CoreEval: Automatically Building Contamination-Resilient Datasets with Real-World Knowledge toward Reliable LLM Evaluation

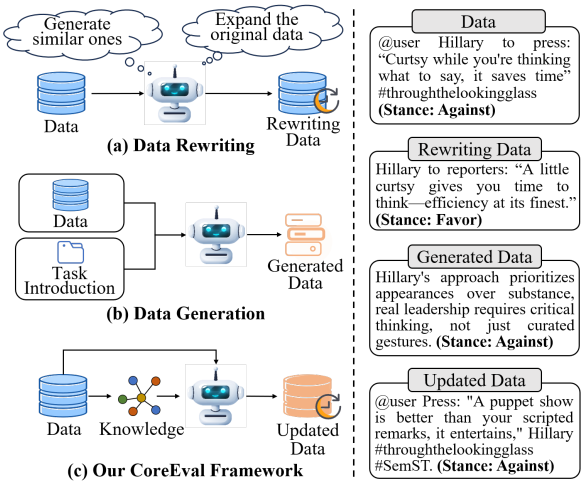
Jingqian Zhao[1*], **Bingbing Wang**[1*], **Geng Tu**[1], **Yice Zhang**[1], **Qianlong Wang**[1],
Bin Liang[4†], **Jing Li**[5], **Ruifeng Xu**[1,2,3†]
[1] Harbin Institute of Technology, Shenzhen, China [2] Peng Cheng Laboratory, Shenzhen, China
[3] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[4] The Chinese University of Hong Kong, Hong Kong, China
[5] The Hong Kong Polytechnic University, Hong Kong, China
{zhaojingqian, bingbing.wang}@stu.hit.edu.cn, xuruifeng@hit.edu.cn

Abstract

Data contamination poses a significant challenge to the fairness of LLM evaluations in natural language processing tasks by inadvertently exposing models to test data during training. Current studies attempt to mitigate this issue by modifying existing datasets or generating new ones from freshly collected information. However, these methods fall short of ensuring contamination-resilient evaluation, as they fail to fully eliminate pre-existing knowledge from models or preserve the semantic complexity of the original datasets. To address these limitations, we propose **CoreEval**, a **Co**ntamination-**re**silient **Eval**uation strategy for automatically updating data with real-world knowledge. This approach begins by extracting entity relationships from the original data and leveraging the GDELT database to retrieve relevant, up-to-date knowledge. The retrieved knowledge is then recontextualized and integrated with the original data, which is refined and restructured to ensure semantic coherence and enhanced task relevance. Ultimately, a robust data reflection mechanism is employed to iteratively verify and refine labels, ensuring consistency between the updated and original datasets. Extensive experiments on updated datasets validate the robustness of CoreEval, demonstrating its effectiveness in mitigating performance overestimation caused by data contamination.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated exceptional performance across a wide range of Natural Language Processing (NLP) tasks (Li et al., 2024a; Ma et al., 2024). Publicly available datasets serve as standardized benchmarks for evaluating model performance, ensuring consistency and reproducibility in assessments. However, the static and public nature of these datasets poses a significant challenge: **data contamination**, where test data may inadvertently appear in the training sets of newer LLMs. This contamination can artificially inflate model performance, compromising the reliability of LLM evaluations (Banerjee et al., 2024; Li et al., 2024c).



@user Hillary to press: “Curtsy while you're thinking what to say, it saves time” #throughthelookingglass **(Stance: Against)**

Hillary to reporters: “A little curtsy gives you time to think—efficiency at its finest.” **(Stance: Favor)**

Hillary's approach prioritizes appearances over substance, real leadership requires critical thinking, not just curated gestures. **(Stance: Against)**

@user Press: "A puppet show is better than your scripted remarks, it entertains," Hillary #throughthelookingglass #SemST. **(Stance: Against)**

Figure 1: Different workflows for mitigating data contamination: (a) Data Rewriting, where LLMs modify existing data, potentially altering original labels; (b) Data Generation, where LLMs create new data from original data and task instructions, risking loss of semantic complexity; and (c) Our CoreEval Framework, where LLMs integrate external knowledge with original data for robust, semantically coherent, and label-consistent updates.

To mitigate data contamination, curating new datasets has become a widely adopted approach. Recently, researchers have explored automated dataset construction methods to reduce the time and labor costs associated with manual curation (Ying et al., 2024). These approaches using LLMs can be broadly categorized into two types: data rewriting, which modifies existing data while preserving its original structure, and data generation, which leverages newly collected data to create task-specific datasets (Li et al., 2024b; Wu et al., 2024).

[*] The first two authors contribute equally to this work.
[†] Corresponding Author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22284–22306
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Despite their widespread adoption, these methods have significant limitations. As illustrated in Figure 1 (a), **data rewriting** employs prompt-based instructions to guide LLMs in modifying existing data. While this approach is straightforward, it often risks generating data with labels that deviate from the original annotations. Additionally, the rewriting process may inadvertently introduce contaminated data, as models could rely on pre-existing information from their training corpus. On the other hand, **data generation**, which directly produces new datasets based on data and task introduction, shown in Figure 1 (b), fails to preserve the semantic richness and complexity of the original dataset, leading to information loss. These limitations undermine the reliability and effectiveness of existing approaches for contamination-resilient evaluation.

