

ChulaNLP at SemEval-2026 Task 4: Neural Aspect Composition for Narrative Story Embeddings

James Michael Gampper and Attapol T. Rutherford*

Department of Linguistics

Faculty of Arts

Chulalongkorn University

6640037922@student.chula.ac.th, attapol.t@chula.ac.th

Abstract

Comparing stories and narratives has proven to be a difficult task to automate because traditional vector representations fail to capture the layered and multi-faceted aspects of stories such as theme, plot progression, and resolution. We address SemEval-2026 Task 4, which requires generating vector embeddings that preserve narrative similarity relationships. We propose Neural Aspect Composition, which functions by using a Large Language Model (LLM) to decompose stories into 13 semantic narrative aspects (theme, course of action, outcomes, etc.), encodes each aspect separately using an encoder model, and learns a global importance weight for each aspect through a trained weighting layer. Our approach achieves the official test scores of 0.64 on Track A and 0.61 on Track B. During validation, it outperformed vectors produced by inputting the raw story text directly into an encoder model and a sentence-averaging baseline. The analysis of the learned weights on the development set reveals that thematic elements and narrative resolutions were the primary drivers of perceived similarity, receiving significantly higher weights than intermediate plot events and other minor details such as character introductions.

1 Introduction

The ability to quantify semantic similarity between narratives is fundamental to NLP applications like recommendation systems and plagiarism detection. However, narratives are hierarchical structures composed of themes, plots, and outcomes, which traditional monolithic embeddings often conflate. SemEval 2026 Task 4 (Hatzel et al., 2026), Narrative Story Similarity and Narrative Representation Learning, is a shared task that addresses this challenge. In this task participants compete on two subtasks (tracks): Track A asks systems to take

*Corresponding author

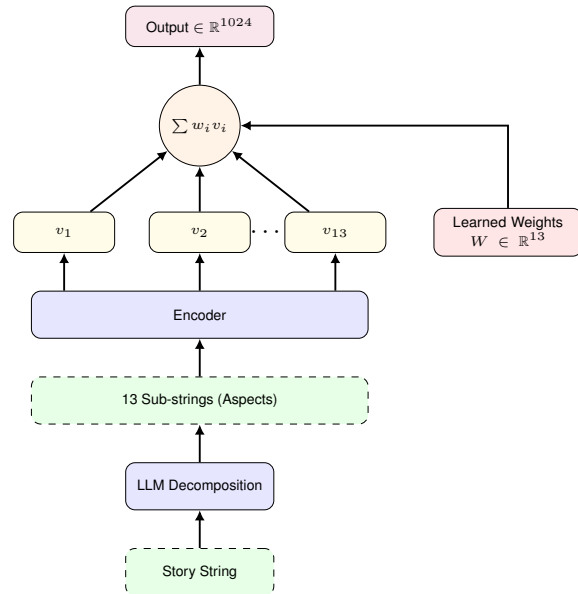


Figure 1: Schematic diagram for Neural Aspect Composition. Aspect vectors are aggregated via learned weighted averaging.

triples of an anchor story and two choices and decide which choice is narratively closer to the anchor, and Track B requires systems to produce vector embeddings for individual stories so that the cosine similarities of those embeddings reflect underlying narrative similarities. The goal is both to directly compare narrative similarity judgments and to learn representations that capture nuanced story-level semantic relationships.

We propose **Neural Aspect Composition**, a modular architecture that explicitly models this hierarchy. Our approach first prompts a large language model to analyze each story and extract thirteen narrative aspects based on (Chun, 2024) (e.g., *Main Theme*, *Climax*, *Moral Lesson*). Specifically, we provide the LLM with the full story text and prompt it to return a brief summary of each aspect as strings. These aspect summaries (text strings) are then independently encoded into dense vectors

and combined via a learnable weighting layer that assigns a scalar importance score to each component. This design allows the model to learn which narrative elements most strongly predict human similarity judgments and weight them accordingly.

We validate our method on SemEval-2026 Task 4, which requires generating embeddings that reflect narrative similarity. Our system outperforms all baselines on the development set. Our system was registered as `james-gamper`. Our final official test scores were 0.64 on Track A and 0.61 on Track B, suggesting that while the approach is highly interpretable, it is prone to overfitting on small datasets. The validation results indicated that a simple weighted combination of high-quality aspect embeddings is more effective than complex projection architectures in data-constrained environments. Furthermore, our analysis of the learned weights provides interpretability, revealing that *Theme* and *Resolution* were the most critical factors for narrative similarity on the development set, while specific plot events like *Inciting Incident* were less significant.

