Comhis at SemEval-2026 Task 4: Embedding-Space Adaptation and LLM-Assisted Inference for Narrative Similarity

Ke Shu
University of Helsinki
ke.shu@helsinki.fi

Eetu Mäkelä
University of Helsinki
eetu.makela@helsinki.fi

Mikko Tolonen
University of Helsinki
mikko.tolonen@helsinki.fi

Abstract

We present a two-stage system for the SemEval Narrative Similarity task that separates representation learning from comparative decision making. In Track B, we adapt a frozen large-scale embedding model using a lightweight projection layer trained with a triplet objective and hard example mining, producing a task-specific similarity space. In Track A, similarity scores derived from the adapted embedding space are incorporated into a large language model, which performs the final binary decision. On the official test set, our system achieves 0.68 accuracy on Track A and 0.66 on Track B, clearly outperforming the provided baselines and ranking 20th out of 44 teams on Track A and 10th out of 27 teams on Track B. These results demonstrate that lightweight embedding adaptation, combined with LLM-based reasoning, provides an effective approach for modeling high-level narrative similarity.

1 Introduction

In this shared task, narrative similarity is defined at a structural level, focusing on shared themes, event progressions, and outcomes rather than surface lexical overlap. The SemEval Narrative Similarity task evaluates this ability in two complementary settings: pairwise narrative comparison (Track A) and similarity-based retrieval using narrative representations (Track B).

We adopt a two-stage embedding-centered architecture. For Track B, we start from a pretrained large embedding model and adapt the representation space using a lightweight linear projection layer trained with a triplet loss objective. The base embedding model remains frozen, and only the projection layer is optimized. We further incorporate hard example mining and weighted sampling to emphasize difficult narrative distinctions during training.

For Track A, we leverage the adapted embedding space learned in Track B to compute similarity scores between the anchor and candidate stories. These similarity scores are explicitly provided to a large language model as auxiliary signals within the prompt. The final decision is produced by the language model, which integrates embedding-based similarity cues with its own narrative understanding.

Our system clearly outperforms the official baselines on both tracks. Experimental analysis shows that hard example mining plays a critical role, while synthetic training data provides noticeable performance gains, although it is not the primary driver of improvement compared to embedding space adaptation. The results demonstrate that lightweight embedding adaptation, combined with embedding-informed language model reasoning, provides an effective and computationally efficient solution for narrative similarity.

2 Background

We participated in SemEval-2026 Task 4: Narrative Story Similarity and Narrative Representation Learning (Hatzel et al., 2026). The task asks systems to identify narratively similar stories based on three core aspects: **abstract theme**, **course of action**, and **outcomes** of a story. Participants are given an anchor story and two candidate stories, and the goal is to predict which of the two candidates is narratively closer to the anchor.

The shared task consists of two complementary tracks:

- **Track A**: Narrative Similarity Judgment — Systems are required to perform pairwise comparison between two candidate narratives given an anchor, and assign which one is closer in narrative similarity to the anchor. This is evaluated as a binary judgment task, with accuracy as the primary metric.

- **Track B**: Narrative Representation Learning

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 862–868

— Systems must learn suitable vector representations (embeddings) of narratives such that similarity computations in the learned space reflect narrative similarity. The output for Track B is a set of narrative embeddings for the evaluation data, which are then used to assess representation quality via similarity ranking or retrieval metrics.

Our approach addresses both tracks by learning an embedding model optimized on narrative similarity data and generating embeddings for use in both the pairwise comparison and representation tasks.

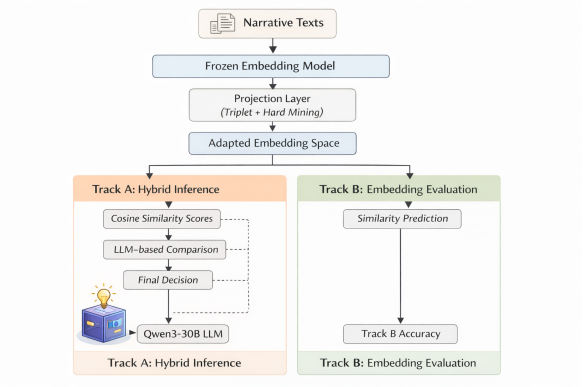

Figure 1: Overview of our two-stage framework. Track B learns an adapted embedding space; Track A uses the resulting similarity signals for hybrid inference.

