CascadeMind at SemEval-2026 Task 4: A Hybrid Neuro-Symbolic Cascade for Narrative Similarity

Sebastien Kawada[†,‡]
[†] Kaons K^*
sebastien@kaons.com

Dylan Holyoak[‡]
[‡] Epoch Learn ϕ_0
dylan@epochlearn.com

Abstract

Across self-consistency samples from an LLM, vote agreement tracks instance difficulty: on SemEval-2026 Task 4 (Narrative Story Similarity), supermajority cases ($\geq 7/8$ votes) resolve at 85% accuracy, split votes at 67%, and perfect ties at 61%, a monotone gradient that holds across the development set. We exploit this in CascadeMind, which routes eight Gemini 2.5 Flash votes by consensus, escalates split votes to additional sampling rounds, and falls through to a symbolic ensemble of theory-inspired narrative signals only on perfect ties (5% of cases). The system reached 72.75% on Track A test, placing **10th of 44 teams**. Ablations show that the symbolic component contributes negligibly end-to-end and that nearly all gains come from confidence-aware routing. The takeaway is methodological: for narrative similarity, calibrating when to spend more compute on a hard instance matters more than adding auxiliary representations to reason about it. Code is available at https://github.com/chreia/CascadeMind-ACL.

1 Introduction

Comparing two stories for similarity goes beyond surface wording because it requires judging shared theme, sequence of events, and outcome (Piper et al., 2021; Tversky, 1977). SemEval-2026 Task 4 frames this as comparative judgment. Given an anchor and two candidate stories, systems decide which candidate is more similar to the anchor along three axes: abstract theme, course of action, and outcomes (Hatzel et al., 2026).

LLMs handle most cases well but degrade on ambiguous comparisons (Keluskar et al., 2024), making it important to know when to trust a single answer. CascadeMind treats vote agreement across self-consistency samples as a confidence signal and routes accordingly. Confident cases resolve fast, uncertain cases escalate, and perfect ties fall back to a symbolic ensemble. The system (1) samples eight self-consistency votes (Wang et al., 2023) and treats vote distribution as an uncertainty signal, (2) commits when at least 7 of 8 votes agree, (3) escalates split votes to 32 votes total, and (4) falls back to a symbolic ensemble grounded in narrative theory only on perfect ties.

Our contributions are:

- A cascade in which LLM vote agreement determines decision pathway, with pathway-level accuracy monotone in vote consensus (85% / 67% / 61%)
- A symbolic ensemble of five theory-inspired similarity signals (lexical, story-grammar, semantic embedding, tension curve, event chain) used as a fallback only on perfect ties
- Ablations isolating the source of the gain, showing that confidence-aware routing accounts for nearly all of it while the symbolic fallback contributes negligibly because only 5% of cases reach it

2 Related Work

Self-Consistency Decoding Self-consistency belongs to the ensemble tradition, aggregating multiple hypotheses to improve a prediction (Hansen and Salamon, 1990). Wang et al. (2023) introduced self-consistency as a decoding strategy that samples multiple reasoning paths and selects the most consistent answer through majority voting, with diverse reasoning traces yielding higher accuracy than single samples. Subsequent work treats sample consistency as a black-box uncertainty signal (Xiong et al., 2024). Extensions cover medical question answering (Maharjan et al., 2024) and code generation (Huang et al., 2024a). Mirror-consistency addresses overconfidence in minority responses (Huang et al., 2024b).

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1575–1581

Narrative Representation Learning Computational approaches to narrative understanding draw on structural narrative theory as well as neural embedding representations. Hatzel and Biemann (2024) trained story embeddings on reformulations of the same story, making their setup the closest existing work on fictional-narrative similarity. The CoRRPUS framework (Dong et al., 2023) showed that structured code-based representations can improve story understanding in a neurosymbolic setting.

Ensemble Methods and Selective Prediction Dietterich (2000) characterized ensembles as weighted voters over component classifiers. Selective prediction allows models to abstain when uncertain, trading coverage for accuracy (Geifman and El-Yaniv, 2017). Our cascade borrows the abstention trigger from selective prediction but always returns a prediction. Uncertainty selects pathway, not abstention.

