Mendel292 at SemEval-2026 Task 4: Disentangled Narrative Embeddings for Story Similarity

Sankalpa Rijal[1], **Maurício Gruppi**[1], **Justin DeBenedetto**[1]
[1]Department of Computing Sciences, Villanova University
{srijal,mgouveag,justin.debenedetto}@villanova.edu

Abstract

This paper describes Mendel292, our system for SemEval-2026 Task 4 on Narrative Story Similarity. We introduce a narrative encoder that decomposes story representations into explicit subspaces for abstract theme, course of action, and outcome, built on a pre-trained sentence embedding model and a trainable BiLSTM projection layer with a triplet margin loss objective. We augment the training set via back-translation and incorporate weakly supervised multi-task objectives derived from unsupervised narrative clustering. The proposed architecture was designed to learn a latent representation of narratives in a few-shot setting due to a limited amount of training data. Despite using a rich pre-trained transformer, the model was outperformed by an unsupervised pooling approach on the classification task. While our systems do not match the top leaderboard scores, they allow us to systematically study the effects of subspace factorization, weak labels, and data augmentation on narrative similarity modeling.

1 Introduction

The core purpose of stories is to share ideas, communicate experiences, and promote influence, among other things. The implied narrative through stories affects individual and social behavior, cognition, and emotions (Boyd, 2018). Narrative similarity is typically defined as similarity at the level of events, themes, and claims rather than surface lexical overlap (Waight et al., 2025) with human judgment focusing on the core plot rather than wording or environmental setting (Chaturvedi et al., 2018; Chun, 2024). Recent evaluations show LLMs often miss causal and holistic narrative understanding despite strong pattern-matching abilities (De Langis et al., 2025; Inani et al., 2025).

SemEval-2026 Task 4 advances this challenge by formalizing narrative similarity along three dimensions—*abstract theme*, *course of action*, and *outcome*—using story summaries where systems compare two candidates against an anchor via triplet classification (Track A) or embedding cosine alignment (Track B) (Hatzel et al., 2026; Hatzel and Biemann, 2024). Specifically, in Track A, the input is a triple of narratives ($Anchor$, A, B)–summaries of stories from Wikipedia–and the output is a binary decision as to whether $Anchor$ is closer to A than to B. Track B consists of projecting the narratives onto a latent space that properly captures narrative similarities. A key challenge is the limited amount of gold-standard labels, with only 200 triplets available during development.

Early approaches to story similarity relied on global sentence and document embeddings, which blur distinct narrative factors into single representations and can struggle with long, multi-event texts (Hatzel and Biemann, 2024). Recent studies explore richer structure by leveraging LLMs as narrative judges or teachers, scoring story pairs along fine-grained dimensions before training smaller models (Chun, 2024). Hatzel and Biemann (2024) show that contrastive and triplet-loss-based embeddings, learned via metric learning, produce geometrically meaningful semantic distances in narrative-focused representations. Reimers and Gurevych (2019) and Chavan (2025) further demonstrate that contrastive losses on weakly supervised text pairs, curated via BM25, Wikipedia links, and Reddit threads, yield strong representations without hand-labeled data (Wang et al., 2024).

Our submitted system learns disentangled narrative embeddings by decomposing the space into explicit theme/action/outcome subspaces, trained with triplet margin loss on synthetic data and transductive (i.e., using weak labels mined from the *unlabeled* test data during training (Vapnik, 1998)) multi-task objectives using weak programmatic labels on synthetic + unlabeled test data. In the transductive variant, only mined sub-labels from unlabeled test set are used; no gold test annota-

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2575–2582

tions are accessed. This choice is intended to study whether task-structure priors transfer under the official rules. We use a frozen `all-MiniLM-L6-v2` SBERT model to encode the input texts, and a trainable Bi-LSTM layer to create projections of the narratives with online hard triplet mining and incorporating test data transductively. To tackle the lack of training data and enhance the training of the Bi-LSTM model, we employ a data augmentation procedure via round-trip translation.

2 System Overview

The system is designed to learn structured narrative embeddings through a joint multi-task learning framework. Instead of learning a single holistic representation for each story, the model explicitly decomposes the embedding space into semantically meaningful subspaces corresponding to three narrative aspects: Abstract Theme, Course of Action, and Outcome.

