CICL26 at SemEval-2026 Task 4: Narrative Story Similarity and Narrative Representation Learning

Yue Yu and Wanzhao Zhang
Computational Linguistics, University of Tübingen
{y.yu, wanzhao.zhang}@student.uni-tuebingen.de

Abstract

This paper describes our submission to SemEval-2026 Task 4 (Track A) on narrative similarity. The task requires systems to determine which of two candidate stories is more narratively similar to a given anchor story. While large language models (LLMs) demonstrate strong semantic reasoning abilities, their predictions in comparative settings can be sensitive to stochastic decoding and input order. We propose a lightweight inference-time cascade strategy that improves robustness without modifying the underlying model. Our approach combines self-consistency voting to reduce sampling variance, a swap-based symmetry test to mitigate positional bias, and a margin-based deterministic decision rule to resolve disagreements. This design explicitly leverages model uncertainty while maintaining reproducibility and simplicity.

Evaluated on the official CodaBench test set, our system achieves 59% accuracy. Results show that structured inference-time aggregation provides measurable gains over single-call LLM predictions in comparative narrative reasoning tasks.

1 Introduction

Narrative similarity is the task of determining whether two stories share meaningful commonality, which is a central problem in computational narratology (Piper, 2021). The SemEval-2026 Task 4 (Hatzel et al., 2026) adopts a comparative judgment paradigm, which is divided into 2 subtasks:[1]

- Track A: Given an anchor story and two candidates, systems must decide which candidate (story A or story B) is more narratively similar to the anchor.
- Track B: The participants are asked to produce a vector representation for an individual story, which should have a cosine similarity that aligns with the underlying stories' narrative similarities. The submitted representations will be validated by the task organizers.

Unlike surface-level textual similarity, track A requires reasoning about deeper narrative dimensions, which consists of three aspects (Hatzel et al., 2026):

1. **Abstract Theme**: The defining constellation of problems, central ideas, and core motifs of a story. For this task, the setting of the story is **NOT** considered as the abstract theme.
2. **Course of Action**: The sequences of events, actions, conflicts, and turning points in a story and the order of the events.
3. **Outcome**: The results of the plot at the end of the text.

The comparative setup mirrors how humans naturally evaluate stories, which introduces extra challenges for the machine learning model. First of all, the model has to extract information in these three aspects from three stories (anchor, A, and B) then provides an overall evaluation, which can be complicated for them to capture the exact details from the text. Secondly, LLMs are known to exhibit position bias (Wang et al., 2024), which mean that they might prefer the option that appears in a certain position. Lastly, many stories in the given dataset are long and complex, containing confusing factors such as multiple characters and plot twists, which leverage the difficulty of the task.

LLMs demonstrate strong semantic understanding and contextual reasoning abilities. However, their predictions can be sensitive to stochastic sampling and input ordering, particularly in ambiguous cases where multiple interpretations are plausible (Keluskar et al., 2024). In pairwise comparison

[1]The detailed description, dataset, annotation guidelines, and baseline models provided by the organizers can be found at: https://narrative-similarity-task.github.io/

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2456–2460

settings, small variations in prompt phrasing or candidate order may lead to inconsistent outputs, limiting reliability when single-pass inference is used.

To address this issue, we propose a lightweight inference-time cascade strategy that explicitly leverages model stochasticity and mitigates positional bias.[2] Our approach combines: (1) self-consistency voting to reduce sampling variance, (2) a swap-based symmetry test to detect order sensitivity, (3) a margin-based deterministic decision rule to resolve disagreements between forward and swapped predictions. Instead of modifying or retraining the underlying LLM, our method improves robustness purely through structured querying and aggregation.

Our approach provides:

- A simple yet effective cascade inference framework combining self-consistency voting and symmetry testing;
- A margin-based resolution mechanism that uses vote stability as a confidence proxy without relying on model probabilities;
- Empirical evidence that inference-time ensembling improves robustness over single-call LLM predictions in comparative narrative similarity tasks.

2 Background

2.1 Dataset

The dataset for track A provided by the organizers consists of 3 subsets:

- Sample Data: 39 entries with labels.
- Development Data: 200 entries with labels.
- Test Data: 400 entries, the labels were only released after the evaluation phase.

The stories in the dataset were manually written, manually written and modified by LLMs, or sampled.

2.2 Related Work

Narrative Similarity and Computational Narratology. Narrative similarity has been studied within computational narratology, which aims to model thematic structure, event progression, and abstract story representations (Piper, 2021). Recent work emphasizes multi-level narrative analysis, including thematic alignment and outcome structure (Hatzel and Biemann, 2024). SemEval-2026 Task 4 adopts a comparative judgment framework, requiring systems to select the more similar candidate given an anchor. While prior research focuses primarily on narrative representation, less attention has been paid to robustness and stability in pairwise narrative comparison settings.

