Team UBSE at SemEval-2026 Task 4: Adapting Generalist Embeddings for Narrative Representations

Marius Marogel
University of Bucharest, Romania
marius.marogel@s.unibuc.ro

Marius Popescu
University of Bucharest, Romania
marius.popescu@fmi.unibuc.ro

Abstract

The Narrative Story Similarity and Narrative Representation Learning (NSNRL) task measures the narrative similarity between two stories based on three core aspects: the abstract theme, the course of action, and the outcomes. Our system leverages LLMs both for extracting high-level aspects and to encode them with state-of-the-art generalist embedding models. We then apply a series of embedding post-processing steps and learn to fit the embedding space with a Mahalanobis-like diagonal metric. We show that some of these techniques should not be applied universally, as they do not necessarily increase performance or overfit, depending on the base encoder. Our system outperforms the baseline only in Track B, ranking twelfth out of twenty-seven on the final leaderboard, while performing lower than the baseline accuracy in Track A.

1 Introduction

Textual similarity (Gomaa et al., 2013) is a fundamental concept that appears in core NLP tasks such as Information Retrieval, Text Clustering, or Text Summarization. From knowledge-based methods (Corley and Mihalcea, 2005) to state-of-the-art LLM-based embeddings (Cao, 2024), measuring textual similarity benefited and remained interconnected with the research of meaningful text representations. Although in most formulations, textual similarity focuses primarily on lexical overlap or sentence-level semantic relationships, narrative texts with varied story progression, structure, or twists cannot be compared with word matching techniques or co-occurrence statistics in the same framing as classical sentence-based similarity. The SemEval 2026 Task 4 of Narrative Similarity and Narrative Representation Learning (Hatzel et al., 2026) provides a setup for participants to explore textual similarity and narrative representations with a focus on three core aspects: the abstract theme, the course of action, and the outcome.

We participate in both tracks of this competition, with a primary focus on learning competitive narrative representations (Track B) that we use in both tasks. Our approach takes advantage of the ability of LLMs to extract high-level aspects from stories and encode documents with task-specific instructions for NLP applications. Then, we perform two debiasing techniques shown to improve embedding performance in downstream tasks (Mu et al., 2018). We first remove from each embedding the projection along the mean embedding direction, lowering the mean bias, hoping the cosine similarity becomes more sensitive to actual narrative differences. Furthermore, we remove the contribution of the top-k singular vectors from a given data split as another post-processing method that reduces anisotropy and improves narrative representations. Ultimately, we reshape the embedding space to fit the provided labels by learning a regularized Mahalanobis-like diagonal matrix using non-negative least squares (NNLS).

Although our approach outperforms the baseline model StoryEmb (Hatzel and Biemann, 2024a) on Track B, the track of interest, ranking twelfth out of twenty-seven systems, it fails to reliably select the candidate story most similar on Track A compared to prompting GPT-4o-mini (OpenAI et al., 2024). Even though both post-processing techniques and the learned diagonal scaling improve narrative representations on the development set, we observe the opposite when varying LLM-based embedding models on the test set, leading to overfitting effects and highlighting that the aforementioned techniques should not be applied universally.

2 Background

Task Description The Narrative Story Similarity and Narrative Representation Learning (NSNRL) task encourages participants to create systems that research narrative similarity of stories. The data