Therefore, this paper introduces **CoreEval**, a framework designed to mitigate data contamination and enable reliable, up-to-date LLM evaluation. As illustrated in Figure 1, CoreEval goes beyond simple data rewriting and generation. Instead, it systematically integrates newly acquired knowledge, preserving data quality, enhancing robustness, and maintaining semantic richness while ensuring alignment with task objectives. Specifically, CoreEval first extracts entity relationships from the original data and utilizes the Global Database of Events, Language, and Tone (GDELT) Project to retrieve up-to-date, real-world knowledge. This knowledge is then recontextualized with original data to refine and restructure the dataset, ensuring semantic coherence and alignment with task objectives. Finally, a rigorous data reflection mechanism enforces label consistency and preserves dataset integrity. We systematically evaluate CoreEval on multiple NLP datasets across different LLMs. Extensive experiments on these updated datasets validate the stability of our framework, demonstrating that CoreEval not only upholds high data quality but also effectively mitigates performance overestimation caused by data contamination. The contributions of this paper can be summarized as follows:

- We propose CoreEval, an automatic contamination-resilient evaluation strategy that integrates real-world knowledge to update datasets.
- We design a structured workflow inspired by cognitive learning theory to ensure reliable and timely LLM evaluation.
- Extensive experiments across multiple tasks and a series of LLMs demonstrate the effectiveness of CoreEval in mitigating data contamination.

2 Related Works

2.1 Data Contamination

Many datasets are widely used to evaluate models in NLP tasks like sentiment analysis (Saif et al., 2013; Rogers et al., 2018), stance detection (Li et al., 2021; Glandt et al., 2021), and emotion classification (Chen et al., 2017). With LLMs, it is often assumed that a more advanced base model yields superior performance (Pathak and Rana, 2024). However, despite their critical role in benchmarking, the lack of transparency regarding the training data of these models makes it challenging for researchers to verify whether a given model has been contaminated by specific datasets.

Recent studies have explored data contamination in the evaluation of LLM. Aiyappa et al. (2023) analyzed ChatGPT's stance detection, highlighting risks associated with its closed nature and updates. Li et al. (2024c) reported contamination rates from 1% to 45% across six Question Answering (QA) benchmarks. To tackle these challenges, researchers have explored methods for detecting contamination, revealing the limitations of string-matching techniques like n-gram overlap (Yang et al., 2023; Jiang et al., 2024a; Ippolito et al., 2023). Simple test variations, such as paraphrasing, can bypass these methods, allowing even a 13B model to overfit benchmarks and perform comparably to GPT-4. Dekoninck et al. (2024a) further emphasized these issues with the introduction of Evasive Augmentation Learning (EAL).

2.2 Contamination-Resilient Method

To achieve contamination-resilient evaluation, updating datasets by collecting new data is an intuitive solution. However, due to the time-consuming and labor-intensive nature of this process, automatic update methods have emerged (Wu et al., 2024). These methods primarily fall into two categories: data rewriting and data generation.

Data rewriting modifies existing data to generate updated versions. Ying et al. (2024) proposed two strategies: mimicking, which preserves style and context, ensuring consistency, and extending, which introduces varied difficulty to broaden the dataset's cognitive scope. Data generation relies on