Each story is first segmented into sentences using a lightweight rule-based approach. The embeddings for each sentence are extracted using a frozen pretrained transformer encoder. These fixed embeddings are then passed to a trainable sequence encoder that aggregates temporal information across the sentence sequence to form a single story-level embedding. The final embedding vector $v \in \mathbb{R}^d$ is partitioned into task-specific slices,

$$v = [v_t, v_a, v_o],$$

where each slice is supervised by its own auxiliary objective: *theme*, *action*, and *outcome*, respectively. This explicit allocation of dimensions encourages geometric separation between narrative components while enabling shared global structure. Figure 1 presents an overview of the system.

Training is performed jointly on (1) synthetic triplet-labeled data for metric learning and (2) multitask-labeled synthetic + test data for auxiliary supervision. The mined sub-labels from the test data is incorporated transductively during training to improve generalization to the evaluation distribution.

2.1 Architecture

Encoder Backbone: The baseline system uses the frozen `all-MiniLM-L6-v2` SBERT model (Reimers and Gurevych, 2019) as a sentence encoder for computational efficiency and strong performance on semantic similarity. For a text S with n sentences, the encoder produces:

$$S = [s_1, \ldots, s_n], s_i \in \mathbb{R}^{384}$$

Sequential Projection Encoder: To model narrative progression, the sequence of sentence embeddings is processed by a Bidirectional LSTM (Bi-LSTM)(Graves and Schmidhuber, 2005). The final hidden states from both directions are concatenated, $h = [\overrightarrow{h}_n \| \overleftarrow{h}_1] \in \mathbb{R}^{2H}$, and mapped to the target embedding space using a linear projection with dropout regularization followed by ℓ_2-normalization:

$$v = \frac{Wh + b}{\|Wh + b\|_2}, \quad v \in \mathbb{R}^d$$

Subspace Allocation: the total embedding dimension d is divided across tasks according to predefined ratios. Let d_t, d_a, d_o denote the dimensions for theme, action, and outcome, respectively:

$$d = d_t + d_a + d_o$$

Each slice is trained with task-specific supervision while sharing the same encoder backbone.

2.2 Training Objectives

The model is optimized using a weighted combination of global metric learning and auxiliary multi-task clustering losses:

$$\mathcal{L}_{total} = \lambda_{triplet}\mathcal{L}_{triplet} + \lambda_{mt} \sum_{k\in\{t,a,o\}} \big(w_k \mathcal{L}_k\big) + \underbrace{\lambda_{mt}^{(test)} \sum_{k\in\{t,a,o\}} \big(w_k \mathcal{L}_k\big)}_{\text{active only with transductive test sub-labels}}$$

Here, $\lambda_{triplet}$ and λ_{mt} control the relative importance of global and auxiliary objectives, while w_k distributes the weight across tasks. $\lambda_{mt}^{(test)}$ is used if sub-labels mined from unseen test data are used for transductive training; otherwise assigned 0.

2.3 Triplet Margin Loss

To capture overall narrative similarity, the model is trained using a standard triplet margin loss on the full embedding v. A triplet is a 3-tuple of *anchor*, *positive* and *negative* samples. The goal is to learn a latent space that keeps anchor and positive close, while moving anchor and negative apart. Triplets (a, p, n) are constructed offline from the dataset:

$$\mathcal{L}_{triplet} = max\Big(0,\, d(a,p) - d(a,n) + M\Big),$$

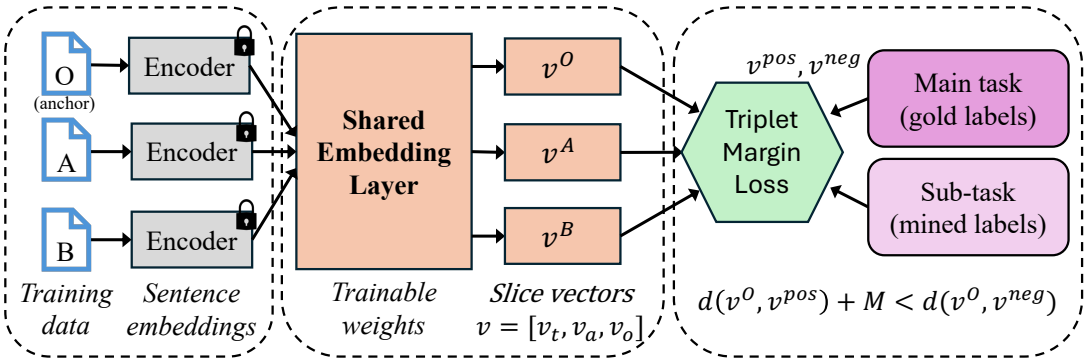


Figure 1: Overview of the system. Input text is processed by a frozen encoder, which is fed through a trainable layer with a triplet margin loss objective using labels from gold annotations and label mining.

where d is the cosine distance, and M is a hyperparameter controlling the gap between positive and negative samples. This objective enforces global geometric structure in the shared embedding space.

2.4 Auxiliary Multi-Task Subspace Loss

To encourage disentanglement of narrative factors, each embedding slice is ℓ_2-normalized and trained using weak labels corresponding to its narrative aspect. Within each mini-batch, online hard triplet mining is used to form positive and negative pairs inside the task-specific subspace. Hard triplet mining chooses triplets where the negative is initially close, and the positive is initially far from the anchor.

For task k:

$$\mathcal{L}_k = \frac{1}{N}\sum_{i=1}^{N} max\Big(0,\, d_E(a_i, p_h) - d_E(a_i, n_h) + \beta\Big)$$

where p_h is the farthest (hardest) same sub-label embedding from anchor a_i, and n_h is the closest different sub-label embedding, computed via batch distance matrix masking. d_E denotes Euclidean distance, and β is the sub-label margin.

This intra-cluster positive + inter-cluster negative mining encourages tight, discriminative subspace clusters. Because the sub-labels are mined heuristically from positional slices and clustering, the subspace losses should be understood as noisy structural supervision.

2.5 Data Augmentation via Back-Translation

To enhance the training on the 200 triples in the gold development set, we augment the data via round-trip translation. Each story is translated into French, Russian, and Hindi, then back to English using `deep_translator`'s[1] `GoogleTranslate` interface. This generates three versions per story while preserving narrative structure, and yields up to 9× more triples per pivot language, producing about 1,800 total augmented examples. Paraphrase quality, assessed with BERTScore (Zhang et al., 2020), remains high (French 0.87 ± 0.05, Russian 0.82 ± 0.05, Hindi 0.83 ± 0.05), confirming semantic consistency (Fig. 2).

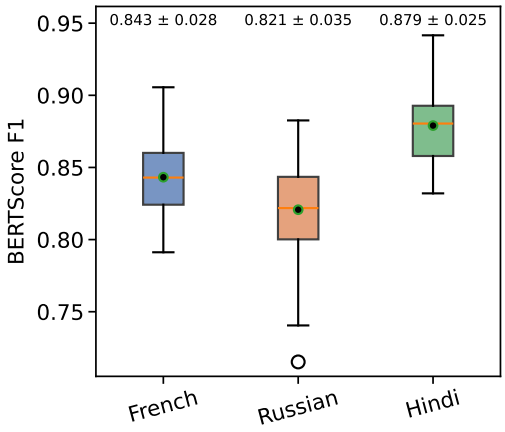


Figure 2: BERTScore F1 distributions across pivot languages (mean ± stdev annotated). Circles denote outliers.

2.6 Label Mining (Weak Supervision)

Since the dataset provides no ground-truth labels for narrative sub-components, we mine labels via unsupervised clustering on positional text slices, assuming chronological narrative structure: first 25% sentences (Theme/setup), middle 50% (Action/conflict), final 25% (Outcome/resolution).

Each slice is encoded with SBERT, then clustered by K-($k = 8$ Theme, $k = 5$ Action, $k = 4$

[1] https://pypi.org/project/deep-translator/

Outcome). Cluster assignments serve as weak labels for multi-task subspace training.

2.7 Weak-Label Subspace Training

Weak sub-labels (mined labels for theme, action, and outcome subspaces from the clustering pipeline in Section 2.6) are computed for all samples in synthetic and augmented development sets, and for the test set when training transductively. These sub-labels drive the online batch hard mining in Section 2.4 during joint training with triplet loss.