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2057–2064

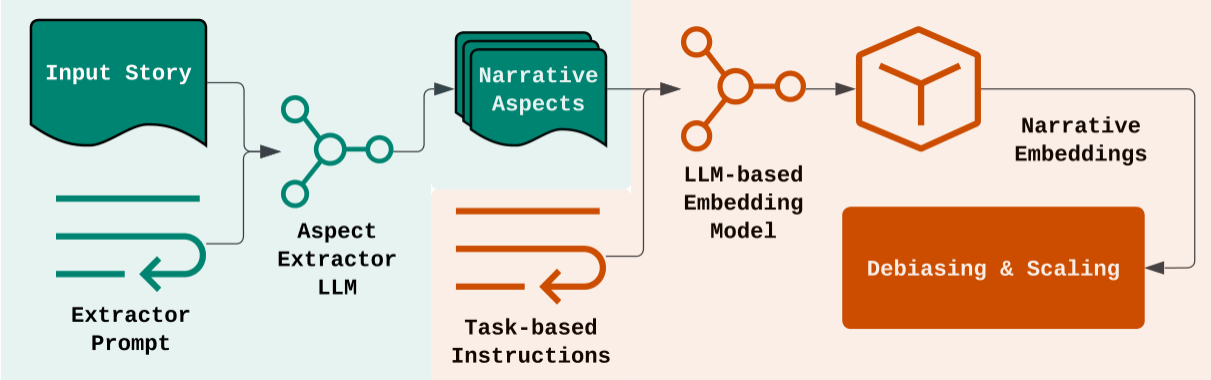

Figure 1: Overview of our narrative representation pipeline. Components are color-coded to reflect functional stages: teal denotes aspect extraction steps and amber indicates embedding generation and post-processing steps.

is comprised of three splits: *sample*, *dev*, and *test*. For Track A, participants have access to triplets in the form of (anchor story, story A, story B) and are tasked with finding which one of the two candidate stories A and B are more narratively similar to the anchor story. In this context, narrative similarity is defined by the following three aspects:

1) Abstract Theme: The underlying motifs of the story.

2) Course of Action: The succession of events that define the story.

3) Outcomes: The final outcome of the story.

In Track B, the participants receive a single story as input, with the requirement to return a single vector representation that would best numerically encode the story in the same definition of narrative similarity.

Narrative Representation Learning Story representations in a narrative setting have recently been studied from both textual and character perspectives. For example, Wilner et al. (2021) use the transformer architecture (Vaswani et al., 2017) for a novel approach to generate narrative event representations. They use the attention mechanism to re-contextualize events across stories. The authors achieve state-of-the-art results on Multiple Choice Narrative Cloze task and perform well on Story Cloze Task (Mostafazadeh et al., 2016). In Lee and Jung (2020), the authors focus on modeling dynamic character interaction networks. They create entire narrative vectors by assuming that the structural interactions among characters reveal narrative content and test their methods on data collected from IMDb and IMSDb. StoryEmb (Hatzel and Biemann, 2024a) presents a novel method in narrative modeling by employing e5-Mistral-7b (Wang et al., 2024), a causal language model based on Mistral-7B (Jiang et al., 2023), in a contrastive training process on Tell-Me-Again dataset (Hatzel and Biemann, 2024b). The authors test the resulting narrative embeddings on various downstream tasks such as Narrative Retrieval and Narrative Understanding outperforming models such as the XXL variant of Sentence-T5 (Ni et al., 2022). Focusing on evaluation rather than narrative representation, Akter and Santu (2023) propose a novel narrative similarity metric called FaNS (Facet-based Narrative Similarity Metric). After collecting data from a third-party news portal, the authors depart from classical semantic similarity inspired by the work of Xie et al. (2008) and incorporate six relevant questions (*Who?*, *When?*, *Where?*, *What?*, *Why?*, and *How?*) by prompting LLMs and weighing the importance of descriptive and entity-specific questions.

3 System overview

We leverage Large Language Models (LLMs) both as controlled language generators and embedding encoders, incorporating post-hoc embedding debiasing techniques and a learned diagonal Mahalanobis metric over triplets. Our approach follows a multi-staged pipeline that ultimately produces a single dense vector representation per story. Figure 1 represents a visualization of our pipeline that produces narrative representations. We use these representations on both tracks of the competition for all anchors and candidates. In Track A, we choose the candidate story with the highest cosine similarity to the anchor representation for prediction, while for Track B we submit the embeddings

directly.

