Long-Form Information Alignment Evaluation Beyond Atomic Facts

Danna Zheng, Mirella Lapata, Jeff Z. Pan
School of Informatics, University of Edinburgh, UK
dzheng@ed.ac.uk, mlap@inf.ed.ac.uk, http://knowledge-representation.org/j.z.pan/

Abstract

Information alignment evaluators are vital for various NLG evaluation tasks and trustworthy LLM deployment, reducing hallucinations and enhancing user trust. Current fine-grained methods, like FactScore, verify facts individually but neglect inter-fact dependencies, enabling subtle vulnerabilities. In this work, we introduce MONTAGELIE, a challenging benchmark that constructs deceptive narratives by "montaging" truthful statements without introducing explicit hallucinations. We demonstrate that both coarse-grained LLM-based evaluators and current fine-grained frameworks are susceptible to this attack, with AUC-ROC scores falling below 65%. To enable more robust fine-grained evaluation, we propose DOVESCORE, a novel framework that jointly verifies factual accuracy and event-order consistency. By modeling inter-fact relationships, DOVESCORE outperforms existing fine-grained methods by over 8%, providing a more robust solution for long-form text alignment evaluation. Our code and datasets are available at https://github.com/dannalily/DoveScore.

1 Introduction

Factual inaccuracies (i.e., hallucinations) (Pan et al., 2023; Huang et al., 2025b) in large language model (LLM) deployment have been identified as a critical challenge (Huang et al., 2025a). To address this challenge, recent approaches (Yan et al., 2025a,b; Wang et al., 2025; Asai et al., 2024; Roy et al., 2024; Ji et al., 2023; Manakul et al., 2023) introduce reflection mechanisms that perform post-hoc verification by comparing generated texts against retrieved documents or given contexts, and subsequently regenerating erroneous segments when necessary. At the core of these approaches are information alignment evaluators,

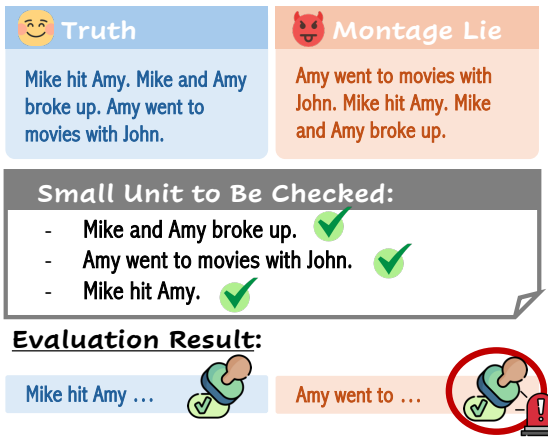


Figure 1: The figure illustrates the limitation of existing fine-grained evaluators (e.g., FactScore and AlignScore): Failure to detect lies composed of accurate factual units.

which determine whether a target text accurately aligns with a source text, playing a vital role for various NLG evaluation tasks and trustworthy (Zheng et al., 2024; Huang et al., 2025c) LLM deployment. Deng et al. (2021) highlight the importance of information alignment evaluation across various NLG evaluation tasks. Consequently, developing robust information alignment evaluators is essential.

Unlike fact-checking tasks (Yan et al., 2025a,b; Si et al., 2024; Ma et al., 2024), which typically involve short, sentence-level comparisons, information alignment evaluation often requires reasoning (Li et al., 2024; Zhang et al., 2024a; Mellish and Pan, 2008) over extended contexts (Zha et al., 2023), where both the source and target may span multiple paragraphs. While recent advances in long-context LLMs (Liu et al., 2025; Huang et al., 2025b) support coarse-grained alignment evaluation across entire texts, many real-world applica-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 11019–11039
November 4-9, 2025

tions (Min et al., 2023; Zhang et al., 2024b; Ye et al., 2025) demand fine-grained evaluators that can assess individual factual units and pinpoint detailed inaccuracies.

Existing fine-grained frameworks (Yan et al., 2025a,b; Min et al., 2023; Song et al., 2024b; Wei et al., 2024) typically decompose the target text into atomic facts, verify each against retrieved evidence, and then aggregate the results into an overall judgment. While effective at identifying surface-level errors, this approach has a critical blind spot: it overlooks relationships and dependencies between facts. Even when all individual statements are accurate, reordering them can reverse implied causal chains and mislead readers. As illustrated in Figure 1, the Lie version suggests that Amy's outing with John triggered Mike's violence, subtly shifting blame onto her. The Truth reveals that the violence preceded the breakup and her subsequent actions. By altering the sequence of accurate statements, the text introduces a discourse-level manipulation that distorts causality without introducing any falsehoods. However, existing fine-grained evaluators are inherently incapable of detecting such manipulations.

To investigate this vulnerability, we introduce **MONTAGELIE**, a novel benchmark designed to test the limitations of current information alignment evaluators. Drawing inspiration from the cinematic concept of montage[1], which creates new meaning by rearranging real scenes in novel sequences, MONTAGELIE constructs "montage-style lies": deceptive texts composed entirely of truthful statements, deliberately reordered to imply misleading narratives. These manipulations do not introduce fabricated facts but instead distort causal relationships by altering the sequence of events. To systematically assess model robustness, our benchmark includes four levels of difficulty, each reflecting increasing subtlety in the causal distortion.

Such rearranging strategies, while factually accurate at the level of small textual units, exploit deceptive tactics commonly used in human communication and pose a sophisticated challenge that current evaluators are ill-equipped to detect. They also represent a realistic and underexplored attack vector in adversarial prompting (Kim et al., 2024; Yu et al., 2024) and misinformation campaigns (Hu et al., 2025; Macko et al., 2025). Experimental results demonstrate that existing fine-grained frameworks, as well as state-of-the-art long-context LLMs in coarse-grained evaluation settings, struggle to identify these subtle manipulations, achieving AUC-ROC scores consistently below 65%.

Besides, to address the limitations of current fine-grained evaluators, we propose **DOVESCORE** (**D**escriptive and **O**rdered-Event **Ve**rification **Score**), a fine-grained evaluation framework that explicitly incorporates both atomic factual accuracy and event ordering consistency. DOVESCORE decomposes target texts into descriptive and event-based facts, verifying their individual correctness and event sequencing against the source text, then computing a weighted precision score. Experimental evaluations show that DOVESCORE outperforms existing fine-grained methods by over 8%.

We anticipate that both our benchmark, MONTAGELIE, and our proposed evaluation method, DOVESCORE, will offer valuable insights and make a meaningful contribution to the ongoing development of robust and reliable information alignment evaluators.

2 Background and Related Works

This section reviews key definitions, existing benchmarks, and evaluators related to information alignment, laying the groundwork for identifying gaps addressed by our work.

2.1 Definition of Information Alignment

The concept of information alignment (also termed factual consistency) was first formally defined by Deng et al. (2021) as a core principle of their unified evaluation framework for natural language generation (NLG) tasks. Initially, it was defined at the token level: each token in the target should be supported by the source. Later, Zha et al. (2023) introduced a more practical sequence-level definition: Text b aligns with source a if all information in b appears accurately in a without contradiction. An Information Alignment Evaluator can be formalized as:

$$f : (\boldsymbol{a}, \boldsymbol{b}) \rightarrow y \tag{1}$$

where higher y indicates stronger alignment of b with respect to a.

[1] https://en.wikipedia.org/wiki/Montage_(filmmaking)

2.2 Benchmarks

Early benchmarks mainly addressed sentence-level alignment, such as FEVER (Thorne et al., 2018), FEVEROUS (Aly et al., 2021), and AVERITEC (Schlichtkrull et al., 2023). More recent efforts target longer texts and diverse domains. SummaC (Laban et al., 2022) aggregates six summarization-focused datasets, while TRUE (Honovich et al., 2022) combines 11 datasets across summarization, dialogue, fact verification, and paraphrasing. The latest, LLM-AggreFact (Tang et al., 2024a), curates recent datasets such as AGGREFACT (Tang et al., 2023), TOFUEVAL (Tang et al., 2024b), and ClaimVerify (Liu et al., 2023a).

