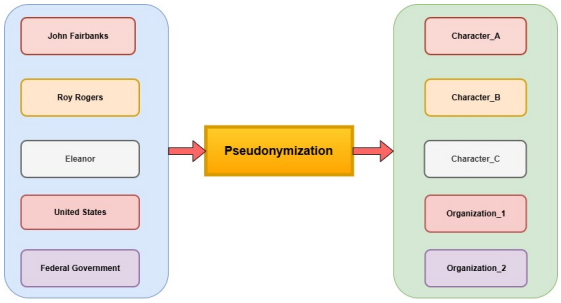

Figure 2: Example of the pseudonymization process applied to Track A narratives.

4.1.1 Architecture

Our model is built upon the `all-mpnet-base-v2` sentence transformers (Song et al., 2020), which serves as a more robust backbone compared to tra-

ditional BERT-based models.

The model produces 768-dimensional embeddings through **mean pooling** over the output token representations. We deploy this backbone for both tracks. To prevent overfitting and preserve pre-trained semantic knowledge, we implement a **Smart Layer Freezing** strategy: we freeze all embedding parameters and the bottom 40% of transformer layers to retain low-level linguistic features, while allowing the top layers to adapt to our task domains without losing generalized language understanding.

Training Strategy. We fine-tune the model using triplet margin loss on training triplets constructed from anchor-positive-negative story pairs:

$$\mathcal{L}_{\text{triplet}} = \text{ReLU}(d(\mathbf{a}, \mathbf{p}) - d(\mathbf{a}, \mathbf{n}) + m) \quad (1)$$

where $d(\mathbf{x}, \mathbf{y}) = 1 - \cos(\mathbf{x}, \mathbf{y})$ represents the cosine distance between embeddings, and $m = 0.3$ is the margin parameter.

Inference. At test time, we encode the anchor and both candidate stories. The candidate with the higher cosine similarity to the anchor is predicted as the narratively closer story.

4.2 Multi-View Contrastive Learning for Track B

For Track B, we implement a multi-view contrastive learning framework (see Figure 3) that explicitly models the three narrative dimensions identified in the task annotation guidelines.

Narrative Element Extraction. We decompose each story into three distinct views using Large Language Models: (1) *theme*, capturing the central controlling idea; (2) *plot events*, representing the chronological sequence of state-changing events; and (3) *outcome*, describing the final resolution (Hobson et al., 2024; Tian et al., 2024). This extraction provides explicit structural representations that enable the model to identify semantically similar narratives that may differ substantially in surface form (Cheng et al., 2023).

Contrastive Learning. For each view (theme, plot, outcome), we construct training triplets by matching anchor narratives with their extracted components. This creates a positive pair that shares the same narrative facet and a negative pair drawn from a different story, encouraging view-specific discrimination. The contrastive loss $\mathcal{L}_{\text{con}}$ follows a cross-entropy formulation over scaled cosine similarities (Jaiswal et al., 2020):

$$\mathcal{L}_{\text{con}} = -\log\left(\frac{\exp(\frac{\hat{\mathbf{a}}\cdot\hat{\mathbf{p}}}{\tau})}{\exp(\frac{\hat{\mathbf{a}}\cdot\hat{\mathbf{p}}}{\tau}) + \exp(\frac{\hat{\mathbf{a}}\cdot\hat{\mathbf{n}}}{\tau})}\right) \quad (2)$$

where $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$ and τ is the temperature hyperparameter. A lower τ sharpens the distribution and increases the penalty for confusing negatives, while a higher τ smooths similarities and can improve stability.

