Narrative Story Similarity and Narrative Representation Learning

Sina Behnam, Marco Pavanati, Francesco Re,
Andrea Ugliano, Klejsi Zeqaj,

Correspondences: s322796@studenti.polito.it, marco.pavanati@studenti.polito.it, s343404@studenti.polito.it, klejsi.zeqaj@studenti.polito.it, s346317@studenti.polito.it,

Abstract

Defining the characteristics of a story can be a tricky task but becomes even more complicated when faced with the trial of automatization through the usage of LLMs.
This work implements a deep learning solution for SemEval-2026 Task 4: Narrative Similarity. The goal is to determine which of two candidate stories is more similar to an anchor based on semantic aspects like abstract theme, course of action, and outcomes, rather than surface level features. We develop an aspect supervised encoder that combines triplet learning with aspect-aware attention and auxiliary heads for aspect representations. A limiting factor was the lack of an appropriate training dataset which caused the model to quickly become overfitted even remaining undertrained. To address limited labeled data, we use LLM-based augmentation and an aspect extraction pipeline to generate training triplets annotated with theme, action, outcome descriptions, and we provide a modular implementation with optional LoRA fine-tuning.

1 Introduction

The objective of this work is to solve the SemEval narrative similarity task, which requires selecting the story that is narratively closest to a given anchor story (Hatzel et al., 2026). One of the elements of this experiment is the definition of a rappresentation that can describe the story by decomposing its elements. The type of story specifically managed in this task is a brief tale, made of a few sentences. In this type of story a few sections are present pretty distinctly. The first is the introduction which defines the initial context and the problem of the story. It is followed by an action or a series of actions in response to the context, in which some plot twists might be present. The story concludes with either the resolution or the consequences of the actions. Such structure can describe any aspect from the fate of specific characters to the conflict resolution or any moral lessons. Specifically this part defines only the results of the events, so it doesn't cover any intermediate statuses that change during the story. The structure is mainly reppresented by summaries of stories, in particular taken from platforms like Wikipedia. From this type of story the rappresentation is created based on three core characteristics.

Abstract theme It defines the constellation of problems, central ideas and motifs and its focus is on an abstract level, ignoring the setting of the story. The abstract theme describe essentially *what the story wants to tell.*

Course of action It describes how the story unfolds, from start to finish, sequences of events, actions, conflicts, and turning points in a story and the order in which they happen. It gives insights on *what happens in the story*.

Outcomes It describes the results of the plot at the end of the text, without covering intermediate statuses that change later in the story. It gives insights on *what are the consequences of the story*.

From these core characteristics a vectorial representation can be extracted and then compared to others through the means of cosine similarity. The first objective of the study was trying to find a meaningful feature representation that is rich enough to describe a story. The second objective can be viewed as an extension of the first one. The context consists of three stories, an anchor and two stories. The target of the task is to decide which of the two stories is more similar to the anchor in terms of its core characteristics previously described.

2 Background

To achieve the reported results, our work makes use of two well-known architectures as the backbone of our pipeline. We briefly review the key characteristics of these models below before detailing their specific integration in the System Overview.

2.1 Llama-3.1

Llama 3 (Grattafiori et al., 2024) represents a family of large language models built upon a standard dense decoder-only Transformer architecture with several key optimizations for inference efficiency and training stability. Architectural modifications include the use of Grouped Query Attention (GQA) with 8 key-value heads to reduce cache size, a vocabulary of 128K tokens for improved data compression, and Rotary Positional Embeddings (RoPE (Su et al., 2021)) with a base frequency of 500,000 to support context windows up to 128K tokens. The models are pre-trained on a massive corpus of approximately 15.6 trillion multilingual tokens and are aligned via a post-training recipe that integrates Supervised Fine-Tuning (SFT), Rejection Sampling (RS), and Direct Preference Optimization (DPO). The version used in this work is **Llama-3.1-Instruct-Turbo**.

2.2 DistilBERT

DistilBERT (Sanh et al., 2019) is a general-purpose language representation model designed to reduce the computational cost of large-scale pre-trained models like BERT (Devlin et al., 2019) while retaining the majority of their performance. The architecture creates a student model that is 40% smaller and 60% faster than its teacher (BERT-base) by reducing the number of layers by a factor of two and removing token-type embeddings and the pooler. The model is initialized by selecting every other layer from the teacher and is trained using a *triple loss* objective, which combines a distillation loss over soft target probabilities, a masked language modeling (MLM) loss, and a cosine embedding loss to align the hidden states of the student and teacher.