3 Related Work

Narrative similarity has been studied in story understanding and event modeling, where early approaches relied on symbolic or schema-based representations and induced narrative event structures from text (Hatzel and Biemann, 2024). While such methods provide interpretability, they often depend on engineered assumptions and may struggle with the diversity of real-world narratives.

With large pre-trained language models, embedding-based methods have become the dominant paradigm for text similarity, retrieval, and ranking (Chen et al., 2022). However, off-the-shelf embeddings are typically optimized for short inputs and tend to emphasize surface-level semantic overlap, which can limit their ability to capture higher-level narrative properties in longer stories (Hatzel and Biemann, 2024).

To better align representations with task-specific similarity scores, recent work has adopted metric learning objectives such as contrastive and triplet losses. Sentence-BERT and related models fine-tune encoders with siamese/triplet architectures to learn embedding spaces where similar texts are closer than dissimilar ones (Kumar and Kumar, 2024). Beyond full fine-tuning, lightweight embedding adaptation (e.g., adding a small projection layer on top of frozen embeddings) has been shown to achieve a favorable effectiveness–efficiency trade-off under limited supervision (Younus and Qureshi, 2025). In addition, hard negative mining improves metric learning by emphasizing confusable instances during training (de Souza P. Moreira et al., 2025).

Our approach follows the embedding-adaptation paradigm by training a projection layer over frozen embeddings with a triplet objective and hard example mining, and integrating the resulting similarity space into LLM-based inference.

4 System overview

4.1 Task Formulation and Overall Approach

The SemEval-2026 Narrative Similarity task evaluates systems in two complementary settings. In both tracks, the core problem can be formulated as a comparative similarity task: given an anchor narrative x and two candidate narratives y_1 and y_2, the system must determine which candidate is narratively closer to the anchor (Hatzel et al., 2026).

While **Track A** evaluates this comparison as a binary decision, **Track B** focuses on learning narrative representations that support similarity-based ranking or retrieval. We address the two tracks using a two-stage embedding-centered framework. First, we learn a task-adapted embedding space via metric learning (Track B). Second, for Track A, we integrate similarity scores derived from this adapted embedding space into a large language model (LLM), which produces the final binary decision.

This design decouples representation learning from decision making: the adapted embedding space encodes narrative similarity geometry, while the LLM performs the final reasoning step in pairwise comparison, see Figure 1.

4.2 Embedding-Based Narrative Representation (Track B)

Narratives are represented using dense vector embeddings derived from a large-scale pretrained model. We employ **Qwen3-Embedding-8B** and use the vLLM framework for efficient embedding

inference. The base embedding model remains frozen throughout training.

Instead of full fine-tuning, we perform embedding space adaptation by introducing a trainable linear projection layer. Given an input embedding $e \in \mathbb{R}^d$, the adapted representation $\tilde{e}$ is computed as:

$$\tilde{e} = \frac{We}{\|We\|_2}, \tag{1}$$

where $W \in \mathbb{R}^{d \times d}$ is a learnable projection matrix initialized as a near-identity matrix.

This formulation preserves the general semantic structure of the pretrained embedding space while reshaping it to better reflect task-specific narrative similarity scores. By restricting learning to a lightweight projection layer, the model remains efficient and stable under limited supervision.

4.3 Training Objective and Data Usage

The projection layer is optimized using a triplet-based metric learning objective. Each training instance consists of an anchor narrative x, a positive candidate p, and a negative candidate n derived from the annotation.

