SoloSemantics at SemEval-2026 Task 4: Triplet-Tuned MPNet for Story Similarity

Steven Au
Independent Researcher
[steventinwing@gmail.com]

Abstract

This paper describes Team SoloSemantics' submissions to SemEval-2026 Task 4: Narrative Story Similarity and Narrative Representation Learning. We began with lightweight neuro-symbolic knowledge-graph baselines, but a triplet-tuned MPNet bi-encoder produced stronger semantic separation in our experiments, so we adopted a shared dense encoder family across both tracks and kept the KG and fusion variants as diagnostic baselines. On the organizer leaderboard, Team SoloSemantics ranked 22nd on Track A and 9th on Track B. Our reproducibility audit further shows that the KG branch was often too sparse on short summaries to represent abstract narrative relations reliably under the current extraction pipeline.

1 Introduction

SemEval-2026 Task 4 introduces two related challenges: pairwise semantic choice in Track A and narrative embedding generation in Track B (Hatzel et al., 2026). Our starting hypothesis was that narrative similarity could be modeled through neuro-symbolic signals derived from entities, relations, and graph motifs. In practice, the resulting knowledge-graph baselines were interpretable but less effective than contrastive text embeddings (Reimers and Gurevych, 2019) for this task. We therefore shifted to a shared triplet-tuned bi-encoder, a setup that has worked well in zero-shot dense retrieval (Thakur et al., 2021), and used it across both tracks.

2 Background

Narrative similarity remains a fundamental challenge in computational narratology (Piper, 2021). Previous work explored story similarity by aligning plot events and character relationships across different text boundaries (Chaturvedi et al., 2018), whereas symbolic representations have attempted to model these structures through extracted narrative schemas (Chambers and Jurafsky, 2009) and interconnected plot units (Lehnert, 1981).

Recent work has emphasized the need to move beyond rigid symbolic frameworks and surface lexical overlap in order to capture deeper, more continuous structural elements through representation learning (Sterner et al., 2026). Earlier narrative datasets, such as those of Fisseni and Löwe (2012) and Chen et al. (2022), largely relied on scalar similarity ratings. Those scales are hard to apply consistently and offer limited discrimination for closely related stories. Building on that line of work, the task organizers (Hatzel et al., 2026) adopted a contrastive setup and defined narrative similarity through three components: the abstract theme, the course of action, and the outcomes of a story (Hatzel and Biemann, 2024a).

Hatzel et al. built the task from Wikipedia synopses drawn from the English Tell Me Again dataset (Hatzel and Biemann, 2024b). To keep the benchmark computationally manageable while covering many stories, the organizers restricted the data to short summaries of four to eight sentences. The annotation scheme is a variant of Best Worst Scaling. To make the task less dependent on surface text overlap, the dataset was also filtered with rejection sampling to retain only more difficult candidate triples on which commercial language models disagreed. That design choice underscores the subjective nature of narrative similarity and is reflected in the modest inter-annotator agreement reported by the organizers.

Track A evaluates pairwise semantic choice. The input is a triple containing an anchor story and two candidate stories, requiring the system to predict which candidate exhibits greater narrative similarity to the anchor. Figure 1 illustrates this comparative setup. Track B focuses on representation learning, requiring a single vector representation per story. These representations are evaluated based

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1547–1553

Dataset Split	Track A Data	Track B Data
Sample	39 pairs	108 stories
Development	200 pairs	478 stories
Test	400 pairs	849 stories
Synthetic	1900 pairs	1900 triplets

Table 1: Dataset counts used in the local experiments. The Track B synthetic file contains 1900 contrastive triplets (5700 story slots).

on how well their cosine distance aligns with the underlying partial similarity orderings annotated for the individual summaries.

Table 1 gives the split counts used in our experiments. For Track B synthetic data, the in-repo file contains 1900 contrastive triplets, corresponding to 5700 story slots.