2 Related Works

Chun (2024) introduced AIStorySimilarity, a benchmark for measuring long-text story similarity based on narrative elements. Their work demonstrated that LLM-based extraction of structural elements provides better similarity metrics than whole-text comparisons. We build upon this by formalizing the combination of these elements into a learnable vector space.

Decomposing text into semantic aspects has proven effective for various NLP tasks, from sentiment analysis to structured text representation. Schopf et al. (2023) introduced AspectCSE, demonstrating that aspect-based contrastive learning significantly improves semantic textual similarity tasks by capturing specific dimensions of similarity that generic embeddings miss. Their work highlights that multi-aspect embeddings, which simultaneously consider multiple specific aspects, outperform single-aspect approaches. Our work extends this paradigm to narrative similarity by decomposing stories into theory-grounded aspects (theme, course of action, outcomes) before vectorization, similarly leveraging contrastive learning principles to weigh these aspects effectively.

The use of Large Language Models as automated evaluators has gained traction. Bavaresco

et al. (2024) investigated the reliability of LLMs as judges across 20 NLP tasks, finding that while they can approximate human judgments, their performance varies by task complexity. Our Track A approach directly utilizes this capability, treating the similarity ranking as a zero-shot LLM evaluation task.

3 Our Approach

Partially inspired by (Chun, 2024), we break down each story into 13 specific narrative components (Table 1) before vectorization. Rather than using whole-text embeddings that often lose fine-grained narrative structure, we use a Large Language Model (LLM) to perform this decomposition. Importantly, this "extraction" is actually a generation task. Instead of pulling exact quotes directly from the text, the LLM reads the full story and writes a short, 1-2 sentence summary for each aspect. For example, given a test-set story about a captured Apache warrior attempting to return home, the LLM summarizes the *Protagonist introduction* as: "*Massai is introduced as 'the last Apache warrior,' captured and being transported after Geronimo's surrender.*" For the *Main narrative theme*, it generates: "*The struggle for freedom and a return to one's homeland against overwhelming opposition.*"

We use a strict JSON prompt (provided in Appendix A.1) to capture these summaries. Each of these 13 generated descriptions is then independently converted into its own vector using a dense encoder model, producing 13 separate vectors per story.

Category	Aspect
Theme	Main narrative theme
	Secondary narrative theme
	Main resolution theme
	Secondary resolution theme
Course of Action	Protagonist introduction
	Inciting incident
	Subplots
	Rising action
	Climax
	Falling action
Outcomes	Conflict resolution
	Characters' fates
	Moral lessons

Table 1: The thirteen narrative aspects extracted from each story based on Chun (2024)

We propose **Neural Aspect Composition**

(Method 07) as our primary contribution. This method learns a global importance score for each narrative aspect, allowing the model to emphasize elements that drive human similarity judgments (e.g., themes) while downweighting less relevant details.

We parameterize aspect importance using learnable logits $\alpha \in \mathbb{R}^{13}$ transformed via softmax to ensure valid probability weights (Figure 1):

$$\begin{aligned} w_i(\alpha) &= \frac{\exp(\alpha_i)}{\sum_{j=1}^{13} \exp(\alpha_j)}, \\ \mathbf{e}(s; \alpha) &= \sum_{i=1}^{13} w_i(\alpha) \mathbf{v}_i(s) \end{aligned} \quad (1)$$

where $\mathbf{v}_i(s)$ is the embedding of the i -th aspect of story s . The final embedding $\mathbf{e}(s; \alpha)$ is a weighted sum of aspect vectors.

We optimize α using a margin-based ranking loss:

$$\mathcal{L}(\alpha) = \max(0, m - s(a, p) + s(a, n)) + \lambda \|\alpha\|_2^2 \quad (2)$$

where $s(x, y)$ is the cosine similarity between story embeddings. This formulation allows us to learn the optimal contribution of each narrative aspect from labeled triplet data.

4 Experimental Setup

The competition provided Wikipedia-sourced story summaries. The development set for Track A comprises 200 triplets covering 479 unique stories. We also analyzed 1,900 synthetic triplets provided by the organizers but found them to have a significantly different distribution from the real data ($p < 0.001$), making them unsuitable for training. We randomly partitioned the 200 labeled development triplets into a training set of 160 triplets (80%) and a held-out validation set of 40 triplets (20%). We used stratified sampling to preserve the balanced label distribution.