3.1 Aspect Extraction

The first stage of our system consists of a simple prompting system that uses GPT-4o-mini (OpenAI et al., 2024) to extract in natural language the narrative aspects of interest. We provide the definitions of each aspect, as well as the examples differentiating them, as a `system` message over the `chat/completions` API. The detailed prompt we use and an extracted example are shown in Prompt 1 and examples 2, 3, 4 in the Appendix A.

3.2 Aspect Encoding and Weighting

Formally, the entire story t_0 and the extracted aspects t_1, t_2, t_3 are fed through a generalist embedding model $\mathcal{E}$. The final dense vector representation is a linear combination of these embeddings, where the weights w_i are set to the expected value of the signed triple margin, as shown in Equation 1.

$$\begin{aligned} \mathbf{x} &= \sum_{i=0} w_i \mathcal{E}(t_i), \text{ where} \\ w_i &= \mathbb{E}\Big[y\big(\langle \mathcal{E}(q_i), \mathcal{E}(a_i)\rangle - \langle \mathcal{E}(q_i), \mathcal{E}(b_i)\rangle\big)\Big] \end{aligned} \tag{1}$$

The weights w_i quantify how each aspect (or the whole story) aligns, using an embedding model, with the provided Track A labeled triplets. We also consider these values representative of ranking (or weighing) the importance of each aspect in measuring narrative similarity. We show in Tables 7 Appendix A and 8 Appendix A a comprehensive list of all the instructions that we test, including the accuracy metric using only the entire story or the respective aspect. We keep the instructions concise, formulated similarly to the training instructions, and correlated with NLP tasks such as retrieval, clustering, and classification.

3.3 Embedding Postprocessing for Narrative Similarity

As shown by Liang et al. (2025), text embedding models are prone to output representations similar to the non-zero mean embedding of a corpus. Additionally, Arora et al. (2017) argue in favor of removing the top singular vector since embeddings tend to lie in a subspace dominated by common components, thus focusing on task-relevant information. Our approach in mitigating these effects on a single vector $\mathbf{x} \in \mathbb{R}^{1\times D}$ or on the whole set of embeddings $X \in \mathbb{R}^{N\times D}$ when encoding narrative stories is two-pronged. Firstly, we follow the work of Ren et al. (2025) in subtracting the projection from each embedding along the mean direction: $\mathbf{x} = \mathbf{x} - (\mathbf{x}\hat{\mu})\hat{\mu}$ and then normalized with $\mathbf{x}/||\mathbf{x}||_2$. In order to reduce dominant directions in the embedding space that may be non-informative to narrative similarity, we perform singular value decomposition SVD (Golub and Van Loan (2013)) after centering the embeddings ($X = U\Sigma V^\intercal$) and select top-k singular vectors V_k have their contribution removed. The resulting vector $(X - (XV_k)V_k{}^\intercal)$ is similar to how Mu et al. (2018) transform word-level representations, except that we subtract the projection along the mean direction instead of the mean embedding itself and, most importantly, we compute the singular vectors at corpus-level, not vocabulary-level. We choose the best k that maximizes the accuracy score on the *dev* Track A split.

Aspect	Weight
Whole story (w_0)	0.282
Abstract theme (w_1)	0.267
Course of action (w_2)	0.227
Outcome (w_3)	0.228

Table 1: Weights that we obtain by averaging the products between the candidate choice and difference in similarity for Qwen3-Embedding-0.6B.

Aspect	Weight
Whole story (w_0)	0.232
Abstract theme (w_1)	0.397
Course of action (w_2)	0.165
Outcome (w_3)	0.205

Table 2: Weights that we obtain by averaging the products between the candidate choice and difference in similarity for QZhou-Embedding.