3 System Overview

3.1 Shared Encoder

Our main encoder starts from the sentence-transformers checkpoint `all-mpnet-base-v2`, combining Sentence-BERT pooling (Reimers and Gurevych, 2019) with MPNet pretraining (Song et al., 2020). We then fine-tune it with triplet loss. We chose MPNet as the main checkpoint because the committed clean sweeps with other large encoders were weaker under the same synthetic-only protocol, with best dev accuracy of 0.655 for MPNet versus 0.630 for E5-large-v2 and 0.620 for BGE-large-en-v1.5. We train directly on the provided synthetic anchor, positive, and negative triples and do not add a separate hard-negative mining stage, since the organizer data already defines explicit positive and negative contrasts for each anchor.

3.2 Track A

Track A is a relative narrative ranking task. We evaluate a lightweight knowledge-graph baseline built from a custom extraction pipeline. The pipeline extracts named entities and dependency-based subject–verb–object triples with spaCy, falls back to rule-based regular expressions when needed, and represents the resulting structures with NetworkX. We compare this structural baseline against bi-encoder cosine ranking.

For the KG baseline, sim_{KG} is computed as a weighted sum of three pairwise graph comparisons: cosine similarity between hashed graph vectors built from Weisfeiler-Lehman subtree labels, relation labels, node-type histograms, and appended graph statistics (weight 0.6), plus Jaccard overlap over node labels (0.2) and relation labels (0.2).

To test whether structural and embedding signals are complementary, we also implement a fusion model. It computes the knowledge-graph similarity delta $\Delta_{KG} = \mathrm{sim}_{KG}(\text{anchor}, A) - \mathrm{sim}_{KG}(\text{anchor}, B)$, concatenates that value with dense embedding similarities and margin scores, and trains a logistic regression classifier over the scaled feature vectors. We also include cross-encoder and pretrained pairwise baselines as stronger text-only references.

3.3 Track B

Track B requires one embedding per story, so the goal is to preserve narrative neighborhood structure in a single vector. We test a knowledge-graph vector baseline, whole-text embeddings from the bi-encoder, and chunked newline pooling from the same encoder. In the chunked variant, the model encodes text segments separated by existing newline characters and then mean-pools the resulting vectors, with the aim of retaining local plot motifs. In the current Track B development file in the repo, this segmentation is shallow rather than paragraph-like: stories split into a median of one chunk, with mean 1.59 chunks and a maximum of six. We therefore treat newline pooling as a lightweight formatting heuristic rather than as evidence that newline units are inherently better than sentence-level segmentation.

Figure 2 summarizes the high-level modeling pipeline, including the lightweight KG baseline, the submitted dense branch for Track A, and the whole-text versus chunked Track B embedding variants.

4 Experimental Setup

All neural models were trained and evaluated on a single NVIDIA GeForce RTX 3080 GPU. The main triplet sweep explored learning rates of 1e-5 and 2e-5, margins of 0.1 and 0.2, training durations of 1, 2, 3, and 5 epochs, and maximum sequence lengths of 192 and 256. The strongest clean-dev-selected run used learning rate 2e-5, margin 0.2, 2 epochs, and maximum sequence length 192, while the final sample-selected submission used the same learning rate with margin 0.1.

Because the repo contains both synthetic-only and public-label-adapted artifacts, we report three validation regimes. *Clean* excludes dev and sample labels from training, calibration, and model

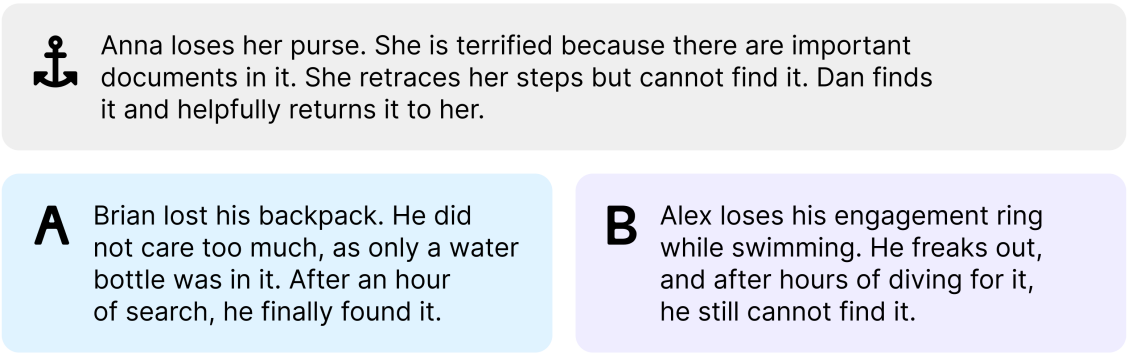

Figure 1: The Track A classification setup asking to choose the narratively more similar text. Story A is considered more similar. A, B, and the anchor all tell the story of a lost item that is retrieved. In the case of A, it is found by a third party (as it is in the anchor), while in B it is not found at all.