For aspect extraction, we used DeepSeek-V3.2-Exp (DeepSeek-AI et al., 2024) with temperature=0.3. For vector encoding, we employed the BGE-M3 model (Chen et al., 2024) (BAAI/bge-m3), which produces 1,024-dimensional embeddings. BGE-M3 was selected due to its strong performance on multilingual and long-text embedding benchmarks, which is suitable for capturing high-level semantic aspects.

All trainable methods were trained on the 160-triplet training set and evaluated on the 40-triplet

validation set. Baseline methods without parameters were evaluated on the full set. We report validation accuracy as the primary metric. For Track B, we evaluate embedding quality by computing cosine similarity between story vectors and using these similarities to make triplet decisions on the validation set. For each triplet (anchor, text_a, text_b), we select the alternative with the higher cosine similarity to the anchor.

The baseline for Track A is a zero-shot prompting on LLM. We tried out many prompt variations on LLM DeepSeek-V3.2-Exp, which has 685 billion parameters. The optimal prompt that we identified achieved 75.0% accuracy (Table 2). The key improvements included enforcing concise output format and providing explicit tie-breaking instructions that prioritize thematic similarity. The final prompt used in our official submission is shown below.

```
You are an expert on stories and narratives. Tell us which of two stories is narratively similar to the anchor. Keep answer concise. Output only 'A' or 'B'. If tie, choose story with closest theme match. Output only 'A' or 'B'.
```

```
[Anchor Story]: {anchor_text}
[Story A]: {story_a_text}
[Story B]: {story_b_text}
```

For Track B, to validate our proposed method, we compare it against several baselines and ablated setups:

Whole-Text Baselines (Methods 01-02). We encode the entire story text using standard encoders (all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) and BGE-M3). These serve to establish the performance floor of non-decomposition methods, testing the capability of modern encoders to capture narrative similarity from raw text alone.

Sentence-Level Encoding (Method 03). To isolate the contribution of *semantic* decomposition (via LLM) from *structural* decomposition, we implement a sentence-level ablation. We split stories into sentences using NLTK, encode each sentence independently, and average the resulting vectors. This tests whether simply breaking the story into smaller, manageable chunks is sufficient to improve representation quality, or if the LLM’s ability to extract specific narrative themes is necessary. If this method performs similarly to our aspect-based approach, it would imply that the cost of LLM inference is unnecessary.

Uniform Averaging (Method 04). We average the 13 aspect vectors with equal weights ($w_i = 1/13$). This ablation tests whether the decomposition itself provides value, independent of learned weighting.

Bayesian-Optimized Averaging (Method 05). We optimize the weights w_i using Bayesian Optimization (Snoek et al., 2012) with a Gaussian Process surrogate, treating validation accuracy as a black-box function. This compares our gradient-based learning against a gradient-free alternative.

Siamese Aspect Projection (Method 06). To test if inter-aspect dependencies matter, we use a Siamese architecture (Bromley et al., 1993) where each aspect v_i is projected to a lower dimension via a shared matrix \mathbf{W}_{proj} before concatenation and final projection:

$$\mathbf{e}(s) = \mathbf{W}_{out} \left[\bigoplus_{i=1}^{13} \mathbf{W}_{proj} \mathbf{v}_i(s) \right] \quad (3)$$

This model has 0.9M parameters compared to Neural Aspect Composition’s 13 parameters. It serves as a high-capacity baseline to test if model complexity improves performance.

5 Results and Discussion

For Track A, prompt-based approaches were evaluated separately. A baseline zero-shot prompt achieved 72.5% accuracy, while prompt refinement raised validation accuracy to 75.0%. In contrast, using aspect-based embeddings to perform triplet classification yielded lower performance (approximately 70%). This indicates that direct LLM reasoning over complete text is more effective for discrete comparative judgments than embedding-based similarity decisions.

The weakest performance comes from non-decomposition approaches: all-MiniLM-L6-v2 achieves 55.0% accuracy, while BGE-M3 whole-text encoding improves slightly to 59.0% (Table 3). Sentence-level averaging achieves 57.5% and does not meaningfully outperform whole-text encoding, indicating that simply splitting stories into smaller segments is insufficient to capture narrative similarity. These results establish a strong lower bound and confirm that monolithic encoding struggles with hierarchical narrative structure. Moreover, the choice of encoder seems to be less critical than the representation structure itself, as even strong encoders underperform without explicit narrative decomposition.