Narrative Theory Our symbolic component draws on classical narrative theory. Propp's morphology of the folktale (Propp, 1968) identified recurring structural elements across stories. Freytag's pyramid models narrative tension through exposition, rising action, climax, falling action, and resolution (Freytag, 1863). Todorov's narrative grammar (Todorov, 1969) analyzes structured transformations in narrative. We operationalize these theories as computable similarity signals.

3 System Architecture

CascadeMind uses a direct comparative prompt asking which candidate story (A or B) is more similar to the anchor on abstract theme, course of action, and outcomes. The model returns JSON with a single `decision` field. No chain-of-thought rationale is requested or used.

The four-stage pipeline lets neural voting handle most cases while a symbolic fallback engages only when voting fails, as shown in Figure 1.

3.1 Neural Self-Consistency Voting

We use Gemini 2.5 Flash (Google, 2025a) with `candidateCount=8` to return eight responses per call through the Gemini API (Google, 2025b). Each response is a single A/B vote.

Decision Logic Let $V = \{v_1, \ldots, v_8\}$ be the set of votes where $v_i \in \{A, B\}$. We define the vote count $c_X = |\{v_i : v_i = X\}|$ for $X \in \{A, B\}$.

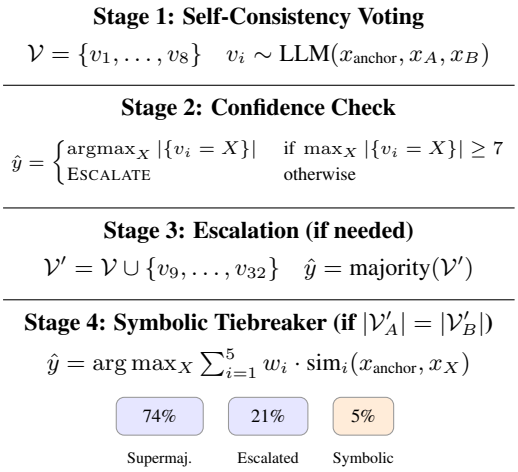

Figure 1: Cascade decision process. Most cases (74%) resolve via supermajority, 21% require escalation, and 5% invoke the symbolic ensemble.

Supermajority: If $\max(c_A, c_B) \geq 7$, we return the majority decision immediately.

Escalation: On splits (4-4, 5-3, or 6-2), we issue three additional `candidateCount=8` calls for 32 votes total and take the majority. A perfect 16-16 tie triggers the symbolic fallback.

3.2 Multi-Scale Narrative Analysis Ensemble

On perfect ties after escalation, the system falls back to a symbolic ensemble of five similarity signals at different levels of abstraction (Figure 2). Let s_i^A and s_i^B denote the similarity scores between the anchor and stories A and B for signal i.

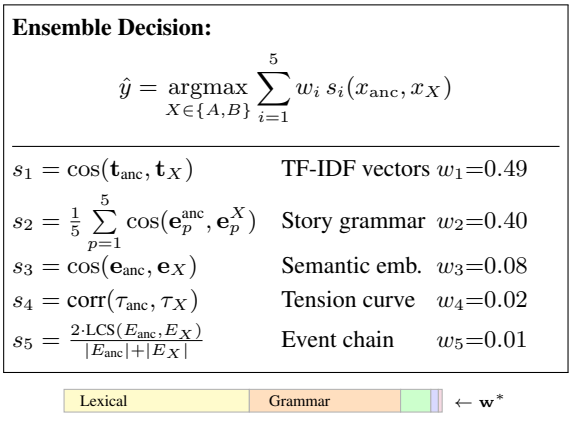

Figure 2: Multi-Scale Narrative Ensemble with five weighted similarity signals ($\mathbf{w}^*$ optimized via differential evolution).

Signal 1: Lexical Similarity (TF-IDF) We TF-IDF-vectorize each story (Salton and Buckley, 1988) and compute cosine similarity between the

anchor and each candidate:

$$s^X_{\text{lex}} = \cos(\text{tfidf}(\text{anchor}), \text{tfidf}(X)) \quad (1)$$

This captures surface lexical overlap, including shared characters and domain terminology.