selection. *Clean-dev-selected* keeps synthetic-only training but uses dev labels for checkpoint or hyperparameter selection. *Public-label-adapted* covers runs that reuse dev or sample labels in training or final model selection.

All local comparison tables therefore report visible-data evidence only, while official hidden-test performance is listed separately in Table 2.

Track	Rank	Accuracy (%)
A	22	68.00
B	9	66.50

Table 2: Official hidden-test scores and ranks from the organizer leaderboard.

5 Results

5.1 Track A

Method	Regime	Dev Acc	Sample Acc
KG similarity baseline	clean	0.435	0.615
KG + logistic regression	clean	0.550	0.513
Triplet bi-encoder cosine	clean	0.630	0.667
Triplet sweep best-dev	clean-dev-selected	**0.655**	0.718
Triplet sweep best-sample	public-label-adapted	0.650	**0.769**
KG + embedding fusion	public-label-adapted	0.675	–

Table 3: Track A local results under clean and adapted evaluation regimes. The fusion sample score is omitted because the saved fusion runs reference the missing encoder path `models/st_mpnet_triplet_a`.

Table 3 summarizes the visible-data Track A comparisons. The KG baselines lagged well behind the triplet MPNet family. These results should not be read as evidence against symbolic narrative modeling in general. Instead, they suggest that our lightweight entity-relation extraction pipeline was too sparse and brittle to capture abstract narrative similarity reliably in short summaries. The highest local dev score came from KG + embedding fusion, but the cleanest reproducible evidence comes from the synthetic-trained triplet family. Because the exact encoder dependency for the committed KG + embedding fusion runs is incomplete, and because the fusion gains depend on public-label-adapted calibration, we treat the fusion aggregate scores as diagnostic rather than primary evidence and use the shared triplet encoder as the main submitted Track A system.

5.2 Track B

Track B has no public local hidden-test labels in the repo, so we report two diagnostics only: proxy accuracy on Track A dev triples and synthetic alignment on contrastive triples. Proxy accuracy is computed by embedding the Track A dev anchor, candidate A, and candidate B independently, then checking whether $\cos(\text{anchor}, A) \geq \cos(\text{anchor}, B)$ matches the annotated `text_a_is_closer` label.

Because synthetic alignment is computed against the same contrastive structure used during training, it should be interpreted only as a sanity check rather than as an estimate of hidden-test performance.

Table 4 summarizes the local Track B diagnostics. The visible-data diagnostics favor the sweep best-sample (whole-text) model, whereas the final sweep best-sample (chunked submission) run preserves perfect synthetic alignment but drops to 0.600 proxy accuracy. These diagnostics helped