3 System overview

3.1 Data

The task organizers provided two datasets: a development set of 200 triplets and a synthetic training set generated via Large Language Model. However, preliminary experiments revealed a critical issue with the provided training data. When we evaluated a baseline DistilBERT model (`distilbert-base-uncased`) without any fine-tuning or modification, we achieved 97% accuracy on the synthetic training set but only 59% on the development set. This substantial performance gap indicated that the synthetic training data was not representative of the actual task and would not be useful for model development.

3.2 Data Generation Attempts

We initially attempted to generate new training data using Llama-70B-Turbo by prompting the model to create triplets based on narrative similarity principles. However, this approach failed to produce diverse training examples; the model repeatedly generated stories with similar topics and themes, lacking the variety needed for robust training. Due to time constraints, we could not refine the generation pipeline sufficiently neither manually refine the generated data for the needed variability.

3.3 Training Data Construction

Given these challenges, we adopted a different strategy: augmenting the original 200 development triplets. We split this dataset into 140 triplets for training and 60 for evaluation. The 140 training triplets were then processed through two sequential pipelines.

Augmentation via Generation To maximize diversity, we employed a multi-model, multi-temperature augmentation strategy. For each of the 140 original triplets, we generated additional triplets using:

- **Llama-70B-Instruct-Turbo**: 4 temperature settings, 3 instances each
- **Qwen-72B**: 2 temperature settings, 3 instances each

The generation prompt instructed models to: (i) preserve the same theme, action sequence, and outcome patterns, (ii) use completely different characters, names, settings, and genres, (iii) match word counts (±20 words). After filtering non-parsable responses, the final dataset contained 2,383 triplets (140 original + 2,243 generated).

Aspect Extraction For each story in all 2,383 triplets, we used Llama-70B-Turbo to extract descriptions of the three narrative aspects using one-shot json-schema-constrained generation. The prompt explicitly directed the model to ignore surface-level features including writing style, setting details, and character names. This produced aspect-level annotations for all 7,149 stories (2,383 triplets × 3 stories each).