3 Experimental Setup

3.1 Data

The data is split into four distinct groups: A development set (dev) containing annotated triples, used for training; a test set used in evaluation; an augmented set consisting built from back-translations of triples in the development set; and a synthetic set with triples created using generative text models. Figure 3 shows the number of triples in each set.

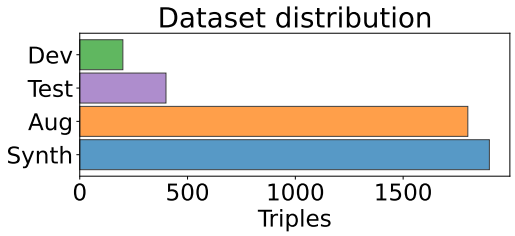


Figure 3: Distribution of triples in the datasets used in this work.

3.2 System Configurations and Hyperparameters

Hyperparameters were optimized using Optuna (Akiba et al., 2019), a framework for parameter search that employs the Tree-structured Parzen Estimator (TPE) sampler for efficient Bayesian optimization. The resulting hyperparameters are shown in Appendix B.

4 Results

4.1 Track A

The following are the results for Track A. For each model in the experiments, input triples ($Anchor$, A, B) are encoded as vectors v^O, v^A, v^B and the decision is made by whichever cosine distance $d(v^O, v^A)$ or $d(v^O, v^B)$ is smaller.

4.1.1 SBERT Baselines

Table 1 reports the results for the baseline results. We used models `all-MiniLM-L6-v2` and `all-MiniLM-L12-v2`. The L6 and L12 variants show similar performance on the development set, while post-task evaluation on the test data indicates that the L6 variant is more robust in the test distribution.

Models	Dev Acc.	Test Acc.†
L6	0.550	0.598
L12	0.560	0.580

†Post-task evaluation.

Table 1: Baseline SBERT performance.

4.1.2 Linear & Multilayer Network Projections

Table 2 reports the results for linear and MLP projections for global embeddings. The training on augmented data shows improvement over the increase in model complexity. For both the linear and MLP projections, the output layer has 256 components.

Models	Avg. CV Acc.	Test Acc.†
LinearAug	0.506	0.490
LinearSynDev	0.490	0.488
LinearSynAug	0.738	0.470
MLPAug	0.548	0.500
MLPSynAug	0.738	0.500

†Post-task evaluation.

Table 2: Performance of linear and MLP models using SBERT inputs.

4.1.3 Bi-LSTM

Table 3 reports the results for the sequence-based models trained with a set triplet loss margin. The introduction of weakly labeled sub-labels (SequentialL6AugInd) significantly improved development performance. The remaining dev-to-test gap suggests that the model can fit weak structural cues on development data but still struggles to transfer those cues reliably under unseen narrative distributions. Our official leaderboard submission (SequentialL6SynAugTestTrd) utilized a transductive approach by incorporating test-set sub-labels. The Bi-LSTM has two layers with hidden dimension 128 and output dimension 256.

Models	Avg. CV Acc.	Test Acc.†
SequentialL6Aug	0.541	0.495
SequentialL6SynAug	0.738	0.468
SequentialL6AugInd	0.593	0.573
SequentialL6SynAugInd	0.794	0.517
SequentialL6SynAugTestTrd	–	**0.565***

†Post-task evaluation. * Official submission result.

Table 3: Performance of sequential models.

4.1.4 Mean-Max Pooling with Sequential Structural Alignment

This ensemble method uses Mean-Max cosine similarity, and a margin-invoked structural alignment. We identify the segments with the maximum similarity between the anchor and the query stories in the triple, and apply a cutoff to determine which story is more similar. This approach showed an increase in performance, which suggests that a simpler structural heuristic can outperform learned projections when the dataset is small and the main signal comes from coarse narrative alignment rather than dense parameterized representation learning. The results are shown in Table 4.

Model	Dev Acc.	Test Acc.
Mean-MaxL6	0.615	0.640
Mean-MaxL12	0.590	0.655

Table 4: SBERT and sequential structural ensemble.

4.2 Track B

Table 5 presents the consolidated results for Track B. The divergence between development accuracy and test robustness in the `SequentialL6AugInd` configuration suggests a high sensitivity to sub-label noise in the consistency domain.