3.4 Scaling the Dimensions

After debiasing the embeddings, we learn a diagonal weighting on the dimensions of the embedding set $\mathbf{X}$. This is equivalent to learning a Mahalanobis-like metric, which rescales each embedding dimension according to the triplets provided in the *dev* Track A. If we denote M as the Mahalanobis metric and $\mathbf{q}, \mathbf{x}$ as the embeddings of the anchor and one of the two provided stories, then $d^2(\mathbf{q}, \mathbf{x}) = (\mathbf{q}-\mathbf{x})^\intercal M(\mathbf{q}-\mathbf{x})$. Setting $M = diag(w_1, ..., w_D)$, we obtain $d^2(q_i, x_i) = \sum_{d=1}^{D} w_d(q_{i,d} - x_{i,d})^2$.

For an input embedding triplet $(\mathbf{q}, \mathbf{a}, \mathbf{b})$, we define $d^2(q_i, a_i) - d^2(q_i, b_i) = \mathbf{F_i^\intercal w}$, where $F_{i,d} = (q_{i,d} - b_{i,d})^2 - (q_{i,d} - a_{i,d})^2$. If we denote $t_i = 2y_i - 1 \in \{-1, +1\}$ and add a ridge

regularization term α, then the objective becomes $\min_{\mathbf{w}\geq 0} ||\mathbf{F}\mathbf{w} - \mathbf{t}||_2^2 + \alpha||\mathbf{w}||_2^2$. We then scale the embeddings with $\sqrt{\mathbf{w}}$ and normalize to norm 1.

4 Experimental setup

Dataset We use the *dev* split of Track A to choose hyperparameter values and learn parameters and the *dev* split of Track B to verify the representations resulting from our development approach. Given that the stories in Track B are contained in Track A, we report any generalizability capabilities of our system only on the *test* splits for both tracks. The metric of interest in both tracks is the accuracy in predicting the correct candidate story.

Aspect Extraction The system message we use to extract each narrative aspect is shown in the Prompt 1. We enforce the output of GPT-4o-mini (OpenAI et al., 2024) to a response format with three fields (*abstract theme*, *course of action*, and *outcomes*) with the help of the library Pydantic. We show examples of aspect extraction from conversations in boxes 2, 3, and 4 in the Appendix A.

Generalist Embedding Models and Aspect Weighting Based on the public leaderboard of MTEB (Muennighoff et al., 2023), we choose Qwen3-Embedding-0.6B (Zhang et al., 2025b) and Qzhou-Embedding (Yu et al., 2025) built on the language modeling foundations of Qwen3 (Yang et al., 2025) and Qwen2.5-7B (Qwen et al., 2025), respectively, after preliminary experiments involving additional models such as e5-mistral-7b (Wang et al., 2024), bge-large-en-v1.5 (Li et al., 2024), and stella-en-400M-v5 (Zhang et al., 2025a) using only the entire story without aspects. We choose the instruction for the models from a set of handcrafted prompts that resemble the training format of the instructions of Qwen-based embedding models. Table 7 Appendix A shows the performance in the *dev* split of Track A of various instructions. In Table 8 Appendix A, we present the accuracy on Track A when encoding only one extracted aspect at a time with different instructions. Out of these, the most consistent keywords that ground the embeddings in a NLP task are *Retrieve* and *Cluster*, so we choose a narrative retrieval-like framework for the whole story and narrative similarity clustering for the aspects. Using the expected value of the signed triple margin, the sets of values we use for aspect weighting are shown in Table 1 and Table 2, highlighting that Qwen3 encoder weighs the aspects almost uniformly and lower than the entire story, while QZhou places higher interest on the *abstract theme* to the detriment of the *course of action*.

System	Track A	Track B
QZ	0.59	0.55
QZ+A	0.55	0.61
QZ+A+P	0.67	0.65
QZ+A+P+D	0.64	0.61

Table 3: Results on *test* split using QZhou-Embedding as embedding model.

System	Track A	Track B
Q3	0.62	0.63
Q3+A	0.65	0.64
Q3+A+P	0.65	0.62
Q3+A+P+D	0.66	0.62

Table 4: Results on *test* split using Qwen3-Embedding-0.6B as embedding model.