Large Language Models in Comparative Reasoning. Large language models (LLMs) demonstrate strong zero-shot reasoning capabilities (Brown et al., 2020). However, prior work shows that LLM outputs can be sensitive to prompt design, sampling stochasticity, and input ordering, particularly in ambiguous cases (Keluskar et al., 2024). Pairwise comparison tasks are especially prone to positional bias, where reversing candidate order may change model predictions. Such instability motivates structured inference strategies beyond single-pass decoding.

Self-Consistency and Inference-Time Aggregation. Self-consistency decoding aggregates multiple stochastic reasoning paths to improve reliability (Wang et al., 2023). This approach has been shown to reduce variance in reasoning tasks without modifying model parameters. More broadly, inference-time ensembling and vote-based aggregation provide a lightweight alternative to model retraining for improving prediction robustness. Our work extends these ideas to comparative narrative similarity, combining self-consistency voting with a swap-based symmetry test and margin-based deterministic resolution.

3 Methodology

3.1 Overview

Given a triplet (A, B, T), the task is to predict a binary label $y \in \{0, 1\}$, where $y = 1$ indicates that A is closer to T than B. Instead of relying on a single LLM call, we adopt a lightweight cascade strategy combining self-consistency voting and symmetry testing.

3.2 Self-Consistency Voting

LLM inference is stochastic under non-zero temperature decoding. For each input, we perform $N = 3$ independent forward calls:

[2] Our code available at: https://github.com/cicl-iscl/SemEval26_Task4_Narrative-Similarity

$$\mathcal{V}_{\text{fwd}} = \{v_1, v_2, v_3\}, \quad v_i \in \{0, 1\}.$$

The forward majority prediction is

$$\hat{y}_{\text{fwd}} = \text{majority}(\mathcal{V}_{\text{fwd}}).$$

We define a vote margin as

$$m_{\text{fwd}} = \left| \frac{\#(1) - \#(0)}{N} \right|,$$

which serves as a proxy for decision stability.

3.3 Swap Test

To mitigate positional bias, we repeat the same procedure with swapped inputs:

$$s_i = f_\theta(B, A, T).$$

Since swapping inverts semantics, predictions are mapped back via

$$\tilde{s}_i = 1 - s_i.$$

We compute

$$\hat{y}_{\text{swap}} = \text{majority}(\{\tilde{s}_1, \tilde{s}_2, \tilde{s}_3\}),$$

and its corresponding margin m_{swap}.

3.4 Cascade Decision Rule

Final prediction follows a deterministic cascade:

1. If $\hat{y}_{\text{fwd}} = \hat{y}_{\text{swap}}$, output the consensus.
2. Otherwise, select the prediction with larger margin.
3. If margins are equal, perform an additional $N = 3$ forward votes and decide by majority over the aggregated forward votes.

3.5 Output Format

Predictions are written in JSON Lines format with one key per instance:

```
{"text_a_is_closer": true/false}
```

The output order strictly matches the input order.

4 Experiment Setup

Dataset We evaluated our method in the official SemEval-2026 Task 4 Track A test data. Each instance consists of an anchor story and two candidate stories (A and B), and the goal is to determine which candidate is narratively closer to the anchor. Before evaluating our method on the test data, we have developed our method with the development data.

Model Configuration We used GPT-4.1-mini as the underlying large language model for our approach. All experiments were conducted via API calls with a temperature of 0.3 to enable limited stochasticity for self-consistency voting. The maximum number of retries per call was set to 5.

Inference Strategy For each instance, we performed self-consistency voting by querying the model three times under the original input order (Anchor–A–B). We then performed an additional three queries under the swapped order (Anchor–B–A) to test permutation stability. If the majority predictions under both orders agree, the result was returned directly. If they disagree, we apply a margin-based cascade rule, selecting the prediction with the larger majority margin. In the rare case of equal margins, an additional voting round with higher temperature (0.4) was conducted as a tie-breaker.

Evaluation Performance is measured using classification accuracy on the official evaluation server. No training or fine-tuning is performed; all experiments rely solely on inference-time ensembling.

5 Results

Our method was evaluated in the official Track A test set using the CodaBench evaluation server. Our final submission achieves an accuracy of **59%**.

The result reflects the effectiveness of inference-time ensembling. Despite using a single underlying LLM, the combination of self-consistency voting and swap-based symmetry testing improves robustness compared to single-call inference.

Swap Consistency Analysis To further examine the stability of our cascade strategy, we analyzed the agreement between forward predictions (Anchor–A–B) and swapped predictions (Anchor–B–A). The consistency matrix (Figure 1) shows that disagreement occurs in a non-trivial portion of instances, highlighting the presence of positional sensitivity even under majority voting.