Despite broader coverage, these benchmarks mostly focus on unsupported or contradictory claims. Our proposed dataset, MONTAGELIE, introduces a harder case: each individual claim aligns with the source, yet their combination yields a misleading narrative.

2.3 Evaluators

Coarse-grained Evaluators These methods assess alignment holistically, typically via overlap or semantic similarity. Traditional metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) emphasize lexical overlap, while embedding-based metrics (e.g., BERTScore (Zhang et al., 2020), BartScore (Yuan et al., 2021)) and fine-tuned models (e.g., BLEURT (Sellam et al., 2020), FactCC (Kryscinski et al., 2020)) capture semantics more effectively. However, most struggle with long-form texts due to model limitations. More recently, LLM-as-Evaluator approaches (Liu et al., 2023b; Luo et al., 2023) prompt LLMs directly for quality scores, enabling more flexible, semantically rich assessments.

Fine-grained Evaluators Fine-grained methods offer interpretable, diagnostic feedback. QA-based approaches (e.g., QuestEval (Scialom et al., 2021), QAFactEval (Fabbri et al., 2022)) extract entities, generate questions, and verify answers using the source. These methods are limited by coverage and are computationally intensive. Another strategy segments the target into sentences (Laban et al., 2022; Zha et al., 2023) and verifies them individually, but this ignores cross-sentence dependencies. Recent work leverages LLMs to decompose texts into atomic facts (Min et al., 2023; Song et al., 2024b,a; Wei et al., 2024), offering more precise semantic units and better handling of indirect references.

Yet, all existing methods verify claims independently, failing to detect montage lies—cases where individually correct claims, when combined, form a misleading whole. This independence assumption prevents them from capturing higher-order semantics across claims. To overcome this, we propose DOVESCORE, a novel framework that explicitly models inter-fact relationships.

3 MONTAGELIE: Information Alignment Evaluation Benchmark

We introduce MONTAGELIE, a dedicated benchmark designed to evaluate whether information alignment evaluators can detect misalignment in texts that retain accurate individual claims, but distort the overall intended narrative. Below we describe (1) the data construction process, (2) dataset statistics and quality checks, and (3) the evaluation metrics.

3.1 Data Construction

The construction process for MONTAGELIE consists of three main stages: seed data sampling, montage-style lie generation, and paraphrasing. The LLM used in data construction process is `gpt-4o-mini-2024-07-18`, and prompts are shown in Table 5 in Appendix A.1.

3.1.1 Seed Data Sampling

We start with publicly available long-form summarization datasets and randomly sample pairs (s, g) from each, where s is the source document and g is its corresponding summary, which we label as correct target text.

- **SummScreen** (Chen et al., 2022): TV-series transcripts paired with human-written recaps capturing dialogue-driven narratives and character actions.
- **BookSum** (Kryscinski et al., 2022): Literary texts paired with long-form summaries, emphasizing long-range causal and temporal dependencies.

These datasets were chosen for their narrative intensity and complementary styles (dialogue versus exposition). We sample uniformly to cover a diverse range of source and target text lengths.

3.1.2 Montage-Style Lie Generation

For each correct target text g, we generate four montage-style lies l_e, l_m, l_h, and l_{eh} at varying difficulty levels: easy, medium, hard, and extreme hard.

Step 1: Decompose g into Events E We prompt the LLM to decompose g into a chronological sequence of independent events, denoted as: $E = [e_1, e_2, \ldots, e_n]$. Figure 6 in Appendix B.2 presents the distribution of the number of decomposed events.

Step 2: Shuffle E with Controlled Difficulty We define difficulty based on the Shuffle Degree ShuffleD, which is to measure how out-of-order a permuted list $F = [f_1, f_2, \ldots, f_n]$ is relative to the original E. Let π be the unique permutation such that $f_i = e_{\pi(i)}$. Then ShuffleD is defined as:

$$\text{ShuffleD}(E, F) = \frac{\text{Inv}(\pi)}{\text{Inv}_{\max}(n)} \in [0, 1] \quad (2)$$

where $\text{Inv}(\pi)$ is the inversion count:

$$\text{Inv}(\pi) = \left|\{(i, j) \mid 1 \le i < j \le n,\ \pi(i) > \pi(j)\}\right| \quad (3)$$

and $\text{Inv}_{\max}(n) = \binom{n}{2} = \frac{n(n-1)}{2}$ is the maximum possible number of inversions.

A lower value of ShuffleD implies a more similar sequence to the original, making the lie harder to detect. We define the difficulty levels as follows: easy when $D \in [0.80, 0.90]$, medium when $D \in [0.55, 0.65]$, hard when $D \in [0.30, 0.40]$, and extreme hard when $D \in [0.05, 0.15]$. These intervals are disjoint to ensure clear separation between difficulty levels.

To generate a permutation corresponding to a given difficulty level, we first determine the appropriate range for $\text{Inv}(\pi)$ and randomly sample a target inversion count within this range. Fig 7 (see Appendix B.2) shows the distribution of the sampled inversion count in our data construction process. We then use Lehmer codes (Knuth, 1998) to construct a permutation[2] that has the exact sampled inversion count. As illustrated in Algorithm 1 (see Appendix B.1), the process begins with the maximal Lehmer code, and iteratively decrements its entries at random until the total inversion count equals the target. The resulting Lehmer code is then decoded into the corresponding permutation.

Step 3: Incremental Lie Generation After obtaining F, we generate the lie text incrementally. Giving the full F to the LLM often leads to failure in preserving the order, so instead, we use an incremental generation strategy. We start with the first event in F and sequentially add the rest. At each step, LLM is explicitly instructed that the new event occurs after the existing text and should be integrated as the next logical event in the narrative. The LLM is asked to continue the paragraph in a natural and coherent manner, without inserting unnecessary transitional phrases or restructuring earlier content. This approach enables precise control over event order while ensuring that the generated lie remains fluent and lexically close to the original g, yet semantically altered due to the reordering.

3.1.3 Paraphrasing

To test whether alignment evaluators are sensitive to narrative variation, we also generate paraphrases of both the correct target text and the lies. These paraphrases preserve the meaning of the original but present the events in a different narrative order or style.

We instruct LLM to rephrase the text using a different narrative technique (chronological, flashback, interjection, supplementary narration) than the original. For each correct target text g and its lies l_e, l_m, l_h, l_{eh}, we generate corresponding paraphrases g', l'_e, l'_m, l'_h, and l'_{eh}.

3.2 Dataset Summary

3.2.1 Dataset Format and Statistic

Each data instance in the MONTAGELIE benchmark is represented as a tuple:

$$d = <s, g, l_e, l_m, l_h, l_{eh}, g', l'_e, l'_m, l'_h, l'_{eh}> \quad (4)$$

where s is the source text. The texts g and g' are aligned with the source text and labeled as 1, while l_e, l_m, l_h, l_{eh} and their paraphrases are not aligned with the source and are labeled as 0.

Table 1 presents detailed statistics for MONTAGELIE, which comprises 1,303 data instances.

[2]Exhaustively enumerating permutations is computationally infeasible due to the combinatorial explosion of the permutation space.

Property	Number
Total Instance	1303
Instance from BOOKSUM	637
Instance from SUMM_SCREEN	666
Word Lengths of Source Text (Min, Max, Avg)	(312, 9937, 4201.79)
Word Lengths of Target Text (Min, Max, Avg)	(62, 991, 258.61)

Table 1: Statistics of MONTAGELIE Benchmark

Difficulty	Generated Lies			Paraphrase	
	SemanticS	EventI	Coherence	SemanticF	StructuralV
Easy	100.00	100.00	94.00	98.00	98.00
Medium	100.00	98.00	98.00	100.00	98.00
Hard	98.00	100.00	96.00	98.00	100.00
Extreme	96.00	100.00	98.00	100.00	98.00

Table 2: Human Evaluation of MONTAGELIE Benchmark Quality (SemanticS=Semantic Shift; EventI=Event Integrity; SemanticF=Semantic Fidelity; StructuralV=Structural Variation).