Signal 2: Story Grammar Similarity As a heuristic inspired by story-grammar and narrative-phase models (Propp, 1968; Freytag, 1863; Thorndyke, 1977), we segment each story into five narrative phases based on position: setting (first 20%), conflict (20-40%), rising action (40-60%), climax (60-80%), and resolution (80-100%). We compute a sentence-transformer embedding for each aligned phase and average cosine similarities across the five phases:

$$s^X_{\text{grammar}} = \frac{1}{5} \sum_{p \in P} \cos(\text{enc}(a_p), \text{enc}(x_p)) \quad (2)$$

where P = {setting, conflict, rising, climax, resolution}.

Signal 3: Semantic Similarity We encode each full story with all-MiniLM-L6-v2 (Reimers and Gurevych, 2019; Sentence Transformers, 2020) and compute cosine similarity:

$$s^X_{\text{sem}} = \cos(\text{enc}(\text{anchor}), \text{enc}(X)) \quad (3)$$

This complements the phase-aligned signal with whole-story semantics.

Signal 4: Narrative Tension Curve Inspired by Freytag's pyramid (Freytag, 1863) and computational work on sentiment-derived story arcs (Reagan et al., 2016), we use a heuristic positional tension proxy. For each sentence, we compute a tension score as the sum of sentiment intensity (absolute polarity) and subjectivity using TextBlob (Loria, 2026). Per-sentence tension is linearly interpolated to a 10-point curve, and we then compute Pearson correlation between curves:

$$s^X_{\text{tension}} = \text{corr}(T_{\text{anchor}}, T_X) \quad (4)$$

This procedure captures similarity in emotional dynamics and pacing.

Signal 5: Event Chain Similarity We extract action verbs using spaCy's part-of-speech tagger (Explosion AI, 2026), filtering to verbs matching a list of 47 narrative action words inspired by event-chain modeling and Propp's character functions (Chambers and Jurafsky, 2008; Propp, 1968) (e.g., *discover*, *fight*, *escape*, *transform*). These form event sequences E_{anc} and E_X (ordered verb lemmas). We compute the longest common subsequence (LCS):

$$s^X_{\text{event}} = \frac{2 \cdot \text{LCS}(E_{\text{anc}}, E_X)}{|E_{\text{anc}}| + |E_X|} \quad (5)$$

This procedure approximates plot-structure similarity using ordered action-verb overlap.

Weighted Ensemble The final score is a weighted sum:

$$\text{score}^X = \sum_{i=1}^{5} w_i \cdot s^X_i \quad (6)$$

The ensemble returns A if $\text{score}^A > \text{score}^B$ and B otherwise, with exact ties defaulting to B in our implementation.

We fit w_i via differential evolution (Storn and Price, 1997) on the organizers' 1,900-triplet synthetic split (SemEval-2026 Task 4 Organizers, 2026b). This split is LLM-generated and is used only for calibrating symbolic weights. Development and test reporting in this paper use the task's human-labeled splits.

The optimization objective minimizes classification error:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} \mathbf{1}\left[\text{sign}\left(\textstyle\sum_i w_i \Delta s^j_i\right) \neq y_j\right] \quad (7)$$

where $\Delta s^j_i = s^A_i - s^B_i$ and $y_j \in \{-1, +1\}$ is the ground truth label. We use differential evolution due to its effectiveness on non-convex, non-differentiable objectives.

The optimized weights are shown in Figure 2. Lexical (49%) and story-grammar (40%) signals dominate, suggesting the synthetic training data discriminates on surface wording and phase structure rather than deeper semantics. The ensemble achieves 99.5% accuracy on a held-out synthetic validation split (n=200).

4 Experiments

4.1 Dataset

The SemEval-2026 Task 4 dataset and evaluation protocol are described by the task organizers (Hatzel et al., 2026; SemEval-2026 Task 4 Or-

System	Track	Test Accuracy	Rank
CascadeMind	A	72.75%	**10th/44**

Table 1: Official shared-task result for our final submission. Rank is among the 44 Track A teams.

Gold Label	Pred A	Pred B
A closer (n=208)	143	65
B closer (n=192)	43	149

Table 2: Post-hoc confusion matrix on released Track A test labels (n=400).