Embedding Post-Processing and Diagonal Scaling We use the SVD implementation from Pytorch (Paszke et al., 2019) to find the top-k singular vectors. For each embedding model, we return the optimal k from 1 to 10 that maximizes the accuracy on the *dev* split of Track A, finding $k = 2$ and $k = 1$ for Qwen3 and QZhou, respectively. We learn a diagonal $\mathbf{w}$ to scale the dimensions using the NNLS implementation provided by SciPy (Virtanen et al., 2019) and set the regularization parameter $\alpha = 0.1$ by validating on held-out samples and choosing from the set $\{0.01, 0.1, 1.0\}$. After learning $\mathbf{w}$ on the *dev* split of Track A, we apply the scaling on both the *dev* split of Track B and the entire *test* split.

5 Results

Overall, the submitted proposal system ranks twelfth out of twenty-seven teams in Track B, the track of interest, and ranks twenty-sixth out of forty-four in Track A. In order to assess the performance of our methods in both tracks, we perform incremental examinations of each stage of our system. We denote the base models with Q3 and QZ for Qwen3-Embedding-0.6B and QZhou-Embedding respectively, without a string attached when not using instructions when encoding the entire story, with *(A)* when embedding and weigh-

System	Track A	Track B
Q3	0.59	0.63
Q3+A	0.69	0.67
Q3+A+P	0.73	0.71
Q3+A+P+D	0.85	0.78

Table 5: Results on *dev* split using Qwen3-Embedding-0.6B as embedding model.

System	Track A	Track B
QZ	0.54	0.55
QZ+A	0.62	0.59
QZ+A+P	0.66	0.65
QZ+A+P+D	0.73	0.68

Table 6: Results on *dev* split using QZhou-Embedding as embedding model.

ing the aspects, with *(P)* when we apply post-processing debiasing techniques, and with *(D)* for using diagonal scaling learned from a Mahalanobis-like distance matrix.

Encoding the stories with Q3 without instructions obtains 0.59 Accuracy on *dev* Track A, as can be seen in Table 5, lower than any of the experiments conducted in the same settings with instructions incorporated, as in Table 7, thus exemplifying that task-based prompts are necessary for generalist embedding models to produce better downstream representations. We also report that, on the *dev* split of Track A, encoding only the extracted aspects slightly outperforms Q3 without instructions, achieving up to 0.61 accuracy, compared to only 0.59.

Each step of our methodology increases the accuracy on the *dev* split of both tracks, as seen in Tables 5 and 6. With an increase from 0.63 to 0.67 in the *dev* split and from 0.63 to 0.64 in the *test* set on Track B (Table 4), using the weighted embeddings of the extracted aspects as a final Qwen3-based story representation is more appropriate than using QZhou-based embeddings, as they increase the score from 0.55 to 0.59 only in *dev* and to 0.61 in *test*, while decreasing the results in the *test* split of Track A from 0.59 to 0.55.

Post-Processing techniques to debias the resulted embeddings *(P)* considerably improve the accuracy on both tracks and models, with the exception of the *test* split performance for Qwen3 embeddings, where removing the projections on the mean direction and the top 2 singular vectors decrease the performance from 0.64 to 0.62. This behavior is dependent on the model and not on the dataset, as QZhou representations seem to benefit from these post-processing steps uniformly. Although diagonal scaling *(D)* increases the performance in the *dev* split by a considerable margin (10% for Q3 and 5% for QZ), both models fail to generalize in the *test* split, where the diagonal matrix $\mathbf{w}$ applied to representations overfits the embedding space. Note that we leave out the synthetically generated training set when learning the diagonal Mahalanobis-like matrix $\mathbf{w}$ and provide results only on the *dev* and *test* splits. The embeddings of entire stories with the same narrative retrieval instructions achieve 99% accuracy in the *train* split on LLM-generated data. Participants were required to submit up to a maximum of five submissions on the *test* set to discourage heavy hyperparameter tuning, so our final submissions for both tracks were: *Q3+A+P+D*, *Q3+A+P*, *Q3+A*, *QZ+A+P*, *QZ+A*. However, due to a technical issue, the Track A result of *QZ+A+P* of 0.67 was a failed experiment, so 0.65 is the final Track A accuracy. In addition to these official submissions, we computed the accuracies of the rest of the experiments after receiving the labels for the *test* split.