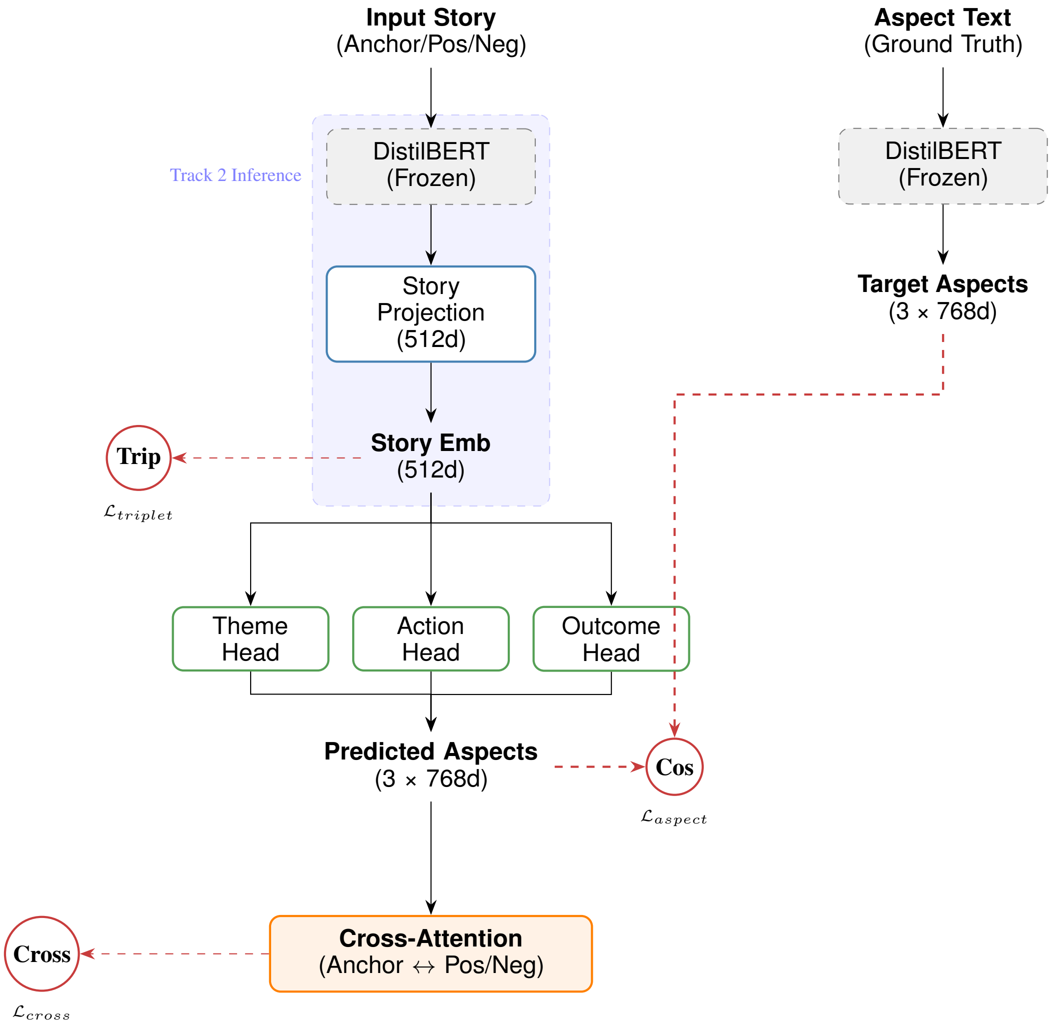


Figure 1: Overall architecture of the Aspect-Oriented Encoder.

3.4 Model Architecture

We implemented an aspect-supervised encoder based on transformer architectures (*AspectSupervisedEncoder*). The motivation for this design is to enable the model to learn not only which stories are similar, but also why they are similar at a reasoning level. Rather than learning similarity solely from the encoded representations, we provide explicit supervision about how each story can be interpreted in terms of its narrative aspects. We leverage the LLM's broader knowledge base to provide this aspect-level reasoning. Since the annotation guidelines specify that human annotators used these three aspects to reason about narrative similarity, we incorporate the same reasoning structure into the encoder architecture. This aspect-oriented supervision encourages the model to learn interpretable, semantically meaningful representations aligned with human narrative understanding. Figure 1 illustrates our aspect-oriented encoder architecture. The model consists of several key components:

3.4.1 Base Encoder

We employed DistilBERT (distilbert-base-uncased) as the backbone encoder due to its computational efficiency and compact size, which facilitate local training. Preliminary experiments exploring Low-Rank Adaptation (LoRA (Hu et al., 2021)) applied to the query and value projection layers yielded no performance improvements. Consequently, we froze all encoder parameters during training to mitigate catastrophic forgetting.

3.4.2 Story Projection Layer

The base encoder output is passed through a projection network that maps the transformer's hidden states to a fixed-dimensional embedding space (512 dimensions). This projection is trained to optimize narrative similarity.

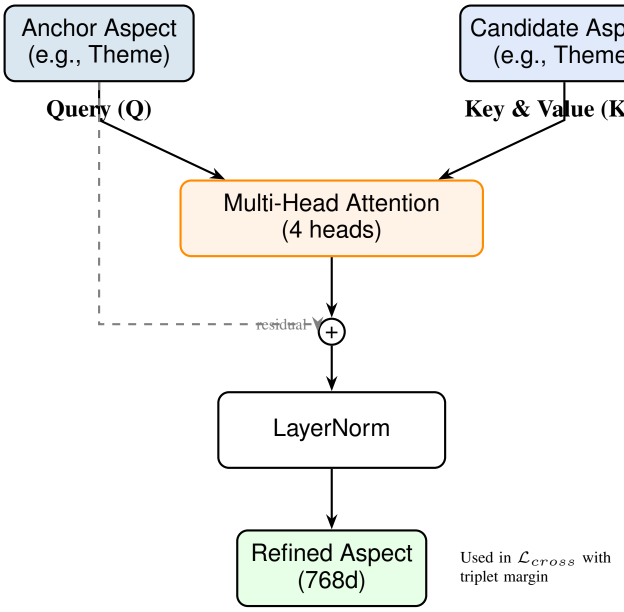


Figure 2: Cross-Attention Module.

3.4.3 Aspect Prediction Heads

Three separate prediction heads generate embeddings for each narrative aspect (abstract theme, course of action, outcomes). Each head takes as input the 512-dimensional story embedding from the projection layer and produces 128-dimensional aspect-specific representations.

3.4.4 Aspect Text Encoding

Ground-truth aspect descriptions extracted by the LLM are encoded through the same frozen DistilBERT encoder. We use the raw output embeddings (768 dimensions) directly as supervision targets for the aspect prediction heads, without additional projection layers.