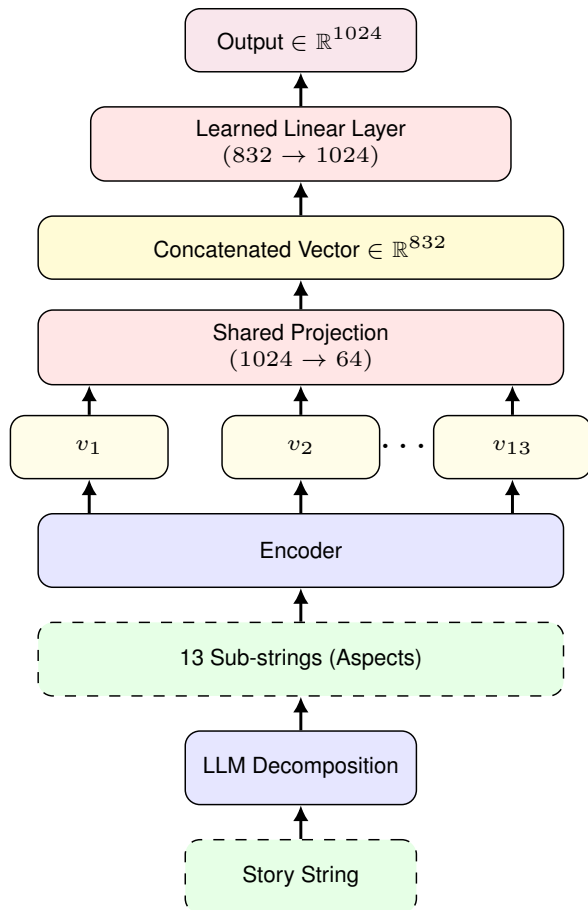


Figure 2: Dataflow for Siamese Aspect Projection (ablation model).

Approach	Accuracy
Baseline prompt	72.5%
Optimal prompt	75.0%
Thirteen narrative aspects	70.05%
Aspect triplet	69.5%

Table 2: Track A classification results. Optimal prompt achieves best performance with minimal engineering.

Introducing LLM-based aspect decomposition yields consistent improvements. Uniform averaging of the 13 extracted aspects (Method 04) increases accuracy to 62.0%, a three-point gain over the strongest whole-text baseline. This demonstrates that structural decomposition alone provides measurable benefit, even without learned weighting. Bayesian-Optimized Averaging (Method 05), which tunes 13 scalar weights via black-box optimization, further improves performance to 67.5%, suggesting that different narrative aspects contribute unequally to similarity judgments.

The highest validation performance is achieved by Neural Aspect Composition (Method 07), which

ID	Method	Params	Acc.
<i>Baselines (No LLM)</i>			
01	all-MiniLM-L6-v2	–	55.0%
02	BGE-M3 whole-text	–	59.0%
03	BGE-M3 sentence avg.	–	57.5%
<i>LLM-Based Aspect Decomposition</i>			
04	Uniform Averaging	0	62.0%
05	Bayesian-Optimized Averaging	13	67.5%
06	Siamese Aspect Projection	0.9M	75.0%
07	Neural Aspect Composition	13	80.0%

Table 3: Track B validation accuracy results. Baseline methods (01–04) evaluated on full 200 triplets. Optimized methods (05–07) trained on 160 triplets and evaluated on 40 held-out validation triplets. Neural Aspect Composition (07) achieves best performance with only 13 parameters.

reaches 80.0% accuracy using only 13 trainable parameters. Compared to Bayesian optimization (67.5%), gradient-based learning provides a substantial 12.5 percentage point improvement, indicating that even in small-data regimes (160 training triplets), supervised optimization of aspect weights can effectively capture signal. Importantly, this gain is achieved with minimal model capacity, reinforcing the hypothesis that carefully constrained structure can outperform both naive averaging and unconstrained models in low-resource settings. The high-capacity Siamese Aspect Projection model (Method 06), with 0.9M parameters, achieves 75.0% validation accuracy. Although it outperforms Bayesian-Optimized Averaging, it does not surpass Neural Aspect Composition. This suggests that increasing representational capacity does not automatically translate to better generalization in data-constrained conditions. The shared projection architecture improves performance relative to naive concatenation, but still appears more susceptible to overfitting than the low-parameter weighted averaging model.

On the official SemEval-2026 test set, our final submissions achieved 0.64 accuracy on Track A and 0.61 on Track B, as shown in Table 4. The drop from 80.0% validation accuracy (Track B) to 0.61 test accuracy highlights significant overfitting to the small development set. While aspect decomposition improves in-domain validation performance, generalization across distribution shifts remains challenging. This discrepancy underscores the difficulty of learning stable narrative representations from limited triplet supervision.