6 Conclusion

Generalist embeddings are a reliable tool for encoding stories for narrative similarity and narrative representation learning tasks. Using the capabilities of an LLM to extract high-level aspects, we encode them and the entire story with empirically validated instructions and linearly combine them applying task-aligned weights. Furthermore, our experiments highlight that post-processing techniques applied to generalist embeddings are not universally applicable, as they may hinder performance on certain models. Additionally, we validate that overfitting occurs when applying a diagonal Mahalanobis-like metric to scale the dimensions of story representations.

For future research, we plan to extend our approach by incorporating attention weights on top of encoding the aspects, allowing us to research and interpret multi-aspect narrative similarity scores. Furthermore, inspired by StoryEmb (Hatzel and Biemann, 2024a), we want to focus on unfreezing the base encoder and creating an end-to-end model for narrative representations with specific contrastive losses per aspect.

Acknowledgments

This research is supported by the project "Romanian Hub for Artificial Intelligence—HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 351416.

References

Mousumi Akter and Shubhra Kanti Karmaker Santu. 2023. Fans: A facet-based narrative similarity metric. *arXiv preprint arXiv:2309.04823*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Hongliu Cao. 2024. Recent advances in text embedding: A comprehensive review of top-performing methods on the mteb benchmark. *Preprint*, arXiv:2406.01607.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan. Association for Computational Linguistics.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

Wael H Gomaa, Aly A Fahmy, and 1 others. 2013. A survey of text similarity approaches. *international journal of Computer Applications*, 68(13):13–18.

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

Hans Ole Hatzel and Chris Biemann. 2024a. Story embeddings—narrative-focused representations of fictional stories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5931–5943.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

O-Joun Lee and Jason J Jung. 2020. Story embedding: Learning distributed representations of stories based on character networks. *Artificial Intelligence*, 281:103235.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *Preprint*, arXiv:2409.15700.

Haoyu Liang, Youran Sun, Yunfeng Cai, Jun Zhu, and Bo Zhang. 2025. Jailbreaking llms' safeguard with universal magic words for text embedding models. *Preprint*, arXiv:2501.18280.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2018. All-but-the-top: Simple and effective post-processing for word representations. *Preprint*, arXiv:1702.01417.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan

Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Xingyu Ren, Youran Sun, and Haoyu Liang. 2025. Correcting mean bias in text embeddings: A refined renormalization with training-free improvements on mmteb. *Preprint*, arXiv:2511.11041.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2019. Scipy 1.0-fundamental algorithms for scientific computing in python. *CoRR*, abs/1907.10121.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Sean Wilner, Daniel Woolridge, and Madeleine Glick. 2021. Narrative embedding: Re-contextualization through attention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1405.

Lexing Xie, Hari Sundaram, and Murray Campbell. 2008. Event mining in multimedia streams. *Proceedings of the IEEE*, 96(4):623–647.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Peng Yu, En Xu, Bin Chen, Haibiao Chen, and Yinfei Xu. 2025. Qzhou-embedding technical report. *Preprint*, arXiv:2508.21632.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025a. Jasper and stella: distillation of sota embedding models. *Preprint*, arXiv:2412.19048.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

A Appendix - Prompts and Instructions

In Box 1, we provide the system prompt that we use in conversation with GPT-4o-mini (OpenAI et al., 2024) to extract narrative aspects, alongside a Pydantic class to enforce a structured output. The examples for each aspect are identical to those provided by the organizers in the annotation guidelines.