Let $\tilde{e}_x$, $\tilde{e}_p$, and $\tilde{e}_n$ denote the projected embeddings. Narrative similarity is computed using cosine similarity, and the model minimizes the triplet loss:

$$\mathcal{L} = \max\left(0,\ m - \left(\mathrm{sim}(\tilde{e}_x, \tilde{e}_p) - \mathrm{sim}(\tilde{e}_x, \tilde{e}_n)\right)\right), \tag{2}$$

where m is a margin hyperparameter.

The base encoder remains frozen and only the projection layer is updated using AdamW optimization.

Training Data. Training uses a combination of synthetic narrative comparison data and officially released example data. A subset of the Track A development data is used for validation and early stopping. This setup aligns model selection with the target evaluation setting.

Hard Example Mining. Narrative similarity scores often involve subtle distinctions, resulting in cases where candidate narratives share high semantic overlap. To address this, we incorporate a hard example mining strategy during training. Specifically, we first compute similarity scores using the frozen base embedding model (before projection-layer adaptation) on the training and development data only. Instances that are incorrectly ranked by the base embeddings are identified as candidate hard examples. Among these, those with small similarity margins are selected and upweighted during projection-layer training using weighted sampling.

Hard example identification is performed only once prior to projection-layer optimization and relies exclusively on training and development predictions. No test labels or test predictions are used in this process. This procedure encourages the projection layer to focus on difficult, low-margin narrative distinctions while avoiding any leakage from the evaluation set.

4.4 Track-Specific Inference Procedures

Track B. At inference time, narratives are encoded using the frozen embedding model and transformed by the learned projection layer. The projected embeddings are directly output and used for similarity-based ranking or retrieval evaluation.

Track A. For Track A, we adopt a hybrid inference strategy. Given an anchor narrative and two candidate narratives, we first compute cosine similarity scores in the adapted embedding space. These similarity scores are then incorporated into the prompt of a large language model.

The LLM receives (1) the anchor narrative, (2) the two candidate narratives, and (3) the embedding-based similarity scores and their difference. The final binary decision (A or B) is produced by the LLM. If the LLM output cannot be reliably parsed, the system falls back to a direct comparison of embedding similarity scores.

This design leverages the structured similarity geometry learned in Track B while allowing the language model to integrate higher-level narrative reasoning. Rather than relying solely on cosine comparison, Track A combines embedding-based evidence with language-model-based decision making.

In practice, the embedding-derived similarity scores serve as strong guidance signals for the LLM rather than hard constraints. We did not implement an explicit conflict resolution mechanism; instead, the LLM integrates these numerical cues with its own reasoning. Empirically, we observe that the model typically aligns its decisions with the relative similarity scores, suggesting that the provided signals effectively anchor the generation process.

5 Experimental Setup

5.1 Data Splits and Usage

We follow the official data splits provided by the shared task. Model training is performed using the released training data together with additional synthetic examples where applicable. Hyperparameter tuning and model selection are conducted exclusively on the development set. The test set is strictly reserved for final evaluation and submission and is not used for training, model selection, or hard example identification.

For Track A, development data is used for validation and early stopping during projection-layer training. For Track B, the adapted embedding representations are evaluated under the official similarity-based evaluation protocol on the corresponding development and test splits.

Hard example identification is performed only on training and development data without using test labels.

5.2 Preprocessing and Representation

Each narrative is treated as a single textual unit without sentence-level segmentation. No additional task-specific normalization or filtering is applied beyond the default preprocessing of the embedding model.

To improve efficiency and reproducibility, projected embeddings are cached and reused during Track A inference. This avoids recomputation and ensures that both tracks operate over the same adapted representation space.

5.3 Training Configuration and Hyperparameters

The projection layer is trained using a triplet loss objective optimized with AdamW. The base embedding model remains frozen throughout training. Hyperparameters are tuned on the development set, and early stopping is applied based on development performance when enabled.

Hard example mining is implemented through weighted sampling. Instances identified as difficult—based on incorrect or low-margin similarity predictions during development-stage evaluation—are assigned higher sampling weights. This approach increases the model's sensitivity to subtle narrative distinctions without introducing additional classifier layers or architectural modifications.