Models	Dev Acc.	Test Acc.†
L6-Baseline	0.550	0.585
L12-Baseline	0.565	0.548
L6LinearAug	0.505	0.528
L6MLPAug	0.510	0.473
L6SequentialAug	0.505	0.490
L6SequentialSynAug	0.505	0.510
L6SequentialAugInd	0.660	0.502
SequentialL6SynAugTestTrd	–	**0.58***

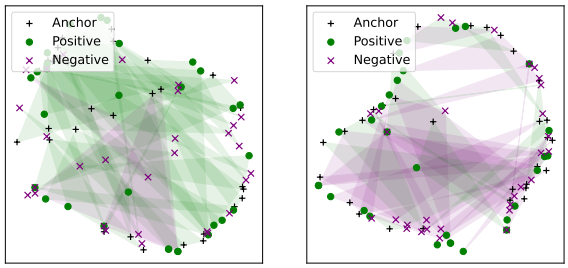
†Post-task evaluation. * Official submission result.

Table 5: Results for Task B: embedding similarity.

4.2.1 Quality of the Narrative Embeddings

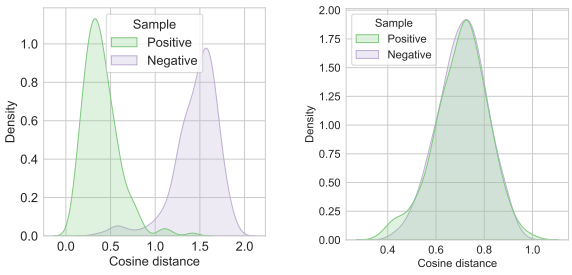
To evaluate the quality of the learned narrative embeddings, we compute the distributions of cosine distances from the anchor vectors to the positive and negative vectors in the development set. For the best performing model (L6SequentialAugInd), the average anchor-positive distance is 0.40 ± 0.2 and the average anchor-negative distance is 1.46 ± 0.25.

Additionally, we used UMAP (McInnes et al., 2018) to project the high-dimensional narrative vectors onto a 2-dimensional space for a visual inspection of the results. The embeddings are seen in Figure 4, which also shows triangles whose vertices are $(anchor, positive, negative)$. Green triangles indicate correctly ordered triplets, i.e., those where the anchor distance is closest to the positive than to the negative sample, purple triangles indicate incorrect triplets. Figure 5 shows the distribution of cosine distances between positive and negative pairs for the baseline and trained L6SequentialAugInd model. Additional figures and model details can be found in Appendix A.



(a) L6SequentialAugInd (b) SBERT L6-Baseline

Figure 4: 2D UMAP projection of narrative vectors. Green triangles denote correctly ordered triplets, purple denote incorrectly ordered triplets. (a) has more correct triples than (b).



(a) L6SequentialAugInd (b) SBERT L6-Baseline

Figure 5: Distributions of cosine distances from anchor to positive and negative embeddings. The baseline (b) shows no separation between positive and negative samples.

4.3 Impact of Data Augmentation

Data augmentation via back-translation boosts cross-validation accuracies. However, test set performance reveals limited generalization benefits, with augmented configurations often underperforming non-augmented counterparts. Mean-Max pooling ensemble, which is unsupervised and thus unaffected by augmentation, introduces noise or domain shift that hinders real-data adaptation.

This analysis indicates augmentation is helpful for overfitting dev-set optimization but not reliable for test robustness in low-data narrative tasks.

5 Conclusion

We presented Mendel292, a narrative similarity system that factors story representations into theme, action, and outcome subspaces on top of frozen SBERT embeddings and a BiLSTM encoder. Using synthetic and back-translated triples together with weakly mined sub-labels, we explored how subspace-aware metric learning interacts with data augmentation and weak supervision in SemEval-2026 Task 4. Our absolute accuracies are modest and below the baseline system, highlighting the potential and brittleness of weakly supervised subspace objectives in this setting. In particular, we observe sensitivity to label noise and a clear gap between synthetic and real narratives.

Some of the limitations in our design include the quality of the weak annotations–the better the model we use, the more reliable are the labels. The positional slicing is currently heuristic and can be improved to capture non-linear narratives where the chronological assumptions do not hold.