Tables 7 and 8 present experiments with different instructions for the entire text or the extracted aspect. The metric used in comparison is the accuracy on the *dev* split of Track A.

Box 1: System message

You are an expert on stories and narratives. You extract from stories the abstract theme, the course of action, and the outcomes for narrative similarity tasks. The attributes you extract will be redirected into a text embedding model for narrative similarity.
We define these three aspects as follows:
- Abstract Theme describes the defining constellation of problems, central ideas, and core motifs of a story. The definition does not cover the concrete setting of a story
- Course of Action describes sequences of events, actions, conflicts, and turning points in a story and the order in which they happen
- Outcomes describe the results of the plot at the end of the text, for example, the conflict resolution, the characters' fates, moral lessons, etc. It does not cover intermediate statuses that change later in the story
Each aspect can take different forms in an actual pair of stories. Below, we list one example for each aspect:
- The general setting of the story, if it strongly influences the events in the story or the events necessitate a specific setting (abstract themes)
- A: On the week-long journey from Europe to the Americas, the crew members get into a heated conflict about the best ration packages
- B: The flight to Mars is long. After several weeks, the astronauts become better friends than ever before, having to share the limited resources
- A and B share some similarities in that the

Instruction	Track A
Retrieve similar stories, based on the abstract theme, course of action, and outcomes.	0.64
Retrieve stories that are narratively similar to the given story	0.67
Retrieve narratively similar stories, based on an anchor story	**0.69**
Retrieve narratively similar text	0.61
Cluster similar narrative stories	0.69
Classify similar narrative stories	0.64

Table 7: Results of experiments encoding the entire story with the shown instructions with Qwen3-Embedding-0.6B. The metric shown on Track A is Accuracy on *dev* split.

Instruction	Theme	Action	Outcome
Retrieve stories with similar {aspect}, based on an anchor story	0.57	0.58	0.6
Retrieve similar {aspects}, based on an anchor {aspect}	0.58	0.6	0.6
Classify narrative {aspects}	0.59	0.6	0.58
Cluster similar narrative {aspects}	0.6	0.6	0.61

Table 8: Results of experiments encoding only the respective aspect with the shown instructions with Qwen3-Embedding-0.6B. The metric shown on Track A is Accuracy on *dev* split.

polar opposite outcomes are both enabled by being cut off from the outside world
- The order of events in the story (course of action)
- A: After the ship capsizes and Alice barely makes it out alive, she starts living life to the fullest
- B: Alice is living life to the fullest until, one day, her ship capsizes. She barely makes it out alive
- A and B are similar in that both tell of a good life and a shipwreck (abstract theme), but they differ in the course of action, and the order is very different
- The outcomes of events (story outcomes)
- A: The man intentionally drops a cup; it breaks
- B: He accidentally swipes the bottle off the table, and it shatters
- A and B are similar in that the events are comparable and lead to similar outcomes

In the following boxes, we exemplify the extracted aspects from the anchor story of data point 0 in the provided *dev* split of Track A.

Box 2: Extracted abstract theme

The struggle for climate justice and the advocacy for future generations in the face of climate change, emphasizing the importance of proactive measures to ensure the stability of economies and the environment.

Box 3: Extracted course of action

The narrative begins in 2025 with the establishment of the Ministry for the Future led by Mary Murphy. She engages with central banks to advocate for the issuance of a complementary currency, the carbon coin, as a response to the threat of climate change on financial stability. Meanwhile, an international geoengineering project is initiated in Antarctica to combat climate change effects, and various innovative climate solutions are explored simultaneously.

Box 4: Extracted outcomes

The Ministry's efforts aim to integrate the rights of future generations into current economic policies, potentially leading to the implementation of the carbon coin as a tangible measure against climate instability. Additionally, the geoengineering project and climate initiatives illustrate the cooperative efforts of countries to mitigate climate change impacts, suggesting a hopeful turn towards global collaboration.