System	Accuracy	Calls/Case
Single Vote	68.0%	1.0
Self-Consistency (k=8)	76.5%	1.0
Majority (k=3 calls)	78.0%	3.0
CascadeMind	81.0%	1.78 avg
+ Symbolic Tiebreaker	81.0%	1.78 avg

Table 3: Development-set system comparison (n=200 for baselines, n=100 for cascade experiments).

ganizers, 2026b).[1] All story summaries and labels used in this work are in English. The task uses Wikipedia plot synopses from the English portion of Tell-Me-Again, filtered to short summaries (typically four to eight sentences). We use the 200-triplet Track A development split for analysis and the 400-triplet test split for the official submission. Each triplet is one anchor with two candidates and a label naming the closer candidate. The synthetic training set comprises 1,900 LLM-generated triplets and is used only for symbolic-weight calibration.

4.2 Implementation Details

We use Gemini 2.5 Flash via the Google AI API with `candidateCount=8` for multi-candidate voting. Temperature is 1.0 to encourage diversity, and other parameters use defaults. The symbolic ensemble module uses scikit-learn for TF-IDF vectorization (Pedregosa et al., 2011), sentence-transformers (all-MiniLM-L6-v2) for embeddings, and TextBlob for sentiment analysis.

4.3 Results

4.3.1 Official Shared-Task Result

Table 1 reports our official result (SemEval-2026 Task 4 Organizers, 2026c). The task received 71 submissions from 46 teams across both tracks, and Track A had 44 ranked teams.

4.3.2 Post-Hoc Test-Set Analysis

After release of Track A test labels (SemEval-2026 Task 4 Organizers, 2026a), we computed post-hoc test-set diagnostics for CascadeMind on all 400 examples. The official leaderboard score remains the submitted 72.75% (291/400) in Table 1. The post-hoc diagnostic predictions score 73.0% accuracy (292/400), a one-prediction difference.

Class-wise behavior is asymmetric. For the *A-closer* class, precision/recall/F1 are 76.9%/68.8%/72.6%. For *B-closer*, they are 69.6%/77.6%/73.4%. Macro-F1 is 73.0% and balanced accuracy is 73.2%. The model predicts B more often than A (53.5% vs. 46.5%), while the released label distribution is slightly A-leaning (52.0% A, 48.0% B), indicating a mild tendency to over-select B in uncertain comparisons.

4.3.3 Development-Set Analysis

Table 3 presents development diagnostics with different denominators. Baselines are computed on the full development split (n=200), while cascade routing diagnostics are computed on a separate subset (n=100). CascadeMind reaches 81.0% on the cascade subset. Because these denominators differ, cross-block percentage-point differences are descriptive only. With 74% resolved at supermajority and 26% escalated to three additional calls, expected API usage is 1.78 calls per case.

All reported LLM decisions are direct A/B outputs. No chain-of-thought rationales are requested, exposed, or used.

4.3.4 Performance by Decision Pathway

Table 4 breaks down accuracy by pathway. Supermajority cases (74% of the development subset) reach 85%, supporting vote consensus as a confidence signal. Escalated cases resolve at 67% under majority, and a separate perfect-tie diagnostic set reaches 61% after symbolic processing. Lower vote consensus tracks lower accuracy, consistent with multi-sample consistency as an uncertainty signal (Xiong et al., 2024).

4.3.5 Symbolic Tiebreaker Analysis

On the separate perfect-tie diagnostic set (n=18), the symbolic tiebreaker achieves 61.1% accuracy (11/18), as shown in Table 4. Manual inspection shows recurring errors involving misleading lexi-

[1] Shared-task data and generated submission artifacts should be retrieved from the task source unless redistribution permission is confirmed.

Pathway	Cases	Accuracy
Supermajority ($\geq$7/8)	74/100	85%
Escalated (majority)	21/100	67%
Escalated (symbolic tie)	5/100	61% (11/18)
Total	*100/100*	*81%*

Table 4: Performance breakdown by decision pathway. Routing shares are measured on the cascade diagnostic subset (n=100). Symbolic-tie accuracy is measured on a separate perfect-tie diagnostic set (n=18).