In the future, the weak labeling pipeline can be refined using more robust clustering methods, incorporating LLM-based narrative annotations, and jointly learning cluster assignment with the encoder rather than treating them as fixed pseudo-labels. We also aim to train and evaluate our subspace-decomposed encoder on additional narrative benchmarks, such as StoryAnalogy (Jiayang et al., 2023) and AIStorySimilarity (Chun, 2024).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

Brian Boyd. 2018. The evolution of stories: from mimesis to language, from fact to fiction. *WIREs Cognitive Science*, 9(1):e1444.

Snigdha Chaturvedi, Shashank Srivastava, and Dan Roth. 2018. Where have I heard this story before? identifying narrative similarity in movie remakes. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 673–678, New Orleans, Louisiana. Association for Computational Linguistics.

Vinit K. Chavan. 2025. Manifold-constrained sentence embeddings via triplet loss: Projecting semantics onto spheres, tori, and möbius strips. *Preprint*, arXiv:2505.00014.

Jon Chun. 2024. AIStorySimilarity: Quantifying story similarity using narrative for search, IP infringement, and guided creativity. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 161–177, Miami, FL, USA. Association for Computational Linguistics.

Karin De Langis, Jong Inn Park, Andreas Schramm, Bin Hu, Khanh Chi Le, and Dongyeop Kang. 2025. How LLMs comprehend temporal meaning in narratives: A case study in cognitive evaluation of LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29174–29191, Vienna, Austria. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610. IJCNN 2005.

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.

Kalit Inani, Keshav Kabra, Vijay Marupudi, and Sashank Varma. 2025. Modeling understanding of story-based analogies using large language models. *Preprint*, arXiv:2507.10957.

Cheng Jiayang, Lin Qiu, Tsz Ho CHAN, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, Yue Zhang,

and Zheng Zhang. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

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.

Vladimir N Vapnik. 1998. *Statistical Learning Theory*. Wiley, New York.

Hannah Waight, Solomon Messing, Anton Shirikov, Margaret E. Roberts, Jonathan Nagler, Jason Greenfield, Megan A. Brown, Kevin Aslett, and Joshua A. Tucker. 2025. Quantifying narrative similarity across languages. *Sociological Methods & Research*, 54(3):933–983.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Additional Embedding Projections

This appendix includes 2D UMAP projections of embeddings generated by the baseline model, as well as those in the Linear and MLP categories. The results can be seen in Figure 6. Green triangles represent *correct triplets*, i.e. $d(v^O, v^{pos}) + M < d(v^O, v^{neg})$, purple triangles represent *incorrect triplets*.

B Optimal Hyperparameters

The following are the optimal hyperparameters as obtained from Optuna:

1. **Baseline (SBERT)**: Zero-shot evaluation using frozen `all-MiniLM-L6-v2` and `all-MiniLM-L12-v2` models. Story embeddings are compared via cosine similarity without task-specific tuning.