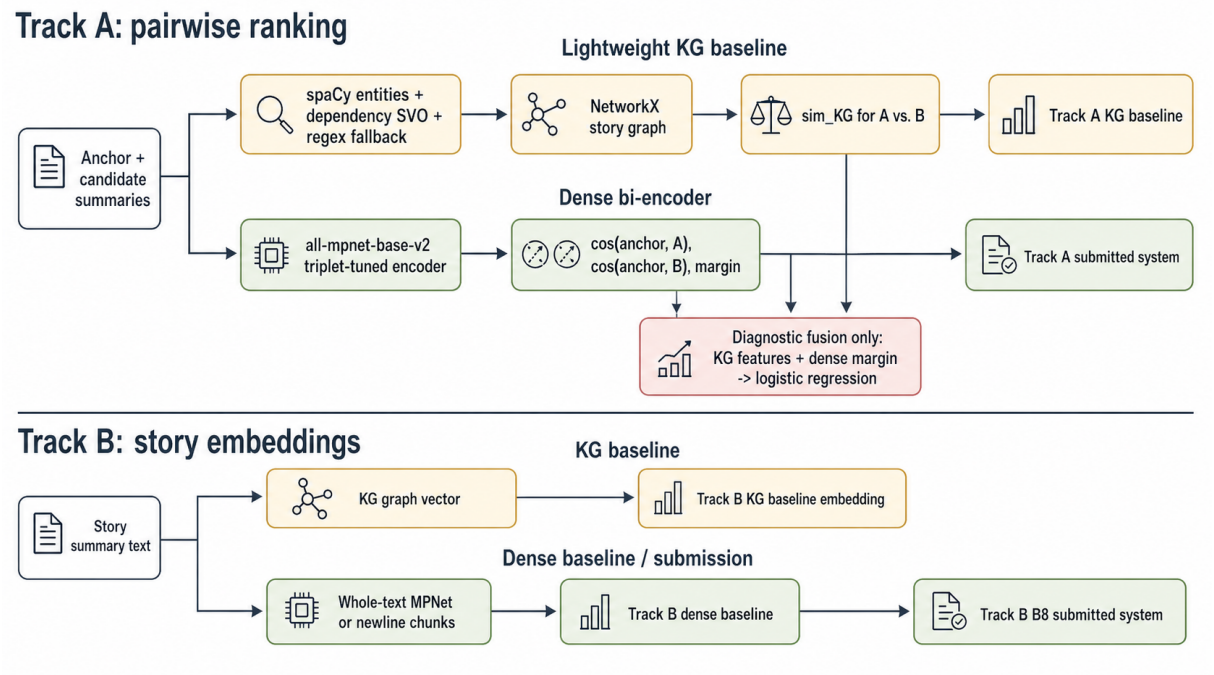

Figure 2: System overview. For Track A, we compare a lightweight KG baseline against a triplet-tuned dense bi-encoder, with fusion used only as a diagnostic. For Track B, we compare a KG vector baseline against dense MPNet-based story embeddings, including the chunked best-sample submission variant used for the final run.

Method	Regime	Proxy (dev)	Synth Align
KG baseline	clean	0.460	0.598
MiniLM embedding baseline	clean	0.550	0.989
MPNet embedding baseline	clean	0.615	0.995
Triplet whole-text embedding	clean	0.630	0.9995
Triplet chunked embedding	clean	0.635	0.9995
Sweep best-sample (whole-text)	public-label-adapted	**0.650**	**1.000**
Sweep best-sample (chunked submission)	public-label-adapted	0.600	**1.000**

Table 4: Track B local proxy and synthetic diagnostics. Synthetic alignment is a sanity check, not a hidden-test estimate.

guide iteration, but they are not a substitute for hidden-test evaluation. The official Track B submission used the same sweep best-sample (chunked submission) configuration listed in the table, but the local evidence shows only small and inconsistent differences between whole-text and chunked pooling. We therefore describe chunked pooling as a task-specific submission choice rather than a uniformly stronger local encoder.

5.3 Analysis of Performance

Table 5 summarizes KG coverage across the available repo splits. The KG pipeline remained sparse on many short summaries. On Track A dev, unique stories averaged 1.21 non-mention relation edges, with 31.9% of stories yielding zero extracted SVO-style edges and 67.4% yielding at most one relation edge. At the triple level, examples correctly ranked by the KG similarity baseline averaged 1.32 relation edges across their three stories, compared with 1.18 for KG similarity baseline failures. The difference is small, but it is consistent with the broader pattern that structural extraction often lacked enough explicit event content to compete with dense semantic matching. A visual summary of the same coverage pattern appears in Appendix Figure 4.

The results suggest that triplet fine-tuning improved separation between narratively similar and dissimilar summaries beyond pretrained sentence embeddings. However, because the task examples are short Wikipedia-style summaries, the system may still rely partly on high-level semantic overlap rather than deeper event-structure reasoning.