We additionally investigate whether internal vote confidence correlates with robustness to input permutation. Figure 2 plots the swap inconsistency rate against the original vote margin. Interestingly, even cases with maximal internal agreement (3–0 majority) exhibit substantial inconsistency after swapping, suggesting that internal self-consistency does not guarantee permutation robustness. This observation indicates that voting primarily reduces

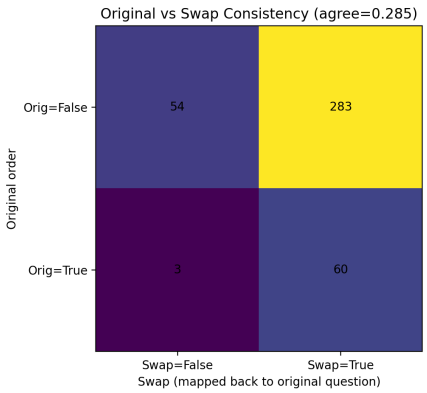

Figure 1: Consistency between forward (Anchor–A–B) and swapped (Anchor–B–A) majority predictions. Diagonal entries indicate agreement.

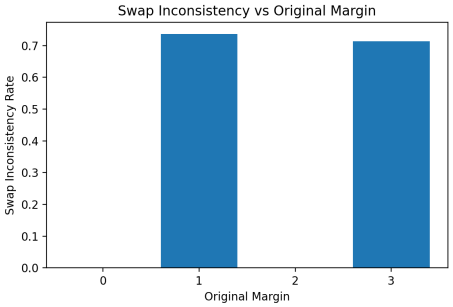

Figure 2: Swap inconsistency rate as a function of original vote margin. Even high-margin cases remain sensitive to input permutation.

sampling noise, while swap testing specifically addresses positional bias. The margin-based cascade rule helps stabilize conflicting cases without increasing model complexity.

Ablation Study LLMs are known to exhibit position bias (Wang et al., 2024), which indicates that the position of the text shown in the triples may affect the judgments of the model. Therefore, to test how much position bias might affect the performance of the model, besides our major experiments, we performed an ablation study with GPT-3.5-turbo using zero-shot naive prompting on the development set. The model was prompted to judge if A or B is closer to the anchor directly in two runs, with the presenting positions of A and B were swapped in each run. The results show that the overall accuracy of the predictions is around 50% (with sequence A-B 52% and B-A 50%), and 46% judgment inconsistency. By inspecting the output, we observed that the model preferred story B when the stories were displayed in A-B sequence, and the other way around when we switched the display order. Therefore, combining with the results from our main experiments, we conclude that GPT models are prone to position bias, as they showed strong inconsistency in their output.

6 Conclusion

The results indicate that judging narrative similarities between stories is still challenging for a state-of-the-art large language model. Our cascade design improves robustness without changing the base model: Self-consistency voting reduces sampling variance, swap test mitigating order sensitivity, and margin-based selection provides a simple confidence-aware resolution mechanism. Our approach remains fully deterministic given fixed sampling settings. However, despite our attempts to reduce the inconsistency of LLM judgments on narrative similarity, there remains substantial room for improvement. A possible direction for future research is to develop an annotation system to capture certain semantic aspects of the stories.

Limitations

Our method depends on commercial LLM APIs, which may affect reproducibility and long-term stability. The cascade strategy also increases inference cost, requiring multiple forward passes per instance. Although swap-based testing reduces positional bias, it does not fundamentally eliminate order sensitivity. Moreover, the approach is designed for symmetric pairwise comparison tasks and may not directly generalize to other narrative reasoning settings.

Acknowledgments

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33.

Hans Hatzel and Chris Biemann. 2024. Modeling narrative similarity across levels of abstraction. In *Proceedings of *SEM*.

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.

Aniket Keluskar and 1 others. 2024. Ambiguity and instability in large language model reasoning. In *Proceedings of EMNLP*.

Andrew Piper. 2021. *Enumerations: Data and Literary Study*. University of Chicago Press.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, and 1 others. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

A Appendix:

A.1 System prompt of our approach with cascade strategy

""" You are an expert judge for narrative similarity in SemEval Track A.

Goal: Decide which story (A or B) is narratively closer to the Anchor.

Narrative similarity is based ONLY on core story structure: 1) Outcomes (most important) 2) Course of Action 3) Abstract Theme

Rules: - Make a forced choice: even if neither is perfect, choose the closer one. - Ignore writing style, names, length, time period, and specific locations/setting. - Theme alone is weak evidence; prefer matching outcomes and event progression.

Output ONLY valid JSON:

```
{"text_a_is_closer": true/false}
```

No explanation. No extra text. No code fences. """