System	Dev A	Dev B	Test A	Test B
Baseline	0.48	0.58	–	–
Full system	0.78	0.82	0.68	0.66

Table 1: Main results on the SemEval Narrative Similarity task. All scores are accuracy.

For Track A inference, cosine similarity scores derived from the adapted embedding space are incorporated into the prompt of a large language model, which produces the final binary decision. If the LLM output cannot be reliably parsed, a fallback decision based on embedding similarity is applied.

Detailed hyperparameter values are reported in the Appendix A.

5.4 External Tools and Libraries

Our experiments are implemented in Python using PyTorch.[1] Embeddings are generated with the vLLM framework using a large pre-trained embedding model.[2] Additional libraries include NumPy and pandas for data handling. All experiments use publicly available implementations with default settings unless specified otherwise.

5.5 Evaluation Metrics

We follow the official evaluation metrics defined by the shared task. Track A is evaluated using accuracy over pairwise similarity comparisons. Track B is evaluated using the task-provided similarity-based evaluation protocol. No additional evaluation measures are introduced.

6 Results

This section reports the quantitative performance of our system on both tracks, followed by ablation studies and qualitative analysis.

6.1 Main Results

Table 1 summarizes the performance of our system on the official development and test sets. Accuracy is reported for both Track A and Track B.

For Track A, the reported scores reflect the hybrid inference strategy that combines adapted embedding similarity with LLM-based decision making. For Track B, performance reflects the quality of the adapted embedding representations evaluated under the official protocol.

[1] https://pytorch.org

[2] https://github.com/vllm-project/vllm

Setting	Dev A	Dev B	Test A	Test B
No synthetic data	0.63	0.63	0.57	0.55
No hard examples	0.60	0.61	0.54	0.53
Full system	0.78	0.82	0.68	0.66

Table 2: Ablation results on development and test sets. All scores are accuracy.

On the development set, our full system clearly outperforms the official baselines, achieving 0.78 accuracy on Track A and 0.82 on Track B, compared to baseline accuracies of 0.48 and 0.58 respectively. This confirms that embedding space adaptation significantly improves narrative similarity modeling.

On the test set, our system achieves 0.68 accuracy on Track A and 0.66 on Track B. Although absolute performance decreases relative to development results, the performance advantage over the baselines remains clear, indicating reasonable generalization to unseen narratives. This performance gap is likely due to distributional differences between the development and test sets, particularly in narrative complexity and ambiguity. In addition, the relatively small development set may lead to optimistic estimates during model selection, while hard example mining may bias the model toward borderline cases seen during development.

6.2 Ablation Analysis

To better understand the contribution of individual components, we conduct ablation experiments removing synthetic data and hard example mining. All variants share the same embedding model and projection architecture, differing only in training configuration.

Effect of Synthetic Data. Removing synthetic training data results in a substantial drop in performance (approximately 10–11 points across evaluation settings), indicating that synthetic examples play an important role in improving generalization. While embedding space adaptation provides the primary modeling capacity, synthetic data contributes meaningful additional coverage of narrative patterns, particularly for less frequent or structurally diverse cases.

Effect of Hard Example Mining. Disabling hard example mining leads to a further drop in accuracy across both tracks. The decrease is consistent on development and test sets, indicating that emphasizing difficult comparisons during training plays a critical role in shaping a more discriminative embedding space.

Overall, the ablation results show that embedding space adaptation provides the main performance improvement, while hard example mining further refines decision boundaries. Synthetic data contributes auxiliary gains, while hard example mining shows a consistently strong impact under our current training configuration. We note that this comparison reflects relative importance within this setup rather than isolated training regimes.

6.3 Error Analysis

Analysis of misclassified test instances reveals that errors are relatively evenly distributed across labels, suggesting no strong prediction bias toward either candidate. Most errors occur when the similarity difference between candidates is small, indicating that borderline narrative distinctions remain challenging even after embedding adaptation.