Preliminary inspection of dev examples suggests that the system struggles most when two candidates share a broad theme with the anchor but differ in causal structure, agency, or final outcome. Dense bi-encoders compress each story into a single vector, which can blur whether a conflict is resolved, whether the same character type drives the action, or whether the ending reverses the anchor's narrative trajectory. The KG baseline occasionally helps when explicit participants or event schemas align cleanly, but such cases are relatively rare under the current extraction pipeline. A reproducible fusion baseline sometimes rescues dense errors, yet

Split	Stories	Avg entities	Avg rels	Avg nodes	Avg edges	Zero SVO (%)	Sparse (%)
Track A sample	108	9.80	1.30	50.00	16.07	33.3	64.8
Track A dev	479	9.38	1.21	49.16	16.87	31.9	67.4
Track A test	848	9.47	1.33	49.01	16.99	28.2	64.3
Track B sample	108	9.80	1.30	50.00	16.07	33.3	64.8
Track B dev	478	9.37	1.20	49.14	16.88	32.0	67.6
Track B test	849	9.41	1.25	49.10	16.81	30.4	66.5

Table 5: KG coverage by split. Relation counts exclude `mentions` and `co_occurs` edges.

coefficient analysis still points to the dense similarity margin as the strongest individual feature, which is consistent with the adapted-only nature of the larger fusion gains. Appendix Figure 3 shows one development triple in which the dense encoder correctly recovers the broader romance-and-class trajectory, while the lightweight KG baseline is drawn toward a candidate whose extracted local actions look superficially similar.

6 Conclusion

This paper presented a unified triplet-tuned MPNet bi-encoder for SemEval-2026 Task 4. Across both tracks, dense contrastive representations outperformed our lightweight entity-relation graph baselines and provided a simple, consistent architecture for pairwise ranking and story embedding generation. The system ranked 22nd on Track A and 9th on Track B on the organizer leaderboard. These results suggest that synthetic contrastive triples can serve as useful supervision for narrative similarity, while our diagnostics also point to the need for stronger symbolic representations, stricter leakage controls, and hierarchical encoders for longer narratives.

7 Limitations

Our methodology has several limitations. First, the lightweight entity-relation pipeline is sparse on short summaries, which weakens the KG branch precisely on the implicit thematic cases that dominate this shared task. This should not be read as evidence against symbolic narrative modeling in general; it reflects the limits of the lightweight extraction pipeline used here. Second, the main neural training signal comes from synthetic triples, so the encoder may inherit synthetic-data biases even when hidden-test transfer is acceptable. Third, some historical checkpoints referenced by the repo, notably `models/st_mpnet_triplet_a` and the clean sweep directories, were not preserved, so part of the historical record can be verified only through stored result artifacts rather than exact reruns. Finally, the encoder settings are tuned for short Wikipedia-style synopses and would likely need hierarchical extensions for substantially longer narratives.

References

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610. Association for Computational Linguistics.

Snigdha Chaturvedi, Shashank Srivastava, and Dan Roth. 2018. Where have i heard this story before? identifying narrative similarity in movie remakes. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemysław Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106. Association for Computational Linguistics.

Bernhard Fisseni and Benedikt Löwe. 2012. Which dimensions of narrative are relevant for human judgments of story equivalence? In *Proceedings of the 3rd Workshop on Computational Models of Narrative*.

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*. Association for Computational Linguistics.

Hans Ole Hatzel and Chris Biemann. 2024a. Story embeddings for narrative similarity. In *Proceedings of the Association for Computational Linguistics*.

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 Forteenth Language Resources and Evaluation Conference*, Turin, Italy.

Wendy G. Lehnert. 1981. Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331.

Andrew Piper. 2021. Computational narratology. *Journal of Cultural Analytics*, 6(4).

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 (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.

Igor Sterner, Alex Lascarides, and Frank Keller. 2026. Contrastive learning with narrative twins for modeling story salience. *arXiv preprint arXiv:2601.07765*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

A Reproducibility

All generated evidence files for the paper are recreated by running `python scripts/build_paper_artifacts.py`. The command writes audited tables, verification files, and revision notes to `artifacts/paper/`.

B Qualitative Example

Anchor
Lucile leaves a secure but loveless arrangement, falls for a poorer man, becomes pregnant, and eventually returns to the older benefactor who waited for her.