Self-Supervised View Alignment. To ensure consistency across views with limited data, we introduce an alignment loss $\mathcal{L}_{\text{align}}$. Let

$$\bar{\mathbf{z}} = \frac{1}{3}\left(\mathbf{z}_{\text{theme}} + \mathbf{z}_{\text{plot}} + \mathbf{z}_{\text{outcome}}\right).$$

We align the fused embedding $\mathbf{z}_f$ with $\bar{\mathbf{z}}$ via:

$$\mathcal{L}_{\text{align}} = \lambda \left\| \frac{\mathbf{z}_f}{\|\mathbf{z}_f\|_2} - \frac{\bar{\mathbf{z}}}{\|\bar{\mathbf{z}}\|_2} \right\|_2^2. \quad (3)$$

where λ is the scaling weight. By minimizing this distance, the model encourages a coherent joint space grounded in all three narrative dimensions.

5 Results

In this section, we present the experimental findings for both Track A and Track B. To conduct a comprehensive evaluation, we employed four distinct state-of-the-art sentence embedding models: the powerful general-purpose **all-mpnet-base-v2**; the lightweight, distilled **all-MiniLM-L6-v2** optimized for efficiency; the **LaBSE** (Language-agnostic BERT Sentence Embedding) model specialized for multilingual bitext retrieval; and **paraphrase-multilingual-mpnet-base-v2**, a multilingual variant trained on paraphrase data. For brevity, we use **para-mul-mpnet-v2** to denote the *paraphrase-multilingual-mpnet-base-v2* model in subsequent tables and analysis.

5.1 Track A Results

In Track A, our system achieved a private score of **0.6925**, securing the **14$^{\text{th}}$** position out of 42 participating teams. For reference, the top-performing system on the leaderboard reached 0.7800. As shown in Tables 5 and Tables 1, pseudonymized data yields consistent improvements. The full Dev-set comparison is provided in Appendix A.

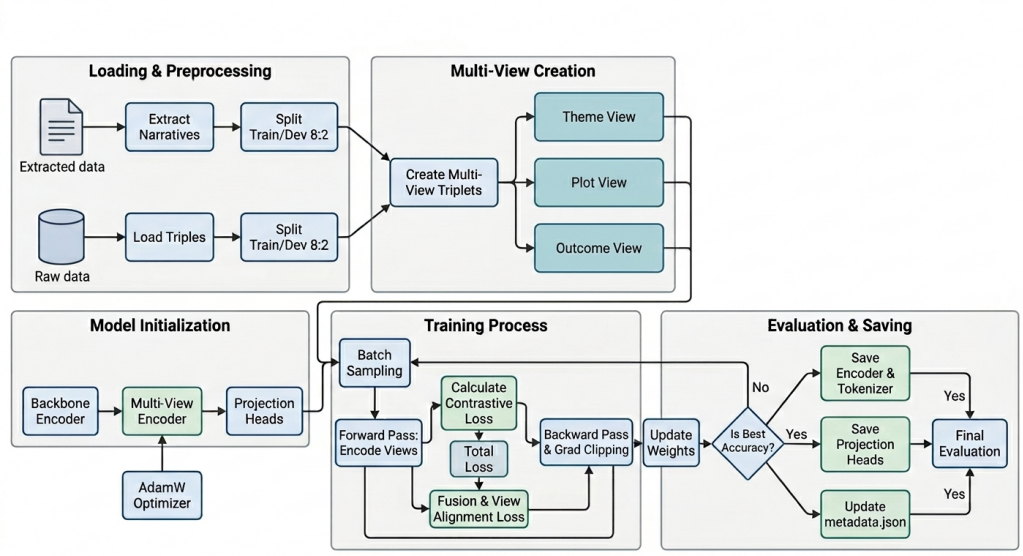

Figure 3: Multi-view contrastive learning pipeline.

Model	Raw Data	Pseudonymized Data
all-mpnet-base-v2	0.6650	**0.6925**
para-mul-mpnet-v2	0.6625	0.6300
LaBSE	0.6050	0.5975
all-MiniLM-L6-v2	0.6125	0.5800

Table 1: Performance comparison of transformer models on Track A (Test set). Bold indicates the best overall score.