Source texts contain up to 9,937 words, while target texts have lengths of up to 991 words. The length distributions are illustrated in Figure 5 (see Appendix B.2).

3.2.2 Data Quality

To verify data quality, we conducted a human evaluation, separately assessing montage-style lies and paraphrases.

Montage-style lies were evaluated on three criteria: **Semantic Shift** — whether the meaning differs from g; **Event Integrity** — whether the core events remain unchanged (no additions, deletions, or alterations); **Coherence** — whether the text reads smoothly, without awkward transitions or overuse of conjunctions.

Paraphrases were assessed on two criteria: **Semantic Fidelity** — whether the meaning remains faithful to g; **Structural Variation** — whether the narrative structure is meaningfully altered.

Annotators labeled each instance as `yes` or `no`. To standardize evaluation, two annotators jointly labeled five instances, achieving agreement scores of 100%, 100%, and 87.5% for montage-style lies, and 87.5% and 75.0% for paraphrases.

We then randomly sampled 50 instances, yielding 200 lies and 200 paraphrases for human evaluation. Each annotator evaluated 25 instances. As shown in Table 2, most examples meet all criteria, confirming the overall quality of the benchmark.

3.3 Evaluation Metrics

We assess the effectiveness of the alignment evaluator using the AUC-ROC score, which quantifies the evaluator's ability to distinguish the correct target texts from deceptive alternatives ("lies") across varying levels of difficulty. For each target text, the evaluator assigns a score indicating how well it aligns with the given source. To compute the AUC-ROC for a specific difficulty level, we compare the scores of correct target texts with those of their corresponding deceptive counterparts. This process is repeated independently for each of the four difficulty levels. Finally, we report the overall effectiveness as the average AUC-ROC across all difficulty levels.

4 MONTAGELIE Challenges Evaluators

4.1 Evaluated Evaluators

Coarse-grained Evaluators MONTAGELIE comprises long-form source and target texts, which limits the applicability of evaluators that are not designed to handle long contexts. Therefore, we restrict our evaluation to evaluators that support long-form input. Specifically, we report ROUGE-1, ROUGE-2, and ROUGE-L scores, as well as evaluations from LLM-as-Evaluator methods. For the latter, we adopt prompts adapted from G-Eval's consistency evaluation template (see Table 6 in Appendix A.2), and employ long-context LLMs, including `gpt-4o-mini-2024-07-18`, `Qwen-3` models (1.7B, 4B, 8B, 14B, 32B), and various `Llama3-instruct` models (1B, 3B, 7B, and 70B).

Fine-grained Evaluators As discussed in Section 2.3, existing fine-grained evaluators inherently struggle to detect montage-lies. To empirically validate this limitation, we report results from four representative methods: SummaC-ZS, SummaC-Conv, AlignScore, and FactScore. SummaC-ZS/Conv and AlignScore decompose the target text into individual sentences. SummaC-ZS/Conv leverages a natural language inference (NLI)-based model to assess the factual consistency of each sentence with the source, whereas AlignScore employs a fine-tuned model for the same purpose. In contrast, FactScore decomposes the target text into atomic facts and verifies each one via LLM prompting. In our experiments, we

	Evaluator	Easy	Medium	Hard	Extreme	AVG
Coarse-Grained	ROUGE-1	53.92	54.06	53.77	54.11	53.96
	ROUGE-2	54.57	54.93	54.71	54.63	54.71
	ROUGE-L	53.91	54.15	53.91	54.40	54.09
	qwen-3-1.7b	51.14	51.21	50.76	50.48	50.90
	qwen-3-4b	60.27	58.90	57.50	54.93	57.90
	qwen-3-8b	64.43	60.98	57.83	54.20	59.36
	qwen-3-14b	62.22	61.88	60.63	57.95	60.67
	qwen-3-32b	65.80	65.80	63.13	**59.24**	63.49
	llama-3.2-instruct-1b	50.20	49.79	49.75	49.70	49.86
	llama-3.2-instruct-3b	49.44	49.30	49.84	50.08	49.66
	llama-3.1-instruct-8b	56.53	56.52	55.22	53.60	55.47
	llama-3.3-instruct-70b	61.14	60.87	58.41	54.44	58.71
	gpt-4o-mini	**68.77**	**66.39**	**63.17**	58.57	**64.23**
Fine-Grained	SummaC-ZS	51.13	51.69	51.96	51.70	51.62
	SummaC-Conv	**56.54**	56.02	55.17	55.68	55.85
	AlignScore	56.51	**56.89**	**56.82**	**56.30**	**56.63**
	FactScore (gpt-4o-mini)	50.85	51.06	50.24	49.65	50.45

Table 3: AUC-ROC Performance of existing evaluators on MontageLie.

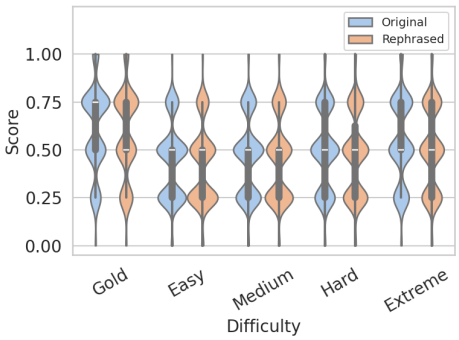


Figure 2: Violin plots of scores from `gpt-4o-mini` on MONTAGELIE. The similar distributions for original and rephrased targets indicate robustness to rephrasing. Comparable trends are observed for other evaluators (see Appendix C).

use `gpt-4o-mini-2024-07-18` as the LLM backbone for FactScore.

4.2 Experiment Setup

For all evaluations, we apply greedy decoding with a temperature of 0 to ensure deterministic outputs. All LLM-based evaluations are conducted in a zero-shot setting.

4.3 Results

Table 3 summarizes the performance of existing evaluators on the MONTAGELIES benchmark.

Upper Bound of Current Evaluators Remains Low The best-performing model, `gpt-4o-mini`, achieves an average AUC-ROC of only 64.23%, reflecting the inherent difficulty of the task and underscoring the pressing need for more effective information alignment evaluators.

Fine-Grained Evaluators Suffers from Inherent Difficulty Fine-grained evaluation methods struggle to detect montage-style lies, as they often overlook the relationships between decomposed factual units. All evaluated methods achieve AUC-ROC scores below 57%, highlighting their limited effectiveness. SummaC and AlignScore outperform the LLM-based FactScore, likely due to their coarser sentence-level segmentation, whereas FactScore operates at the more granular level of atomic facts. When both methods are based on `gpt-4o-mini`, the fine-grained FactScore underperforms its coarse-grained counterpart by 13.78%.

Lexical Similarity Limits ROUGE's Effectiveness The ROUGE score achieves an AUC-ROC of approximately 54%, as the correct target text and the montage lies are lexically similar by design during data construction. As a result, traditional n-gram-based metrics like ROUGE struggle to effectively distinguish between true and false content.