System	Track A	Track B
ChulaNLP	0.64	0.61
Top System (COGNAC)	0.78	0.72
Mean Score	0.67	0.64
Median Score	0.68	0.65

Table 4: Official test set results compared to leaderboard systems.

Aspect	Weight
Theme: Main narrative	19.2%
Outcomes: Moral lessons	19.0%
CoA: Rising action	17.9%
Theme: Resolution main	13.0%
Theme: Secondary narrative	10.8%
CoA: Subplots	8.4%
CoA: Falling action	6.2%
Theme: Resolution secondary	3.0%
CoA: Inciting incident	1.6%
CoA: Protagonist intro	0.2%
CoA: Climax	0.2%
Outcomes: Characters’ fates	0.2%
Outcomes: Conflict resolution	0.2%

Table 5: Learned aspect weights from Neural Aspect Composition (80/20 Split). Thematic elements receive significantly higher weights than specific plot events.

Our results reveal an interesting dichotomy between the two tracks. Track A benefits from direct LLM reasoning over complete text, while Track B requires explicit structural decomposition. This suggests that while LLMs can perform comparative reasoning effectively, dense vector representations benefit from explicit structural decomposition aligned with narrative theory.

The analysis of Neural Aspect Composition’s learned weights reveals which narrative elements contribute most to similarity judgments (Table 5). Thematic elements dominate the learned distribution, with main narrative theme (19.2%) and moral lessons (19.0%) receiving the highest weights. In contrast, specific plot events receive minimal attention—climax (0.2%), protagonist introduction (0.2%), and conflict resolution (0.2%) contribute negligibly. This distribution demonstrates that overarching themes and moral lessons define narrative similarity more effectively than concrete plot points, aligning with narrative theory’s emphasis on thematic coherence over specific story events.

6 Conclusion

In this work, we presented Neural Aspect Composition for SemEval-2026 Task 4, a structured approach to narrative similarity that decomposes sto-

ries into theory-grounded semantic aspects before vectorization. Across seven systematically compared methods, our experiments demonstrate that explicit narrative decomposition consistently outperforms whole-text encoding. Even uniform averaging of aspect embeddings improves over strong encoder baselines, while learning global aspect weights yields the strongest validation performance with only 13 trainable parameters. Our findings suggest that structured semantic decomposition, combined with lightweight parameterization, is a promising direction for story-level representation learning.

References

- Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, and 1 others. 2024. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. *arXiv preprint arXiv:2406.18403*.
- Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *arXiv preprint arXiv:2402.03216*.
- 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.
- DeepSeek-AI and 1 others. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.
- Hans Ole Hatzel, Ekaterina Artemova, Haimo Stiemer, Evelyn Gius, and Chris Biemann. 2026. Semeval-2026 task 4: Narrative story similarity and narrative representation learning. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*, TBD. 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*. Association for Computational Linguistics.
- Tim Schopf, Emanuel Gerber, Malte Ostendorff, and Florian Matthes. 2023. [AspectCSE: Sentence embeddings for aspect-based semantic textual similarity](#)

[using contrastive learning and structured knowledge](#). In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 1054–1065, Varna, Bulgaria. INCOMA Ltd.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, volume 25.

A Prompt Templates

A.1 Track B Aspect Extraction Prompt Mapping

The system prompt string utilized for the generative extraction of narrative aspects is defined below.

```
Analyze this narrative and extract the
following aspects. Provide 1-2
sentence descriptions for each.

Story:
{story_text}

Extract these aspects in JSON format:

{
  "abstract_theme": {
    "main_narrative_theme": "[Primary
    theme/message of the story]",
    "secondary_narrative_theme": "[Secondary
    themes or subthemes]",
    "resolution_main_theme": "[How the main
    theme is resolved]",
    "resolution_secondary_theme": "[How
    secondary themes are resolved]"
  },
  "course_of_action": {
    "protagonist_introduction": "[How
    protagonist is introduced]",
    "inciting_incident": "[Event that triggers
    the main conflict]",
    "subplots": "[Secondary plot threads]",
    "rising_action": "[Escalation of conflict
    and tension]",
    "climax": "[Peak moment of
    tension/conflict]",
    "falling_action": "[Events after climax
    leading to resolution]"
  },
  "outcomes": {
    "conflict_resolution": "[How main conflicts
    are resolved]",
    "characters_fates": "[What happens to main
    characters]",
    "moral_lessons": "[Themes, messages, or
    lessons conveyed]"
  }
}

Respond ONLY with valid JSON, no additional
text.
```