Performance Analysis. The results demonstrate that **all-mpnet-base-v2** consistently outperforms other models, achieving the highest scores on both Dev and Test sets with pseudonymized data. This aligns with benchmarks on the Massive Text Embedding Benchmark (MTEB) leaderboard, where MPNet's Masked and Permuted Language Modeling (MPLM) objective allows it to capture dependency information more effectively than standard BERT-based models (Song et al., 2020; Muennighoff et al., 2023).

Impact of Pseudonymization. Pseudonymization yields consistent improvements for the top-performing MPNet model (increasing from 0.6650 to 0.6925 on the Test set). We hypothesize that pseudonymization acts as a normalization step, removing entity-specific noise and forcing the model to focus on the semantic structure of fictional narratives rather than overfitting to specific proper nouns.

5.2 Track B Results

For Track B, our team ranked **6th** out of 25 participants, achieving a private score of **0.6875** on the Codabench platform (where the top score was 0.7200). Our methodology involved testing various Large Language Models (LLMs): gpt-4o-mini, gpt-4.1, and o3-mini to extract structured schemas from the narratives. Detailed Dev-set results for Track B are reported in Appendix A.

Configuration	w_{full}	w_{theme}	w_{plot}	w_{outcome}	Dev Score
Equal weights	0.25	0.25	0.25	0.25	0.625
Full-only	1.00	0.00	0.00	0.00	0.65
Submitted (dev-tuned)	0.50	0.10	0.20	0.20	**0.725**
Higher theme	0.40	0.30	0.15	0.15	0.675

Table 2: Fusion-weight ablation on Track B (Dev set). Bold indicates the best overall score.

Manual Weight Tuning. The fusion weights are selected through manual tuning on the development set by comparing a small set of interpretable configurations in Table 2. We retain the submitted setting $(w_{\text{full}}, w_{\text{theme}}, w_{\text{plot}}, w_{\text{outcome}}) = (0.50, 0.10, 0.20, 0.20)$ because it yields the highest Dev score (0.7250), indicating that emphasizing the full-text embedding while preserving moderate plot/outcome signals provides the best balance for this setup.

Embedding Model	LLM Extractor	Score
all-mpnet-base-v2	o3-mini	**0.6875**
all-mpnet-base-v2	gpt-4.1	0.6391
all-mpnet-base-v2	gpt-4o-mini	0.6291
LaBSE	o3-mini	0.6291
LaBSE	gpt-4.1	0.6466
LaBSE	gpt-4o-mini	0.6541
para-mul-mpnet-v2	o3-mini	0.5915
para-mul-mpnet-v2	gpt-4.1	0.5940
para-mul-mpnet-v2	gpt-4o-mini	0.5614
all-MiniLM-L6-v2	o3-mini	0.5414
all-MiniLM-L6-v2	gpt-4.1	0.5664
all-MiniLM-L6-v2	gpt-4o-mini	0.5338

Table 3: Track B results across different LLM-based schema extractors and embedding models on Track B (Private test set). Bold indicates the best overall score.

6 Conclusion

We presented a contrastive learning framework for narrative similarity in SemEval 2026 Task 4, using sentence-transformer embeddings to capture theme, plot progression, and outcomes. Our Track A pipeline relies on single-view full-text embeddings with smart layer freezing, while Track B extends this with multi-view representations and alignment across narrative facets. Across both pseudonymized and raw settings, the approach delivers competitive performance while avoiding expensive generative inference, making it practical for large-scale narrative retrieval and representation learning.

Limitations

Our models depend on synthetic training data and may not fully generalize to diverse narrative styles or domains without additional adaptation. The multi-view pipeline also relies on automatically extracted narrative elements, which can introduce noise into the learned representations. Due to the shared-task timeline, we were not able to conduct a dedicated manual quality audit of synthetic extractions (e.g., human verification of theme, plot-event coherence, and outcome fidelity), so residual extraction errors may remain unquantified. Finally, fixed fusion weights and temperature settings were tuned on the development data and may not be optimal for unseen domains.