Split	n	Acc.	F1	A/B
Dev	200	57.0%	57.0	99/101
Test	400	60.5%	60.4	208/192

Table 5: Post-hoc symbolic-only support check using the fixed paper weights and no LLM calls.

cal overlap, non-standard narrative structures that violate positional assumptions, and edge cases that require world knowledge beyond surface signals. Applied to all cases regardless of neural confidence, the symbolic ensemble drops to 53% on the cascade diagnostic subset, making it suitable only as a high-uncertainty fallback.

Table 5 confirms above-chance behavior of the symbolic module as a standalone classifier, scoring 57.0% on development (114/200) and 60.5% on test (242/400). Prediction rates are balanced on both splits, but absolute accuracy stays below the neural cascade. Voting and escalation, not symbolic signals, drive performance.

5 Discussion

Test-Time Behavior Table 1 summarizes official shared-task standing, and Table 2 shows class-level behavior for the post-hoc Track A predictions. The dominant test-time error is predicting B when A is correct (65 cases), which is consistent with lower recall on the A-closer class than on the B-closer class.

Value of Cascade Design On the development set, the cascade's gains come from voting and escalation, not the symbolic tiebreaker. The supermajority rule identifies confident predictions cheaply (85% on 74% of cases), and escalation adds signal for borderline cases. LLM vote distribution serves as a useful uncertainty indicator in this setting because higher consensus correlates with higher accuracy, consistent with broader evidence on sample consistency as black-box LLM uncertainty (Xiong et al., 2024).

Domain Mismatch The symbolic ensemble achieves 99.5% accuracy on a held-out synthetic validation split but only 61.1% on the perfect-tie diagnostic set, and overall system performance drops from 81.0% on the cascade diagnostic subset to 72.75% on official test. This gap is consistent with distribution mismatch, overfitting to synthetic calibration data, or greater difficulty in the final shared-task setting.

Signal Complementarity In this symbolic ensemble trained on synthetic data, weights concentrate on lexical (49%) and story-grammar (40%) signals over semantic embeddings (8%). The 8% semantic weight is small but nonzero, capturing information not covered by the lexical and structural signals.

Measure Limitations The 1% weight on event-chain similarity likely reflects sparsity from exact verb matching against a 47-word list. Richer event representations—fuzzy matching, full narrative event chains (Chambers and Jurafsky, 2008), or semantic role labeling—are a natural next step.

6 Conclusion

We presented a hybrid neuro-symbolic cascade model for narrative story similarity that combines neural self-consistency-style voting with a Multi-Scale Narrative Analysis Ensemble. In official shared-task evaluation, CascadeMind reaches 72.75% Track A test accuracy (Table 1). On development diagnostics (cascade subset, n=100), it reaches 81.0%.

The main contributions are threefold. First, CascadeMind shows that LLM vote distribution can act as a useful uncertainty indicator, with supermajority predictions reaching 85% accuracy on development. Second, ablations isolate where the gain comes from, showing that confidence-aware routing accounts for nearly all of it while the symbolic tiebreaker contributes negligibly because only 5% of cases reach it. Third, the paper documents a development-to-test gap (81.0% to 72.75%) in shared-task conditions.

Limitations

The system depends on a commercial API (Gemini 2.5 Flash) and stochastic decoding. While the routing policy and thresholds are fixed, exact vote distributions can vary across runs and model revisions, which limits strict reproducibility without

pinned model snapshots (Pineau et al., 2021; Chen et al., 2024).

The development-to-test gap is substantial. Accuracy is 81.0% on the cascade diagnostic subset (n=100) versus 72.75% (291/400) in official shared-task evaluation. Post-hoc Track A diagnostics score 73.0% (292/400), a one-prediction difference from the submitted file.

The symbolic tiebreaker has narrow end-to-end impact because only 5% of the cascade diagnostic subset reaches the final tie state. Most gains come from neural voting and escalation rather than symbolic resolution, so improvements to confidence calibration and escalation strategy are likely to matter more than additional symbolic feature engineering.

Post-hoc test analysis shows asymmetric class behavior. Recall is lower for A-closer cases (68.8%) than for B-closer cases (77.6%), and the model predicts B slightly more often than the label distribution warrants. Reducing this decision bias is a direct target for future versions.