Candidate A (gold)
A wealthy heiress falls for a lower-status man, flees family control, and the couple survive class decline before finally stabilizing together.

Candidate B
A reporter marries after a pregnancy but repeatedly leaves for work and war, creating local event overlap without the same romance-and-class trajectory.

Model Outputs
Gold label: **Candidate A**. KG similarity baseline chooses **Candidate B** with $\text{sim}_{KG}(A) = 0.6045$ and $\text{sim}_{KG}(B) = 0.6412$. Triplet bi-encoder cosine chooses **Candidate A** with $\cos(\text{anchor}, A) = 0.6821$ and $\cos(\text{anchor}, B) = 0.3798$. KG + embedding fusion also chooses **Candidate A** with $P(A) = 0.8722$.

Interpretation
The dense encoder tracks the broader romance-and-class arc shared by the anchor and Candidate A, whereas the lightweight KG baseline is drawn toward Candidate B by locally extracted actions such as meeting, pregnancy, and leaving.

Figure 3: Stylized development example from the reproducible error-analysis bundle (`fusion_public_096`). The dense models recover the broader romance-and-class trajectory, while the lightweight KG baseline overweights locally extracted action overlap.

C KG Coverage Figure

Method	Core settings	Data and regime
KG + logistic regression	KG features only; LogisticRegression	Synthetic classification triples; clean; dev excl. yes
Triplet bi-encoder cosine	all-mpnet-base-v2; TripletLoss, bs=16	Synthetic classification triples; clean; dev excl. yes
Triplet sweep best-dev	Triplet MPNet checkpoint; TripletLoss, m=0.2, lr=2e-05, ep=2, maxlen=192, seed=42	Synthetic classification triples; clean-dev-selected; dev excl. yes
Triplet sweep best-sample	Triplet MPNet checkpoint; TripletLoss, m=0.1, lr=2e-05, bs=16, ep=2, maxlen=192, seed=42	Synthetic classification triples; public-label-adapted; dev excl. yes (training) / no (model selection uses sample)
KG + embedding fusion (dev model)	Triplet MPNet checkpoint; LogisticRegression, bs=32, seed=42	Track A dev pairs; public-label-adapted; dev excl. no
KG + embedding fusion (public-MPNet baseline)	all-mpnet-base-v2; LogisticRegression, bs=32	Track A dev pairs + synthetic; public-label-adapted; dev excl. no

Table 6: Key reproducibility settings for the main committed models. Common dense settings are mean pooling, no hard negative mining, and RTX 3080 hardware unless otherwise noted. Fields not logged in committed artifacts are omitted rather than inferred.

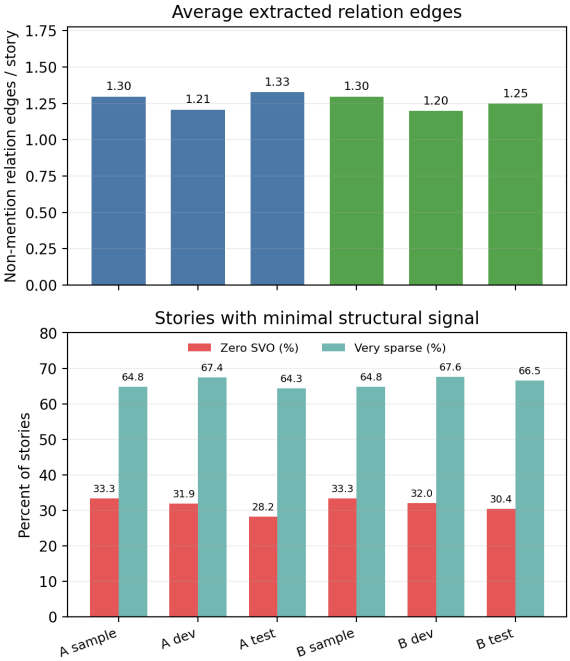

Figure 4: Lightweight KG coverage across the available repo splits. Even before pairwise comparison, many stories produce zero or at most one non-mention relation edge, limiting the structural signal available to the KG baseline.