Acknowledgement

We thank the SemEval-2026 Task 4 organizers (Hatzel et al., 2026) for creating the dataset and evaluation framework.

References

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11122–11138, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hans Ole Hatzel, Ekaterina Artemova, Haimo Stiemer, Evelyn Gius, and Chris Biemann. 2026. SemEval-2026 Task 4: Narrative similarity and narrative representation learning. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*, San Diego, CA, USA. Association for Computational Linguistics.

Hans Ole Hatzel and Chris Biemann. 2024a. Story embeddings: Narrative-focused representations of fictional stories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5927–5943, Miami, Florida, USA. Association for Computational Linguistics.

Hans Ole Hatzel and Chris Biemann. 2024b. Tell me again! a large-scale dataset of multiple summaries for the same story. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15732–15741, Torino, Italia. ELRA and ICCL.

David G Hobson, Haiqi Zhou, Derek Ruths, and Andrew Piper. 2024. Story morals: Surfacing value-driven narrative schemas using large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12998–13032, Miami, Florida, USA. Association for Computational Linguistics.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2. [Online; accessed 2026-03-01].

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen

Huang, Denvy Deng, and Qi Zhang. 2022. Prompt-BERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OJ Lee, JJ Jung, and JT Kim. 2020. Learning hierarchical representations of stories by using multi-layered structures in narrative multimedia. *Sensors*, 20(7):1978.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Jakub Piskorski, Tarek Mahmoud, Nikolaos Nikolaidis, Ricardo Campos, Alipio Mario Jorge, Dimitar Dimitrov, Purificação Silvano, Roman Yangarber, Shivam Sharma, Tanmoy Chakraborty, Nuno Guimaraes, Elisa Sartori, Nicolas Stefanovitch, Zhuohan Xie, Preslav Nakov, and Giovanni Da San Martino. 2025. SemEval 2025 task 10: Multilingual characterization and extraction of narratives from online news. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2610–2643, Vienna, Austria. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Kahneman Walter. 2011. Kahneman, d. (2011): Thinking, fast and slow. *Statistical Papers*, 55.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Can LLMs reason with rules? logic scaffolding for stress-testing and improving LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7523–7543, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Sean Wilner, Daniel Woolridge, and Madeleine Glick. 2021. Narrative embedding: Re-Contextualization through attention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1405, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arjumand Younus and Muhammad Atif Qureshi. 2025. nlptuducd at SemEval-2025 task 10: Narrative classification as a retrieval task through story embeddings. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1742–1746, Vienna, Austria. Association for Computational Linguistics.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland. Association for Computational Linguistics.

A Detailed Experimental Setup

This appendix provides a comprehensive description of the experimental configurations used for both Track A (narrative similarity classification) and Track B (narrative embedding generation).

A.1 Reproducibility Checklist

- **Random seed:** 42 for model training and selection.
- **LLM extraction API/model:** OpenAI Batch API with `o3-mini` for Track B narrative element extraction.
- **Prompt location:** Full extraction prompt is provided in Appendix B (Table 9).
- **Inference settings (LLM extraction):** temperature $= 0.3$, max completion tokens $=$ 2000, and JSON-only output format.
- **Inference settings (embedding submission):** fixed fusion weights $(0.5, 0.1, 0.2, 0.2)$ for (full, theme, plot, outcome), followed by L2 normalization.
- **Hardware:** single NVIDIA GeForce RTX 3090 (24 GB VRAM), detailed in Appendix C.

Parameter	Value
Base model	`all-mpnet-base-v2`
Embedding dimension	768
Maximum sequence length	512 tokens
Training epochs	5
Batch size	16
Learning rate	2×10^{-5}
Weight decay	0.01
Warmup ratio	0.1 (linear schedule)
Triplet margin (m)	0.3
Layer freezing	40% (bottom layers + embeddings)
Mixed precision	AMP with GradScaler
Random seed	42

Table 4: Hyperparameter configuration for Track A.