Small LLMs (<4B) Are Ineffective LLMs with fewer than 4 billion parameters in both the Qwen3 and LLaMA3 families perform poorly, achieving AUC-ROC scores below 51%, which suggests they

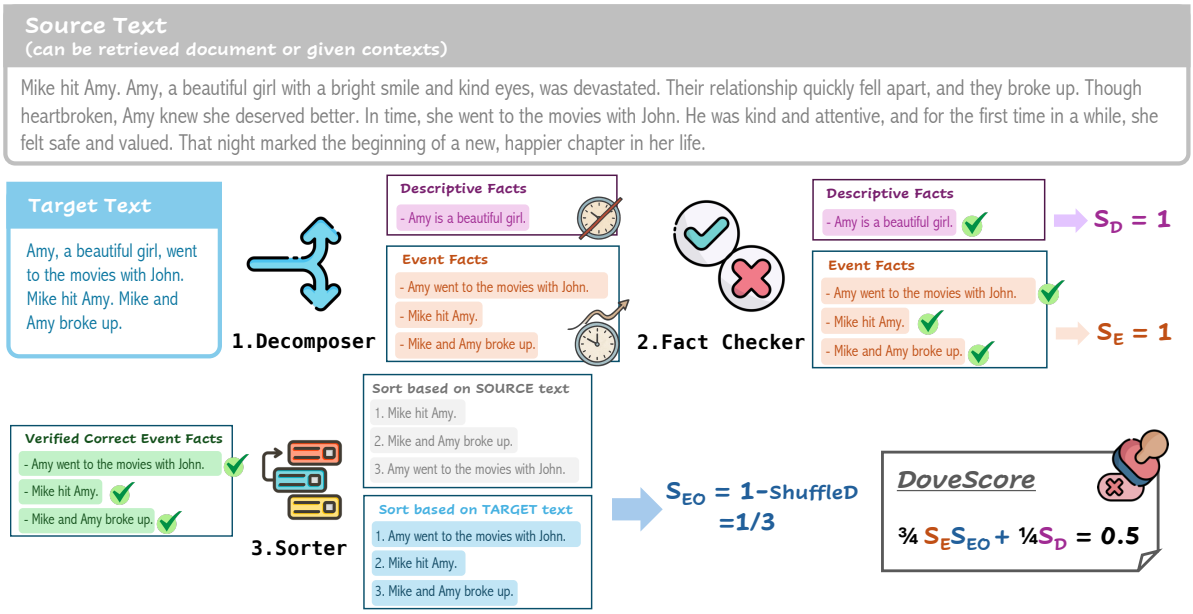


Figure 3: Illustration of DOVESCORE's three core components (Decomposer, Fact Checker, Sorter).

lack the capacity to reliably detect montage-style lies. As shown in the score distribution in Figure 8 (see Appendix C), these smaller LLMs tend to assign high scores to both accurate target text and montage-style lies.

Qwen3 Outperform LLaMA3 Counterparts Across comparable model sizes, Qwen3 consistently outperforms LLaMA3: `Qwen-3-8B` outperforms `LLaMA-3.1-instruct-8B` by ∼4%, and `Qwen-3-32B` surpasses `LLaMA-3.3-70B` by ∼5%.

Evaluators Are Robust to Narrative Variations We find that existing evaluators are generally robust to variations in narrative technique. As detailed in Section 3.1.3, we rephrased each target text to alter the narrative order while preserving its original semantics. As shown in Figure 2, the score distributions for the original and rephrased target texts are highly similar. This suggests that current models are not easily misled by paraphrasing, a desirable property in factuality evaluation.

5 DOVESCORE: A Fine-Grained Information Alignment Evaluation Framework

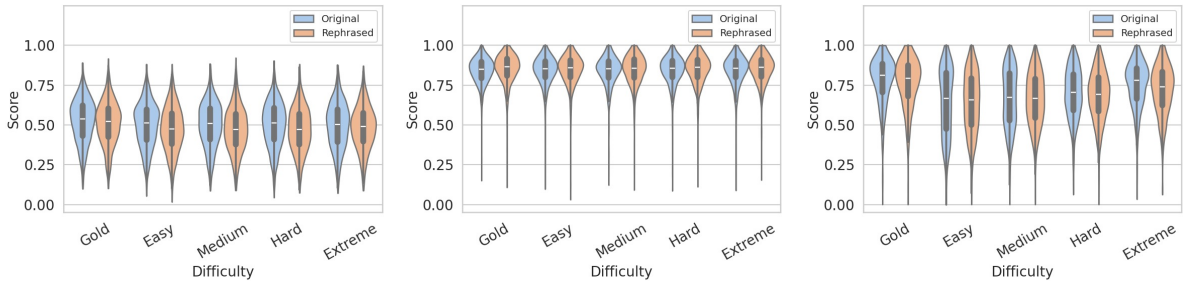
As discussed in Section 4, current fine-grained evaluation methods are insufficiently equipped to detect misinformation tactics like montage-style lies. To address the limitations, we propose DOVESCORE, a novel fine-grained evaluation framework designed to enable a comprehensive and nuanced assessment of information alignment.

5.1 Method

As illustrated in Figure 3, the DOVESCORE framework consists of three core components: the *Decomposer*, the *Fact Checker*, and the *Sorter*.

Decomposer Unlike conventional methods that uniformly segment text into sentences or atomic facts, DOVESCORE accounts for the inherent heterogeneity among factual elements. Specifically, we distinguish between two categories of facts: descriptive facts and event facts. Descriptive facts convey stable, order-independent attributes (e.g., "Octopuses have three hearts"), while event facts denote temporally ordered actions or states (e.g., "Dr. Lin submitted her resignation"). Based on this taxonomy, the decomposer partitions the target text into two lists: the event facts list (F_E) and the descriptive facts list (F_D).

Fact Checker The fact checker verifies each fact in F_E and F_D against the source text, resulting in two validated subsets: the set of correct event facts (F_E^c) and the set of correct descriptive facts (F_D^c). The Event Score is then computed as $S_E = \frac{|F_E^c|}{|F_E|}$, and the Descriptive Score as $S_D = \frac{|F_D^c|}{|F_D|}$.



(a) AlignScore (b) FactScore (c) DoveScore (ours)

Figure 4: Score Distribution Comparison of Fine-grained Evaluators. SummaC exhibits a similar pattern to AlignScore, assigning low scores to both correct and wrong target texts (See Appendix C).

Sorter The sorter reorganizes the verified correct event facts F_E^c into two ordered sequences. The first sequence, denoted $\text{Sorted}(F_E^c, s)$, reflects their chronological order in the source text, while the second, $\text{Sorted}(F_E^c, t)$, reflects their order in the target text. The similarity between these sequences is measured by the Event Order Score, defined as $S_{EO} = 1 - \text{ShuffleD}\left(\text{Sorted}(F_E^c, s), \text{Sorted}(F_E^c, t)\right)$.

Score Computation The final DoveScore is computed by combining S_E, S_{EO}, S_D using a frequency-based weighting factor $\alpha = \frac{|F_E|}{|F_E|+|F_D|}$, which adjusts the relative importance of event and descriptive facts according to their frequency in the target text:

$$\text{DoveScore} = \alpha \cdot S_E \cdot S_{EO} + (1 - \alpha) \cdot S_D \quad (5)$$

5.2 Experiment Setup

For the decomposer, fact checker, and sorter used in this experiment, we prompt LLM with the prompts as shown in Table 7, Table 8 and Table 9. We exampled with `gpt-4o-mini-2024-07-18` as the LLM with temperature 0.

5.3 Results

As shown in Table 3, DOVESCORE achieves the highest average AUC-ROC score of 65.25%, outperforming existing fine-grained evaluators by over 8%. Compared to FactScore, which uses the same LLM backbone, DOVESCORE improves performance by 14.8%. The distribution of its sub-scores (S_E, S_{EO}, S_D) in Figure 10 (Appendix C) highlights the contribution of S_{EO} in enhancing the model's ability to distinguish between truthful and deceptive texts.

	Easy	Medium	Hard	Extreme	AVG
Coarse-Grained	68.77	66.39	63.17	**59.24**	64.23
Fine-Grained	56.54	56.89	56.82	56.30	56.63
DoveScore (Fine-Grained)	**69.06**	**68.27**	**65.80**	57.87	**65.25**

Table 4: ROC-AUC of DOVESCORE on MONTAGELIE, compared to top scores by existing evaluators at each difficulty level. **Bold** marks the best, underlined the second best.