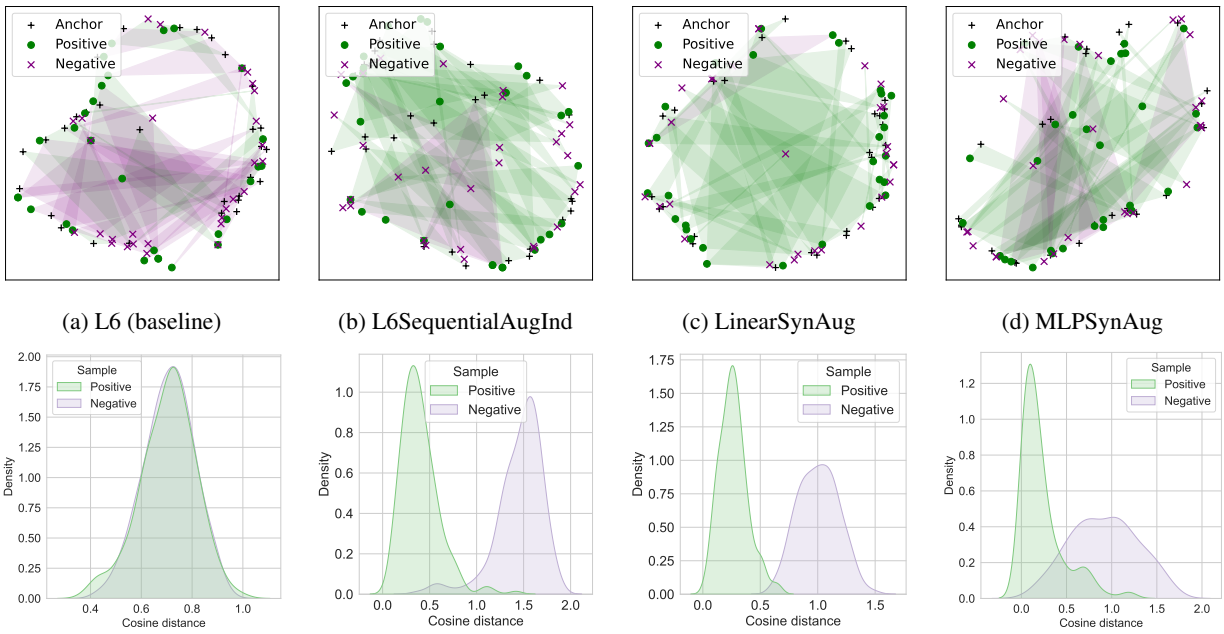
2. **Linear and MLP Projections**: Supervised models trained on global encodings ($d = 364$) from `all-MiniLM-L6-v2`.
   - **Linear**: A single projection layer ($384 \to 256$).
   - **MLP**: A 3-layer architecture ($384 \to 512 \to 512 \to 256$).
   - **Params**: Triplet margin $M = 1.0$, Adam optimizer, $lr = 10^{-3}$, 5-fold cross validation.

3. **LSTM Embeddings**: A BiLSTM encoder processing sentence-wise SBERT sequence embeddings and trained on triplet loss criterion for 5-folds. **Optimal Params**: Hidden dim $= 128$, number of layers$= 2$, Output dim $= 256$, $lr = 1.4 \times 10^{-4}$, Triplet margin(M)$= 1.038$, epochs $= 10$, and batch size $= 64$.

4. **LSTM with Sub-labels (Inductive)**: A BiLSTM encoder trained with a joint objective that combines triplet margin loss on full sequence embeddings and sub-label hard-triplet mining on the task-specific embedding slices. Only the anchor text's sub-labels are used while computing sub-label adjustments. **Optimal Params**: Hidden dim$= 128$, number of layers $= 2$, Output dims$= 256$, Triplet margin(M)$= 1.002$, sub-label margin(β)$= 0.537$, batch size = 64, epochs$= 10$. As discussed in 2.2, $\lambda_{triplet} = 1.437$, $\lambda_{mt} = 0.974$, weights across tasks $(w) = [0.5, 0.3, 0.2]$, and task_ratios$= [0.5, 0.25, 0.25]$.

5. **LSTM with Sub-labels (Transductively using test data)**: Similar approach and set of hyper-parameters used in 4, added with sub-labels mined for test data were used with an additional parameter $\lambda_{mt}^{(test)} = 0.355$.

6. **Mean-Max Pooling with Sequential Structural Alignment**: A hierarchical decision-making algorithm designed to resolve high-entropy narratives. It first computes a primary SBERT-based Mean-Max cosine similarity. If the delta between candidates exceeds γ, the more similar one is chosen; else, a structural tie breaker is initialized. The texts are partitioned into n windows, and similarity is computed across corresponding segments using a diagonal cosine matrix. A tail weight τ is



(a) L6 (baseline) (b) L6SequentialAugInd (c) LinearSynAug (d) MLPSynAug

(e) L6 (baseline) (f) L6SequentialAugInd (g) LinearSynAug (h) MLPSynAug

Figure 6: Top row: 2D UMAP projection of various models used in this study. Bottom row: distribution of cosine distances from anchor to positive and negative embeddings. The SBERT baseline (e) does not exhibit any separation of anchors, positives, or negatives.

applied to the final segment to prioritize narrative resolution. The best parameters observed on dev data are: $\gamma = 0.0183$, $n = 2$, and $\tau = 1.0168$.