Qualitative inspection suggests three recurring patterns:

- **Lexical Overlap Bias:** In some cases, surface-level lexical similarity dominates deeper narrative structure.
- **Theme–Outcome Divergence:** Stories sharing abstract themes but differing in outcomes sometimes lead to incorrect similarity scores.
- **Structural Ambiguity:** Loosely structured or episodic narratives yield less stable similarity assessments.

We further observe that embedding similarity and LLM-based decisions are generally aligned. However, in some cases—particularly those involving shared themes but different outcomes—the LLM is able to override misleading embedding signals by incorporating higher-level narrative reasoning.

6.3.1 Comparison Between Tracks

Track B directly evaluates embedding-space geometry, while Track A integrates embedding similarity with LLM-based reasoning. The slightly stronger

performance of Track A suggests that embedding-informed LLM inference can partially compensate for limitations in pure geometric similarity comparison.

Overall, these findings demonstrate that lightweight embedding adaptation significantly improves narrative similarity modeling, and that embedding-assisted LLM reasoning further enhances pairwise decision accuracy.

7 Conclusion

We presented a two-stage system for the SemEval Narrative Similarity task that decouples representation learning from decision making. A frozen large-scale embedding model is adapted using a lightweight projection layer trained with a triplet objective and hard example mining, producing a task-specific similarity space (Track B). For Track A, similarity scores from this adapted embedding space are incorporated into a large language model, which performs the final binary decision. Results on both development and test sets show that embedding space adaptation provides substantial gains over the baselines, while embedding-informed LLM inference further improves pairwise narrative judgments. These findings highlight the effectiveness of efficient embedding adaptation combined with reasoning-level integration for modeling high-level narrative similarity.

Limitations

Our approach relies on a frozen embedding model with a lightweight projection layer, which limits its ability to model deeper narrative structure beyond what is already encoded in the pretrained representations. In addition, our analysis suggests that the effectiveness of synthetic training data depends on the quality and diversity of the generated examples, and may vary across settings.

Furthermore, the use of a 30B-parameter LLM for inference introduces non-trivial computational cost and latency, which may limit the practicality of the approach in resource-constrained or real-time settings.

Acknowledgments

We would like to thank the organizers of SemEval-2026 Task 4 for their work in designing and coordinating the shared task. This research was supported by the European Union's Horizon Europe research and innovation programme through the MSCA Doctoral Networks 2022 under Grant Agreement No. 101120349 and Grant Agreement No. 101119511. We also gratefully acknowledge CSC – IT Center for Science, Finland, for granting access to the LUMI supercomputer, which is owned by the EuroHPC Joint Undertaking and hosted by CSC in Finland in collaboration with the LUMI consortium.

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.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2025. Nv-retriever: Improving text embedding models with effective hard-negative mining. *Preprint*, arXiv:2407.15831.

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. 2024. Story embeddings — narrative-focused representations of fictional stories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5931–5943, Miami, Florida, USA. Association for Computational Linguistics.

Anand Kumar and Hemanth Kumar. 2024. scaLAR SemEval-2024 task 1: Semantic textual relatednes for English. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 902–906, Mexico City, Mexico. 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.

A Hyperparameter Settings

This appendix reports the detailed hyperparameter values used in all experiments described in Section 5.

Table 3: Hyperparameter configuration used for embedding adaptation and inference.

Hyperparameter	**Value**
Embedding model	Qwen/Qwen3-Embedding-8B
Projection layer	Linear, dimension-preserving
Training objective	Triplet loss
Learning rate	1×10^{-4}
Batch size	32
Training epochs	30
Triplet margin	0.2
Temperature	0.1
Weight decay	5×10^{-4}
Early stopping patience	15 epochs
Hard example ratio (dev)	0.2
Hard example weight	2.0

For Track A inference, we additionally use a large instruction-tuned language model (`Qwen/Qwen3-30B-A3B-Instruct-2507`) via a vLLM OpenAI-compatible API. The maximum generation length is set to 64 tokens.