Further evidence from Figure 4 reveals systematic differences in how evaluators handle complex deception styles. Sentence-split-based methods such as SummaC and AlignScore show limited discrimination, often assigning similarly low scores to both correct and incorrect targets—likely due to inference ambiguity from rigid segmentation. FactScore, which evaluates at the fact level, tends to assign uniformly high scores across targets, ignoring inter-fact coherence. In contrast, DOVESCORE consistently assigns higher scores to correct targets and lower scores to deceptive ones, reflecting stronger discrimination capabilities and robustness to diverse misinformation strategies.

6 Conclusion

In this work, we present MONTAGELIE, a novel benchmark designed to reveal a critical vulnerability in current information alignment evaluators: their inability to detect misleading narratives composed of reordered yet truthful statements. We show that both coarse-grained and fine-grained methods struggle with such manip-

ulations, with AUC-ROC scores falling below 65% on MONTAGELIE. We propose DOVESCORE, a fine-grained evaluation framework that jointly considers factual accuracy and event order consistency, improving performance from 50.45% to 65.25% over FactScore. DOVESCORE is designed as a modular framework, allowing each component to be independently refined and improved. Among these, the sorter stands out as a critical and currently underexplored component that deserves targeted research efforts. While substantial room for further exploration remains, our work marks an important step toward more robust alignment evaluation for long-form content.

As for practical applications, DOVESCORE can be used in media fact-checking platforms (e.g., FactCheck.org). For example, for news reports on protests/accidents, DOVESCORE can flag "reordered events" (e.g., "Police action → Protest" vs. "Protest → Police action") to prevent causal misinformation. In general, DOVESCORE is useful for quality control of LLM-generated content. For example, DOVESCORE can be embedded in AI writing tools (e.g., academic summary tools): for medical reports (e.g., "Medication → Symptom relief" must not be reversed) or legal testimonies (event order is critical for liability judgment), DOVESCORE acts as a "post-generate filter" to block misleading content.

Limitations

MontageLie Benchmark The MontageLie benchmark used in our study is entirely generated by large language models rather than written by human annotators. While this approach enables scalable and diverse data generation, it may introduce distributional artifacts or stylistic patterns that do not fully reflect real-world human-written misinformation. Moreover, the benchmark currently only covers English. Extending MontageLie to include human-curated data and multilingual variants would improve its generalizability and practical relevance.

DoveScore Framework In this work, we demonstrate DoveScore using GPT-4o-mini as the backbone model. However, DoveScore is designed as a modular and model-agnostic framework—each component (e.g., evidence extractor, fact scorer, and sorter) can be flexibly instantiated with different language models. Future work could explore alternative backbones to assess the robustness and adaptability of the framework under varying resource constraints and capabilities. Within DoveScore, our current sorter module takes the full list of candidate responses as input and predicts a globally reordered list. This design is chosen over pairwise comparison-based sorting methods to reduce computational complexity. However, the trade-off between efficiency and ranking accuracy remains an open research question. More sophisticated or hybrid sorting strategies may offer better performance while maintaining tractable runtime.

Ethics Statement

All data used in this study are derived from publicly available datasets and do not contain any personally identifiable or sensitive information. The additional data used for the MontageLie benchmark were generated using LLMs. To ensure the quality of the generated data, we conducted a manual evaluation with two human annotators: one is one of the authors of this paper, and the other is an external contributor who received compensation at the standard hourly rate designated for tutors and demonstrators at our university.

Acknowledgments

This work is supported by Huawei's Dean's Funding (C-00006589) and the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1).

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Beizhe Hu, Qiang Sheng, Juan Cao, Yang Li, and Danding Wang. 2025. Llm-generated fake news induces truth decay in news ecosystem: A case study on neural news recommendation. *arXiv preprint arXiv:2504.20013*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Wenyu Huang, Pavlos Vougiouklis, Mirella Lapata, and Jeff Z. Pan. 2025b. Masking in Multi-hop QA: An Analysis of How Language Models Perform with Context Permutation. In *In the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.

Wenyu Huang, Guancheng Zhou, Mirella Lapata, Pavlos Vougiouklis, Sebastien Montella, and Jeff Z Pan. 2025c. Prompting Large Language Models with Knowledge Graphs for Question Answering involving Long-tail Facts. *Knowledge-Based Systems (KBS)*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Jinhwa Kim, Ali Derakhshan, and Ian Harris. 2024. Robust safety classifier against jailbreaking attacks: Adversarial prompt shield. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 159–170, Mexico City, Mexico. Association for Computational Linguistics.

Donald E Knuth. 1998. *The Art of Computer Programming: Sorting and Searching, volume 3*. Addison-Wesley Professional.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Juncai Li, Ru Li, Xiaoli Li, Qinghua Chai, and Jeff Z. Pan. 2024. Inference Helps PLMs Conceptual Understanding: Improving the Abstract Inference Ability with Hierarchical Conceptual Entailment Graphs. In *In Proc. of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*, pages 22088–22104.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, and 1 others. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023a. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization. *arXiv preprint arXiv:2303.15621*.

Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, Shu Wu, and Liang Wang. 2024. EX-FEVER: A dataset for multi-hop explainable fact verification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9340–9353, Bangkok, Thailand. Association for Computational Linguistics.

Dominik Macko, Aashish Anantha Ramakrishnan, Jason Samuel Lucas, Robert Moro, Ivan Srba, Adaku Uchendu, and Dongwon Lee. 2025. Beyond speculation: Measuring the growing presence of llm-generated texts in multilingual disinformation. *arXiv preprint arXiv:2503.23242*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfcheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Chris Mellish and Jeff Z. Pan. 2008. Natural language directed inference from ontologies. *Artif. Intell.*, 172(10):1285–1315.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. 2023. Large Language Models and Knowledge Graphs: Opportunities and Challenges. *Transactions on Graph Data and Knowledge*, pages 1–38.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nirmal Roy, Leonardo F. R. Ribeiro, Rexhina Blloshmi, and Kevin Small. 2024. Learning when to retrieve, what to rewrite, and how to respond in conversational QA. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10604–10625, Miami, Florida, USA. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeritec: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. 2024. CHECKWHY: Causal fact verification via argument structure. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15636–15659, Bangkok, Thailand. Association for Computational Linguistics.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024a. FineSurE: Fine-grained summarization evaluation using LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922, Bangkok, Thailand. Association for Computational Linguistics.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024b. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings*

of the Association for Computational Linguistics: EMNLP 2024, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hongru Wang, Deng Cai, Wanjun Zhong, Shijue Huang, Jeff Z. Pan, Zeming Liu, and Kam-Fai Wong. 2025. Self-Reasoning Language Models: Unfold Hidden Reasoning Chains with Few Reasoning Catalyst. In *In the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Zixia Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V Le. 2024. Long-form factuality in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhichao Yan, Jiapu Wang, Jiaoyan Chen, Xiaoli Li, Ru Li, and Jeff Z. Pan. 2025a. Decomposing and Revising What Language Models Generate. In *In the Proceedings of the 28th European Conference on Artificial Intelligence (ECAI-2025)*.

Zhichao Yan, Jiapu Wang, Jiaoyan Chen, Xiaoli Li, Ru Li, and Jeff Z.Pan. 2025b. Atomic Fact Decomposition Helps Attributed Question Answering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Junjie Ye, Guanyu Li, SongYang Gao, Caishuang Huang, Yilong Wu, Sixian Li, Xiaoran Fan, Shihan Dou, Tao Ji, Qi Zhang, Tao Gui, and Xuanjing Huang. 2025. ToolEyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 156–187, Abu Dhabi, UAE. Association for Computational Linguistics.