3.4.5 Cross-Attention Module

To capture aspect-level interactions between stories, we implement cross-attention mechanisms (Vaswani et al., 2017; Kim et al., 2017) that compute refined aspect representations by attending between anchor and candidate stories (Figure 2). This module operates on the 128-dimensional aspect embeddings. The anchor's predicted aspects serve as queries, while the candidate's aspects provide keys and values. A residual connection and layer normalization ensure stable training.

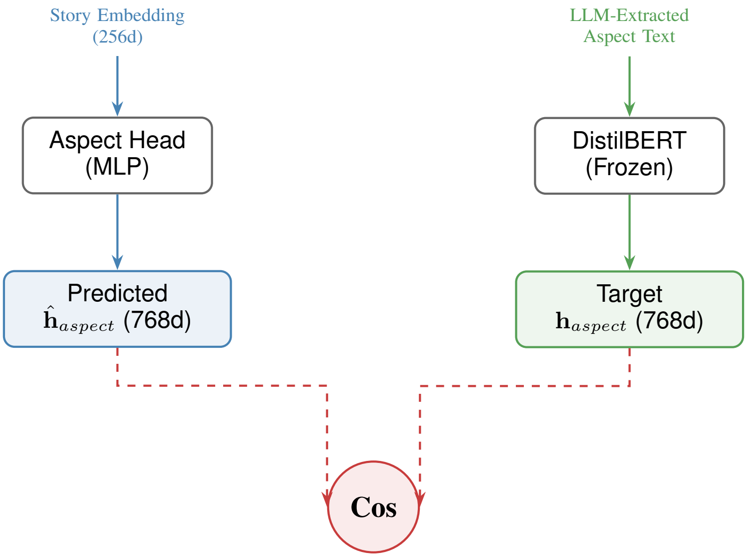
3.5 Inference

During inference for Track 1 compliance, each story is independently encoded through the base encoder and projection layer to produce its final 512-dimensional embedding. Classification is performed by computing cosine similarities between the anchor embedding and each candidate embedding—the candidate with higher similarity is selected. This satisfies Track 1 constraints of independent story encoding using only cosine similarity.

However, since our model also outputs individual aspect representations for each story, we evaluated multiple inference strategies:

1. **Story embedding only**: Using the 512-dimensional projection output
2. **Individual aspect embeddings**: Using theme, action, or outcome representations separately
3. **Concatenated representation**: Combining story embedding with all three aspect embeddings (896 dimensions total)

For model selection during training, we tracked validation accuracy across all five configurations and saved the best-performing checkpoint for each. This approach allowed us to select models that not only learned effective story-level similarity but also developed semantically meaningful aspect-level representations. Since the model produces aspect embeddings independently for each story, all configurations maintain the Track 1 constraint of separate encoding.


Story Embedding (256d)
Aspect Head (MLP)
Predicted $\hat{\mathbf{h}}_{aspect}$ (768d)
LLM-Extracted Aspect Text
DistilBERT (Frozen)
Target $\mathbf{h}_{aspect}$ (768d)
Cos

$$\mathcal{L}_{aspect} = \frac{1}{3}\sum_k (1 - \cos(\hat{\mathbf{h}}_k, \mathbf{h}_k))$$

Figure 3: Aspect loss.

4 Experimental results

4.1 Training Procedure

The model is trained using a combined loss function with three components:

1. **Triplet Loss** ($L_{triplet}$): Encourages the anchor embedding to be closer to the positive story than to the negative story in cosine similarity space. For a batch of size B:

$$L_{triplet} = \frac{1}{B}\sum_{i=1}^{B} \max(0, \cos(\mathbf{a}_i, \mathbf{n}_i) - \cos(\mathbf{a}_i, \mathbf{p}_i) + m_{triplet}) \quad (1)$$

where $\mathbf{a}_i$, $\mathbf{p}_i$, $\mathbf{n}_i$ are the anchor, positive, and negative embeddings respectively, and $m_{triplet}$ is the margin hyperparameter (Schroff et al., 2015).

2. **Aspect Supervision Loss** ($\mathcal{L}_{aspect}$): Cosine similarity loss between predicted aspect embeddings and target aspect embeddings derived from LLM-extracted descriptions (see Figure 3). This loss provides explicit supervision for learning aspect-specific representations:

$$\mathcal{L}_{aspect} = \frac{1}{3B}\sum_{i=1}^{B}\sum_{k\in K}(1-\cos(\mathbf{h}_{i,k}^{pred}, \mathbf{h}_{i,k}^{target})) \quad (2)$$

where $K = \{\text{theme, action, outcome}\}$, $\mathbf{h}_{i,k}^{pred}$ and $\mathbf{h}_{i,k}^{target}$ are the predicted (768d) and target (768d) embeddings for aspect k of story i.