A.2 Data

All experiments use the officially provided SemEval-2026 Task 4 datasets. The development set contains 200 labeled triples, each consisting of an anchor story, two candidate stories, and a binary label indicating which candidate is narratively closer to the anchor. The test set comprises 400 triples for Track A and 849 individual stories for Track B. Labels for the test set are withheld until the conclusion of the shared task.

For Track A, we construct two data variants from the development set:

- **Pseudonymized**: Stories where person names, organizations, and locations are replaced with standardized references to reduce surface-level matching bias.
- **Raw**: Original story texts without pseudonymization.

Both variants use the same 200 development triples for training and evaluation.

For Track B, narrative element extraction is performed on the Track B story corpus using the OpenAI Batch API (see Appendix B). The extracted narrative components are then used to construct multi-view training triplets by matching anchor texts from the Track A development triples with their corresponding extracted theme, plot, and outcome representations.

A.3 Track A: Single-View Contrastive Learning

Table 4 summarizes the hyperparameter configuration for Track A.

Training follows a standard triplet contrastive learning procedure. At each epoch, the full set of training triplets is shuffled and processed in mini-batches. Each triplet is tokenized and encoded in a single forward pass, producing anchor, positive, and negative embeddings. The triplet margin loss is computed using cosine distance, and gradients are scaled via automatic mixed precision. A linear warmup schedule is applied over the first 10% of total training steps, followed by linear decay. After each epoch, development accuracy is evaluated by comparing pairwise cosine similarities. The best-performing checkpoint is saved automatically.

Model	Raw Data	Pseudonymized Data
all-mpnet-base-v2	0.6550	**0.6700**
para-mul-mpnet-v2	0.6400	**0.6700**
all-MiniLM-L6-v2	0.6350	**0.6500**
LaBSE	**0.6450**	0.6350

Table 5: Performance comparison of transformer models on Track A (Dev set). Bold indicates the best overall score.

A.4 Track B: Multi-View Contrastive Learning

Table 6 summarizes the hyperparameter configuration for Track B.

Parameter	Value
Base model	`all-mpnet-base-v2`
Embedding dimension	768
Projection head hidden dim	512
Maximum sequence length	512 tokens
Training epochs	15
Batch size	32
Learning rate	2×10^{-5}
Weight decay	1×10^{-5}
Contrastive temperature (τ)	0.07
Alignment loss weight (λ)	0.5
Gradient clipping (max norm)	1.0
Samples per epoch	32
Evaluation frequency	Every epoch

Table 6: Hyperparameter configuration for Track B.

The multi-view encoder shares a single transformer backbone across three narrative views. Each view has a dedicated two-layer projection head ($768 \to 512 \to 768$). Training alternates between computing per-view contrastive losses and self-supervised view alignment losses. The total loss is the sum of all six terms (three contrastive + three alignment).

At inference, four embeddings are computed per story (full text, theme, plot, outcome) and fused using fixed weights:

$$\mathbf{e}_{\text{final}} = 0.5 \cdot \mathbf{e}_{\text{full}} + 0.1 \cdot \mathbf{e}_{\text{theme}} + 0.2 \cdot \mathbf{e}_{\text{plot}} + 0.2 \cdot \mathbf{e}_{\text{outcome}}$$

Embedding Model	LLM Extractor	Score
all-mpnet-base-v2	o3-mini	0.6750
all-mpnet-base-v2	gpt-4.1	**0.7250**
all-mpnet-base-v2	gpt-4o-mini	0.7000
all-MiniLM-L6-v2	o3-mini	0.5750
all-MiniLM-L6-v2	gpt-4.1	0.6000
all-MiniLM-L6-v2	gpt-4o-mini	0.6000
LaBSE	o3-mini	0.6250
LaBSE	gpt-4.1	0.6500
LaBSE	gpt-4o-mini	0.6500
para-mul-mpnet-v2	o3-mini	0.6750
para-mul-mpnet-v2	gpt-4.1	0.5750
para-mul-mpnet-v2	gpt-4o-mini	0.6000

Table 7: Track B results across different LLM-based schema extractors and embedding models on Track B (Dev set). Bold indicates the best overall score.