Simon Yu, Jie He, Pasquale Minervini, and Jeff Z. Pan. 2024. Evaluating the Adversarial Robustness of Retrieval-Based In-Context Learning for Large Language Models. In *In Proc. of the Conference on Language Models (COLM 2024)*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wen Zhang, Jiaoyan Chen, Juan Li, Zezhong Xu, Jeff Z. Pan, and Huajun Chen. 2024a. Knowledge graph reasoning with logics and embeddings: Survey and perspective. In *ICKG*, pages 492–499. IEEE.

Yue Zhang, Jingxuan Zuo, and Liqiang Jing. 2024b. Fine-grained and explainable factuality evaluation for multimodal summarization. *arXiv preprint arXiv:2402.11414*.

Danna Zheng, Danyang Liu, Mirella Lapata, and Jeff Z. Pan. 2024. TrustScore: Reference-Free Evaluation of LLM Response Trustworthiness. In *In Proc. of the ICLR 2024 Workshop on Secure and Trustworthy Large Language Models (Set LLM 2024)*.

A Prompts

A.1 Prompts Used in Data Construction

The prompt used in MONTAGELIE construction is provided in Table 5.

A.2 Prompts Used in LLM-based Evaluators

The prompts used in coarse-grained LLM-as-evaluator is provided in Table 6. The prompt used in DoveScore decomposer, fact checker, and sorter are provided in Table 7, 8, and 9 respectively.

B About MONTAGELIE

B.1 Shuffle Algorithm Used in Montage-Style Lie Generation

Algorithm 1 shows that how to obtain a random shuffled list given a list and a target inversion count. To generate a random shuffled list with a given number of inversions, the algorithm begins by assigning the maximum possible inversions to each position, assuming the list is fully reversed. It then randomly decreases these inversion values until the total number of inversions equals the target. This creates an inversion sequence that still respects the desired count but adds randomness. Finally, it builds the shuffled list by selecting elements from a sorted version of the original list, guided by the inversion values, ensuring the resulting permutation has exactly the specified number of inversions.

B.2 Data Distribution

The distribution of the decomposed event number during MONTAGELIE construction, the sampled inversion count when shuffling event lists, and the source and target text length are shown in Figure 6, 7, and 5.

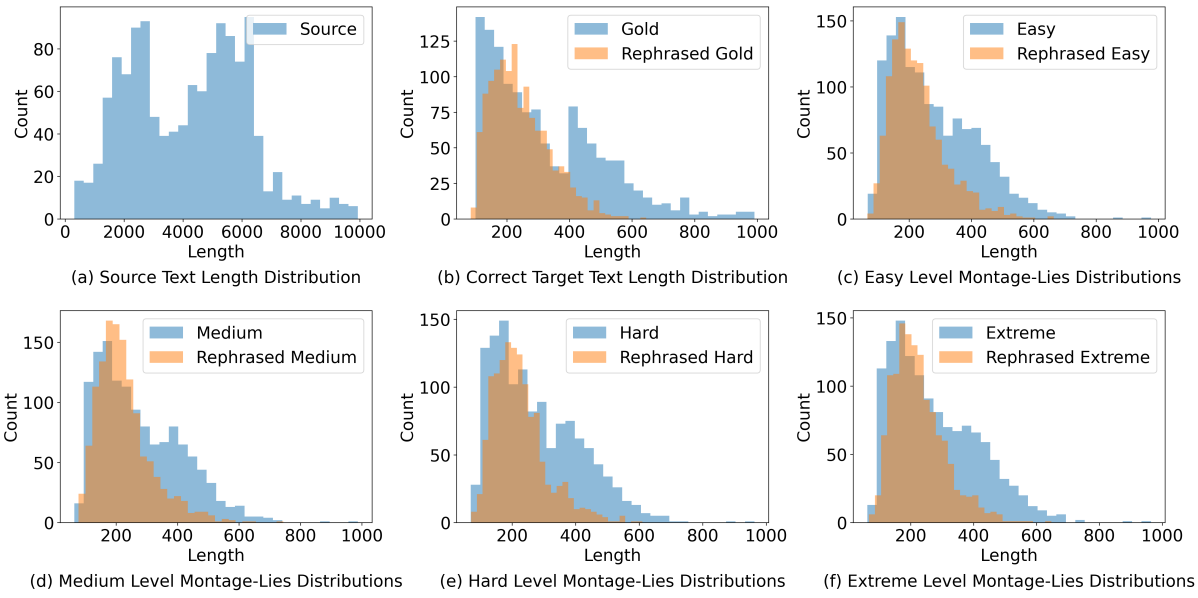
C Score Distribution on MONTAGELIE

The score distribution of coarse-grained evaluators , fine-grained evaluators are shown in Figure 8 and 9. The subscore distribution of DoveScore is provided in Figure 10.

Algorithm 1: RANDOMSHUFFLEWITHINVERSIONS

Input : $A = (a_1, a_2, \ldots, a_m)$: original list of length m
K: target inversion count, $0 \leq K \leq \frac{m(m-1)}{2}$
Output : B: a random permutation of A with $\mathrm{Inv}(B) = K$

/ Step 1: Initialize maximum inversion sequence */*
$m \leftarrow |A|$;
for $i \leftarrow 1$ **to** m **do**
  $e_i \leftarrow m - i$; */* Maximum possible inversions at position i */*
/ Step 2: Reduce total inversions to desired K */*
$\Delta \leftarrow \frac{m(m-1)}{2} - K$
while $\Delta > 0$ **do**
  $i \leftarrow \mathrm{UniformRandomInteger}(1, m)$;
  if $e_i > 0$ **then**
    $e_i \leftarrow e_i - 1$;
    $\Delta \leftarrow \Delta - 1$
/ Step 3: Decode inversion sequence into permutation */*
$C \leftarrow$ sorted copy of A;
$B \leftarrow$ empty list of size m
for $i \leftarrow 1$ **to** m **do**
  $B_i \leftarrow C[e_i + 1]$; */* Choose the $(e_i + 1)$-th smallest available element */*
  Remove $C[e_i + 1]$ from C
return B



(a) Source Text Length Distribution (b) Correct Target Text Length Distribution (c) Easy Level Montage-Lies Distributions

(d) Medium Level Montage-Lies Distributions (e) Hard Level Montage-Lies Distributions (f) Extreme Level Montage-Lies Distributions

Figure 5: Distribution of words length in MontageLie benchmark.

Prompt used in step 1: Decompose g into Events E

Break down the following paragraph into a list of independent events, listed in chronological order. Resolve all pronouns and referring expressions to their corresponding specific entities. Output only the event list and nothing else.

{{Paragraph}}

Prompt used in step 3: Incremental Lie Generation

Here is what has happened so far:
{{CurrentParagraph}}
The following new fact occurred after the events described above:
{{Event}}
Please append this new fact directly to the current paragraph. If the addition feels awkward, make only minimal word adjustments to ensure the paragraph flows smoothly—without adding extra narrative details or transitional phrases such as "next" or "following that." Output only the updated paragraph.

Prompt used in rephrasing with different narrative technic

Task: Rephrase
Rephrase a given paragraph by applying a different narrative sequencing technique. Follow the steps below carefully:

Step 1: Identify the Original Narrative Technique Read the original paragraph and determine which of the following sequencing techniques it uses:

- **Chronological Order** – Events are presented strictly in the order they occurred.
- **Flashback** – The paragraph begins with a later or climactic moment, then shifts back to earlier events.
- **Interjection** – The main narrative is interrupted by a relevant insert such as a memory, reflection, or side story.
- **Supplementary Narration** – Contextual background is added to support understanding, even if the details weren't part of the original sequence.

Step 2: Rephrase Using a Different Technique Choose a different narrative sequencing method from the list above and rephrase the paragraph accordingly.