Ethics Statement

This work uses public story summaries and organizer-provided labels. We did not recruit, interact with, or collect data from human subjects. The LLM component may inherit biases and other risks present in its training data and commercial LLM deployment settings (Weidinger et al., 2022).

Acknowledgements

We thank the SemEval-2026 Task 4 organizers for creating the benchmark and running the shared task.

References

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. How is ChatGPT's behavior changing over time? *Harvard Data Science Review*.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg.

Yijiang River Dong, Lara J. Martin, and Chris Callison-Burch. 2023. CoRRPUS: Code-based structured prompting for neurosymbolic story understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13152–13168, Toronto, Canada. Association for Computational Linguistics.

Explosion AI. 2026. spaCy: Industrial-Strength Natural Language Processing in Python. spaCy documentation; accessed 2026-05-01.

Gustav Freytag. 1863. *Die Technik des Dramas*. S. Hirzel, Leipzig. Later editions and translations popularized the dramatic pyramid.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Google. 2025a. Gemini 2.5 Flash. Google AI for Developers model documentation; model ID `gemini-2.5-flash`; accessed 2026-05-01.

Google. 2025b. Gemini API documentation. Google AI for Developers API documentation; accessed 2026-05-01.

Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Hans Ole Hatzel, Ekaterina Artemova, Haimo Paul 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)*, San Diego, CA, USA.

Hans Ole Hatzel and Chris Biemann. 2024. Story embeddings – narrative-focused representations of fictional stories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5931–5943, Miami, Florida, USA. Association for Computational Linguistics.

Baizhou Huang, Shuai Lu, Xiaojun Wan, and Nan Duan. 2024a. Enhancing large language models in coding through multi-perspective self-consistency. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1429–1450, Bangkok, Thailand. Association for Computational Linguistics.

Siyuan Huang, Zhiyuan Ma, Jintao Du, Changhua Meng, Weiqiang Wang, and Zhouhan Lin. 2024b. Mirror-consistency: Harnessing inconsistency in majority voting. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2408–2420, Miami, Florida, USA. Association for Computational Linguistics.

Aryan Keluskar, Amrita Bhattacharjee, and Huan Liu. 2024. Do LLMs understand ambiguity in text? a case study in open-world question answering. In *2024 IEEE International Conference on Big Data (BigData)*, pages 7485–7490. IEEE.

Steven Loria. 2026. TextBlob: Simplified Text Processing. TextBlob documentation, version 0.19.0; accessed 2026-05-01.

Jenish Maharjan, Anurag Garikipati, Navan Preet Singh, Leo Cyrus, Mayank Sharma, Madalina Ciobanu, Gina Barnes, Rahul Thapa, Qingqing Mao, and Ritankar Das. 2024. OpenMedLM: Prompt engineering can out-perform fine-tuning in medical question-answering with open-source large language models. *Scientific Reports*, 14(1):14156.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Lariviere, Alina Beygelzimer, Florence d'Alche Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, 22(164):1–20.

Andrew Piper, Richard Jean So, and David Bamman. 2021. Narrative theory for computational narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311. Association for Computational Linguistics.

Vladimir Propp. 1968. *Morphology of the Folktale*, 2 edition. University of Texas Press. Originally published in Russian, 1928.

Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter Sheridan Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1):31.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

SemEval-2026 Task 4 Organizers. 2026a. Narrative Similarity Dataset. GitHub repository; accessed 2026-05-01.

SemEval-2026 Task 4 Organizers. 2026b. SemEval-2026 task 4 data. Task data page; accessed 2026-05-01.

SemEval-2026 Task 4 Organizers. 2026c. SemEval-2026 task 4 results. Official task results page; accessed 2026-05-01.

Sentence Transformers. 2020. all-MiniLM-L6-v2: Sentence transformer model. Hugging Face model card; accessed 2026-05-01.

Rainer Storn and Kenneth Price. 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

Perry W. Thorndyke. 1977. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9(1):77–110.

Tzvetan Todorov. 1969. *Grammaire du Décaméron*. Mouton, The Hague and Paris.

Amos Tversky. 1977. Features of similarity. *Psychological Review*, 84(4):327–352.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, and 4 others. 2022. Taxonomy of risks posed by language models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229. Association for Computing Machinery.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.