Model ID / Backbone
`sentence-transformers/all-mpnet-base-v2`
`sentence-transformers/all-MiniLM-L6-v2`
`sentence-transformers/LaBSE`
`sentence-transformers/paraphrase-multilingual-mpnet-base-v2`

Table 8: Backbone models evaluated for both tracks.

These fusion weights are selected via manual tuning on the development set rather than analytical optimization. A comparative ablation of alternative weight settings is reported in Section 5 (Table 2). The fused embeddings are L2-normalized before submission.

A.5 Model Selection Across Backbones

To validate our choice of backbone encoder, we conducted experiments with four sentence transformer models on both tracks under identical hyperparameter settings:

All backbone experiments use the same hyperparameters, data splits, and random seeds. The final submission for both tracks uses `all-mpnet-base-v2`, selected based on development set performance.

B Prompt Templates

This appendix presents the prompt template used for narrative element extraction in the Track B pipeline. Narrative components are extracted from each story via the OpenAI Batch API using the `o3-mini` model with a temperature of 0.3 and a maximum of 2,000 completion tokens. The response format is constrained to JSON output. The extracted components: theme, plot events, and outcome are subsequently used to construct multi-view training triplets and to generate view-specific embeddings at inference time. The prompt design is grounded in computational narratology principles, specifically the notions of Tellability and Changes of State. Events are selected based on whether they cause physical, mental, or social modifications and whether they produce affective fluctuation or alter character relationships.

C Computational Resources

All training and inference experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB VRAM.

Track A training uses automatic mixed precision (AMP) with gradient scaling, which reduces both memory consumption and wall-clock time. Track B training processes batches of multi-view triplets with periodic CUDA cache clearing to manage GPU memory. Both tracks perform evaluation after each epoch, which is included in the reported training times.

System Prompt & Instructions

```
You are an expert in Computational Narratology and Narrative Analysis. Your task
is to extract the core narrative structure from the story summary below, focusing
on "Tellability" and "Changes of State."
```

STORY SUMMARY:

```
{story_summary}
```

EXTRACTION INSTRUCTIONS:

[leftmargin=*, nosep]

- **Theme** (1-3 sentences): Identify the central controlling idea or semantic abstract of the story. Focus on the underlying motivation or the specific "script" (e.g., revenge, redemption, sacrifice) that governs the narrative logic.
- **Plot Events** (5-10 key events): Extract the main chronological sequence of events.
  [leftmargin=1.5em, nosep]
  - *Change of State:* Only select events that cause a physical, mental, or social modification in the characters or the world.
  - *Affective Impact:* Prioritize events that create "Affective Fluctuation" (shifts in tension) or alter the relationships between characters.
  - *Structure:* Phrase each event concisely as [Subject] + [Predicate/Action] + [Object/Outcome].
- **Outcome** (1-2 sentences): Describe the final "Resolution State." Detail how the character network or the narrative world has fundamentally changed compared to the beginning.

RESPONSE FORMAT (JSON):

```
{
  "theme": "...",
  "plot_events": ["Event 1", "Event 2", "..."],
  "outcome": "..."
}
```

IMPORTANT: Filter for High Tellability. Ensure strict chronological order. Return ONLY the JSON object.

Table 9: Prompt template for narrative element extraction used in Track B. The template instructs the model to decompose each story into theme, plot events, and outcome following computational narratology criteria.