Guidelines for Rephrasing:

- Use as much of the original wording as possible.
- Do **not** add any new events or fabricate details not present in the original.
- Avoid ambiguous expression.

Please output the result in the following format:

- Original_Narrative_Technique: <original_narrative_technique>
- Choosed_Narrative_Technique: <choosed_narrative_technique>
- Rephrased: <rephrased_paragraph>

{{Paragraph}}

Table 5: Prompts used in the data construction process

Prompt used in coarse-grained LLM-as-Evaluators

```
You will be given a source text.  You will then be given one target text to be evaluated.
Your task is to rate the information alignment of target text against the source text.  Please
make sure you read and understand these instructions carefully. Please keep this source text open
while reviewing, and refer to it as needed.

Evaluation Criteria:
Consistency (1-5) - the information alignment between the target text and the source text.  A
consistent target text contains only statements that are entailed by the source source text.
Annotators were also asked to penalize target texts that contained hallucinated facts. 1 - worst,
5 - best.

Evaluation Steps:
1. Read the source text carefully and identify the main facts and details it presents.
2. Read the target text and compare it to the source text. Check if the target text contains any
factual errors that are not supported by the source text.
4. Assign a score for consistency based on the Evaluation Criteria.

Note: only output the score for consistency, no other text.

Source Text:
{{Source}}

Target Text:
{{Target}}

Evaluation Form (scores ONLY):
- Consistency:
```

Table 6: The prompts used in benchmarking coarse-grained LLM-as-Evaluators.

Prompt used in Decomposer

```
Please analyze the following paragraph and extract all independent factual statements, categorized
into two types: Event Facts and Descriptive Facts.

## Definitions:
- Event Facts:
Time-dependent facts that describe specific actions, changes, occurrences, or emotional/mental
states.  These involve entities doing something or experiencing something dynamically at a
particular point in time, and can be situated along a timeline.
Examples:
The spacecraft entered Mars' orbit after a six-month journey.
Dr. Lin submitted her resignation.
Mary felt happy about her promotion.

- Descriptive Facts:
Time-independent facts that define, classify, or describe static attributes or relationships of
entities. These do not occur at a specific time, and are considered stable or inherent properties.
Examples:
Helianthus is a genus in the daisy family Asteraceae.
Octopuses have three hearts.

## Instructions:
1. Break down the paragraph into individual, self-contained factual statements.
2. Resolve all pronouns and referring expressions to their full entity names for clarity.
3. Categorize each fact as either an Event Fact or a Descriptive Fact, according to the definitions
above.
3. Output two separate lists:
- Event Facts List: List in chronological order.
- Descriptive Facts List: Order does not matter.

Paragraph:  {{Paragraph}}

Event Facts List and Descriptive Facts List:
```

Table 7: The prompts used in decomposer of DoveScore.

Prompt used in fact checker

```
Check if the fact is true based on the given context. Return True or False.

Context:  {{Source}}

Fact:  {{Fact}} True or False?

Output:
```

Table 8: The prompts used in fact checker of DoveScore.

Prompt used in sorter

```
You are a helpful assistant that determines the correct chronological order of events in a
paragraph. Do NOT add, remove, or change any events. Only reorder the exact events from the input
list.

Example 1:
Paragraph:
Tom woke up early. He brushed his teeth and then had breakfast. After that, he went for a run.
Events:
- Tom had breakfast
- Tom woke up
- Tom went for a run
- Tom brushed his teeth
Ordered Events:
[Tom woke up, Tom brushed his teeth, Tom had breakfast, Tom went for a run]

Example 2:
Paragraph:
After she went out for lunch, Sarah called her friend. Earlier in the morning, she had replied to
a message right after checking her email.
Events:
- Sarah checked her email
- Sarah went out for lunch
- Sarah called her friend
- Sarah replied to a message
Ordered Events:
[Sarah checked her email, Sarah replied to a message, Sarah went out for lunch, Sarah called her
friend]

Now sort the following events based on the paragraph below, and return as a list of events:

Paragraph:  {{Paragraph}}

Events:  {{Events}}

Ordered Events:
```

Table 9: The prompts used in sorter of DoveScore.

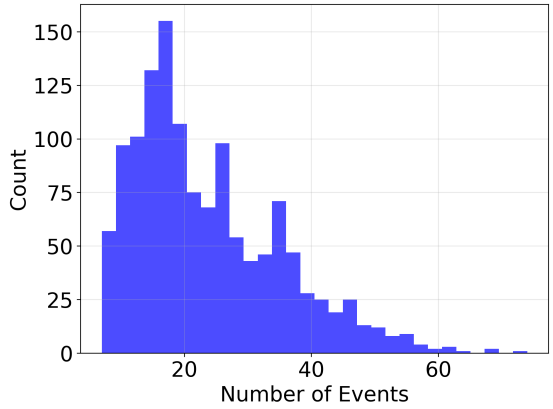


Figure 6: Distribution of number of event decomposed in Step 1 of Montage-Style Lie Generation: Decompose g into Events E.

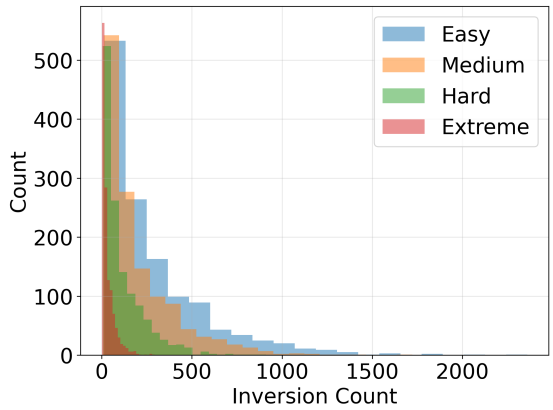
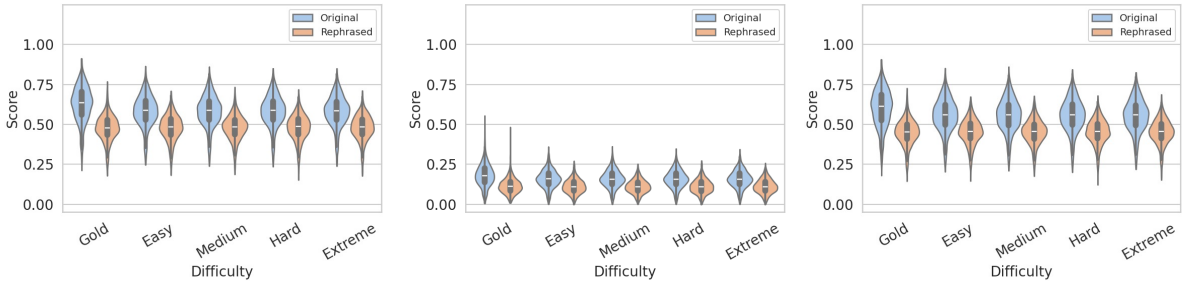
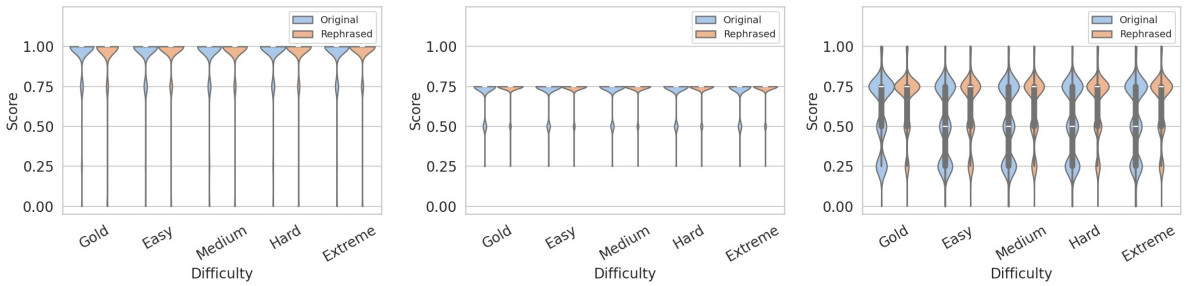