3. **Cross-Attention Loss** (L_{cross}): Applies triplet margin loss at the aspect level using cross-attended representations, encouraging aspect-wise similarity patterns to align with overall narrative similarity:

$$L_{cross} = \frac{1}{3B}\sum_{i=1}^{B}\sum_{k\in\mathcal{K}} \max(0, \cos(\mathbf{a}_i^k, \mathbf{c}_i^{neg,k}) - \cos(\mathbf{a}_i^k, \mathbf{c}_i^{pos,k}) + m_{cross}) \quad (3)$$

where $\mathbf{a}_i^k$ is the predicted aspect k for anchor i, $\mathbf{c}_i^{pos,k}$ and $\mathbf{c}_i^{neg,k}$ are the cross-attended aspect representations for positive and negative candidates, and m_{cross} is the margin.

4. **Total Loss**: The overall loss is

$$L_{total} = L_{triplet} + \gamma \cdot L_{aspect} + \beta \cdot L_{cross} \quad (4)$$

where γ and β are weighting hyperparameters for aspect supervision and cross-attention losses respectively.

4.2 Results

Compared methods. **Baseline** uses only DistilBERT to encode each story (mean pooling) and ranks candidates by cosine similarity. **Our model** uses the same (frozen) DistilBERT backbone with

Model / Representation	Acc.
Baseline: DistilBERT-only	53%
Ours: story_embedding_only	67%
Ours: schema_aspect_only (theme)	63%
Ours: action_aspect_only	**70%**
Ours: outcome_aspect_only	67%
Ours: concat (story + aspects)	**70%**

Table 1: Accuracy on the narrative similarity triplet task.

learned projection and aspect heads; at inference we test: (i) story embedding only, (ii) each aspect embedding alone (theme/schema, action, outcome), and (iii) concatenation of story + all aspects.

Results. Table 1 reports accuracy on the triplet decision task.

Comparative analysis. All variants of our aspect-supervised architecture improve substantially over the DistilBERT-only baseline (+10 to +17 absolute). The best performance (70%) is achieved both by using only the *course-of-action* embedding and by concatenating story+aspects, suggesting that event structure is the strongest single signal in our learned representation, while theme and outcome provide smaller gains when used alone.

5 Conclusion

In this work, we tackled SemEval-2026 Task 4 on narrative story similarity by proposing an aspect-supervised encoder aligned with three narrative dimensions: abstract theme, course of action, and outcomes. Our model uses a frozen DistilBERT backbone with learned projections and aspect-specific heads trained using LLM-extracted aspect descriptions. We also add cross-attention to capture aspect-level interactions during training, while keeping Track 1 compliance at inference by encoding stories independently and comparing them via cosine similarity.

A central challenge throughout the project was the limited availability of high-quality labeled data. We observed that the provided synthetic training set was not representative of the evaluation distribution, leading to an unrealistically high baseline performance on the synthetic data but substantially lower performance on the development set. This mismatch made standard supervised training unreliable and quickly resulted in overfitting, even when the model remained undertrained with respect to the true task distribution. To mitigate this, we augmented the small development set using LLM-based generation and built a training signal through aspect extraction, producing a larger set of triplets annotated at the aspect level. While this strategy enabled training of our aspect-supervised architecture and improved controllability of the training process, the overall performance remained constrained by the small size and limited diversity of the original human-labeled data.

Overall, our experiments suggest that explicit aspect supervision is a promising direction for narrative similarity because it encourages the model to capture not only whether stories are similar but also why they are similar. However, our results also highlight that the effectiveness of this strategy depends strongly on the quantity and diversity of reliable training examples. Future work should focus on obtaining more representative human-labeled triplets, improving augmentation diversity and quality control, and exploring stronger domain adaptation methods to better bridge the gap between synthetic and real evaluation data.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Hans Ole Hatzel, Ekaterina Artemova, Haimo Stiemer, Evelyn Gius, and Chris Biemann. 2026. SemEval-2026 task 4: Narrative story similarity and narrative representation learning. *CoRR*. Shared task description.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. *Preprint*, arXiv:1702.00887.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.