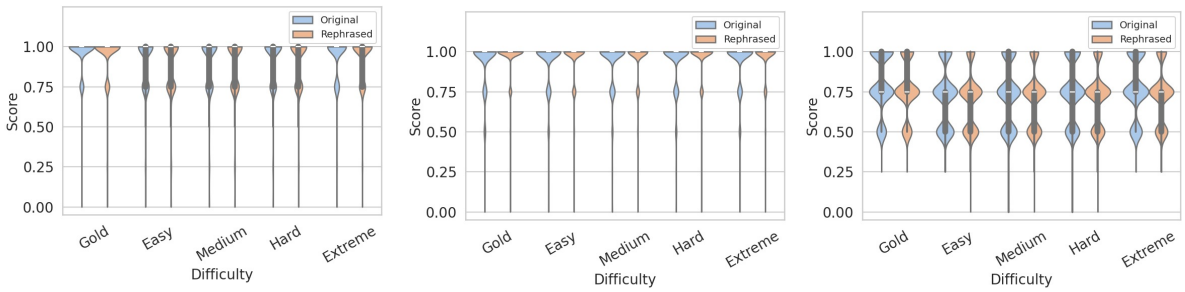
Figure 7: Distribution of Inversion Count Sampled in Step 2 of Montage-Style Lie Generation: Shuffle E with Controlled Difficulty.



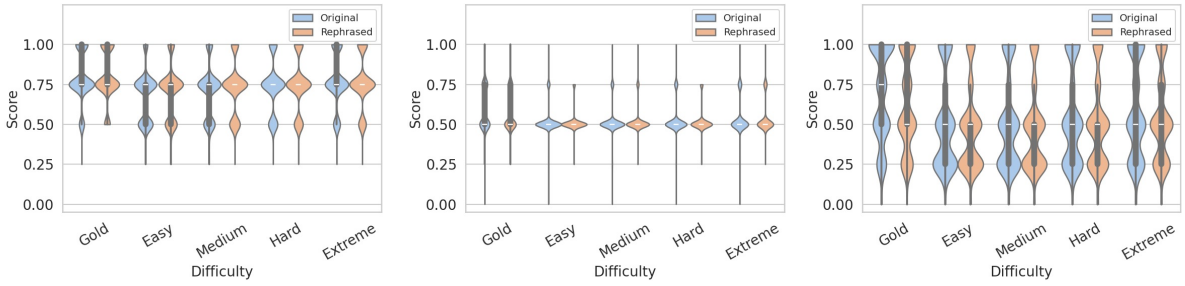
(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-L



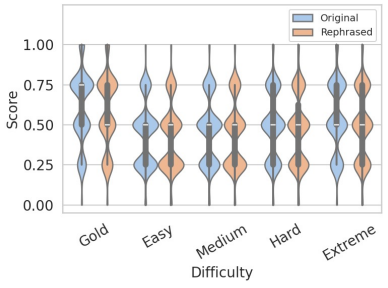
(d) Llama-3.2-instruct-1B (e) Llama-3.2-instruct-3B (f) Llama-3.1-instruct-8B



(g) Llama-3.3-instruct-70B (h) Qwen-3-1.7B (i) Qwen-3-4B

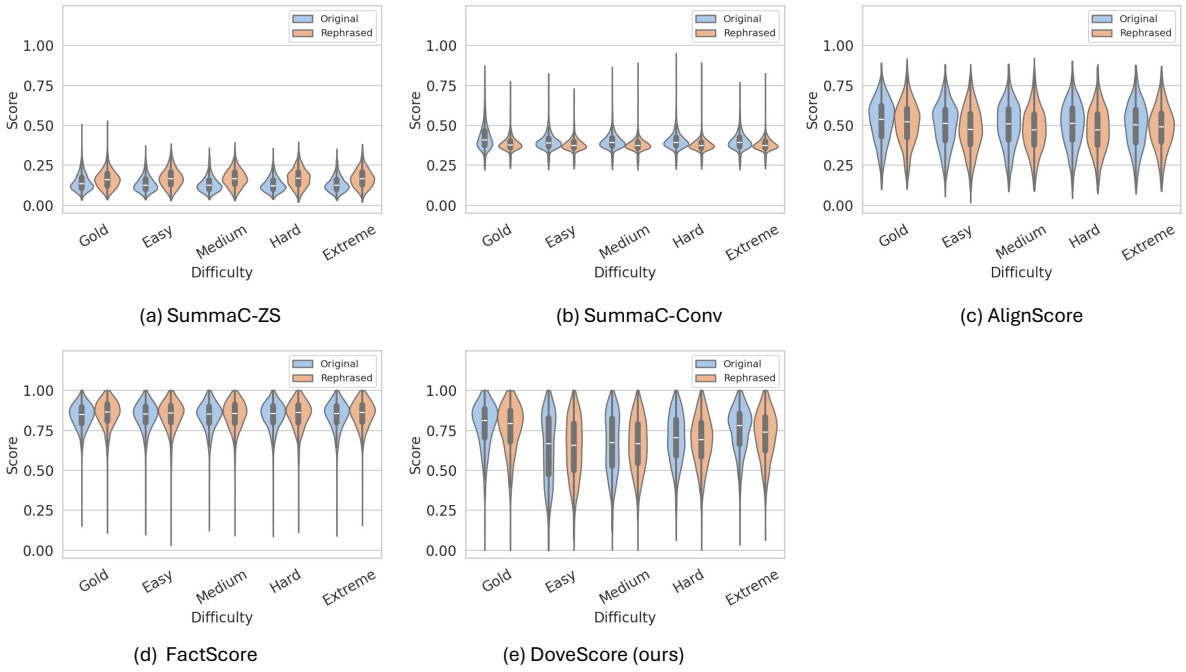


(j) Qwen-3-8B (k) Qwen-3-14B (l) Qwen-3-32B



(m) gpt-4o-mini

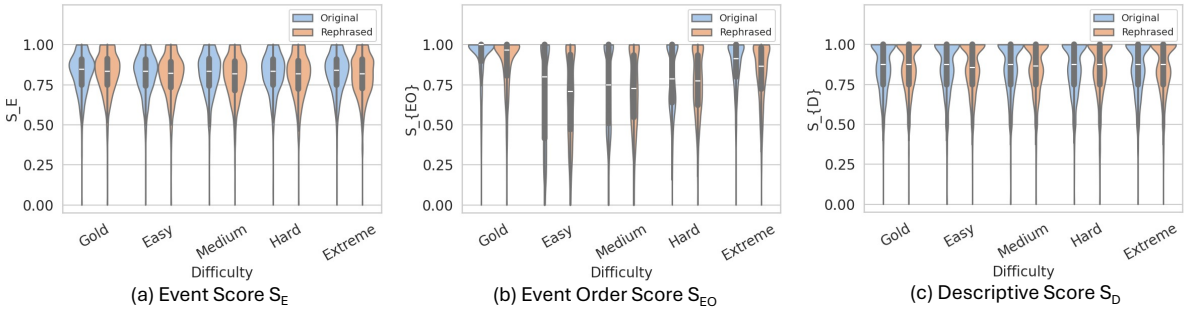
Figure 8: Violin Plots of Score Obtained By Coarse-Grained Evaluators on MONTAGELIE



(a) SummaC-ZS (b) SummaC-Conv (c) AlignScore

(d) FactScore (e) DoveScore (ours)

Figure 9: Violin Plots of Score Obtained By Fine-Grained Evaluators on MONTAGELIE



(a) Event Score S_E (b) Event Order Score S_{EO} (c) Descriptive Score S_D

Figure 10: Violin Plots of SubScores obtained by DOVESCORE on MONTAGELIE.