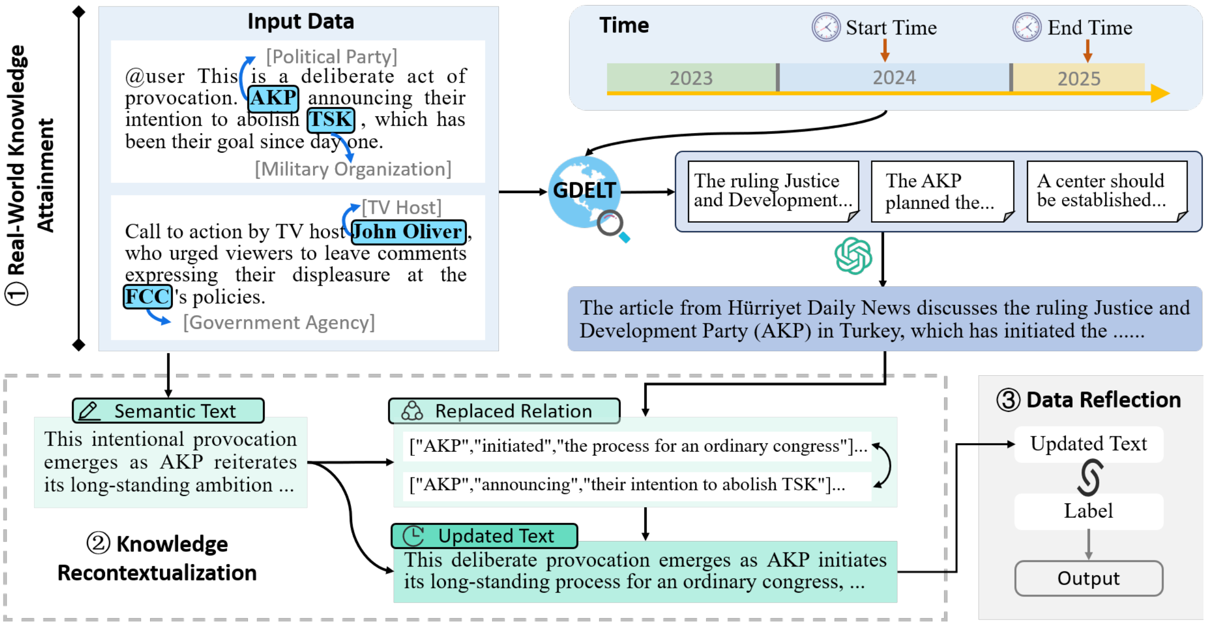

① Real-World Knowledge Attainment
Input Data
[Political Party]
@user This is a deliberate act of provocation. AKP announcing their intention to abolish TSK, which has been their goal since day one.
[Military Organization]
[TV Host]
Call to action by TV host John Oliver, who urged viewers to leave comments expressing their displeasure at the FCC's policies.
[Government Agency]
Time
Start Time
End Time
2023
2024
2025
GDELT
The ruling Justice and Development...
The AKP planned the...
A center should be established...
The article from Hürriyet Daily News discusses the ruling Justice and Development Party (AKP) in Turkey, which has initiated the ......
Semantic Text
This intentional provocation emerges as AKP reiterates its long-standing ambition ...
Replaced Relation
["AKP","initiated","the process for an ordinary congress"]...
["AKP","announcing","their intention to abolish TSK"]...
Updated Text
This deliberate provocation emerges as AKP initiates its long-standing process for an ordinary congress, ...
② Knowledge Recontextualization
③ Data Reflection
Updated Text
Label
Output

Figure 2: Overall flow of our CoreEval framework.

newly collected data to build task-specific datasets. LatestEval (Li et al., 2024b) ensures integrity by using texts from recent sources, avoiding overlaps with pre-trained corpora. Similarly, LiveBench (White et al., 2024) creates novel datasets by extracting challenges from up-to-date sources like math competitions, arXiv papers, news articles, and transforming them into more challenging, contamination-free versions. Despite their innovations, these methods have limitations. Data rewriting may produce inconsistent labels and introduce contamination from model biases, while data generation often fails to fully capture the semantic depth of the original dataset, leading to information loss. These challenges reduce the reliability and practicality of datasets for contamination-resilient evaluations. Unlike these studies, CoreEval combines structured knowledge retrieval, semantic recontextualization, and iterative label verification to ensure dataset quality and robustness. By utilizing real-world updates and a reflection mechanism, CoreEval mitigates contamination while preserving semantic complexity.

3 CoreEval Framework

3.1 Preliminary

In this section, we introduce our novel CoreEval framework, inspired by Bruner's cognitive theory, for constructing contamination-resilient datasets that integrate real-world knowledge. Building upon Bruner's cognitive learning theory (Bruner, 2009), we assert that the essence of learning lies in the active formation of cognitive structures rather than the passive absorption of information. Learners actively construct their own knowledge systems by synthesizing newly acquired knowledge with their existing cognitive frameworks. Learning is conceptualized as involving three nearly simultaneous processes: the acquisition of information, the transformation of information, and its subsequent evaluation. As shown in Figure 2, we organize these processes into three components to better align with LLM evaluation. **1) Real-World Knowledge Attainment** corresponds to information acquisition, collecting real-time knowledge from the GDELT database. **2) Knowledge Recontextualization** component handles information transformation, updating the dataset by incorporating new knowledge. **3) Data Reflection** component addresses the evaluation process by refining and assessing the data. This structure ensures that all learning processes are effectively integrated into a cohesive framework.

3.2 Real-World Knowledge Attainment

To incorporate real-world knowledge, we leverage GDELT (Leetaru and Schrodt, 2013), a comprehensive CAMEO-coded database containing over 200 million geolocated events spanning global coverage from 1979 to the present. Given a dataset $\mathcal{D} = \{(d_1, y_1), (d_2, y_2), ..., (d_n, y_n)\}$ consisting of n samples, where each sample d_i is paired with a corresponding label y_i from the label set $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$. The knowledge extraction process begins by identifying relevant entities from the data using LLM $\mathcal{M}$, where the input d_i acts as information cues for entity extraction.

$$E_i \leftarrow \mathcal{M}(d_i) \tag{1}$$

where $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,j_i}\}$ and j_i represents the set and number of entities extracted from d_i. These extracted entities form the foundation for subsequent knowledge retrieval. To efficiently query large-scale data, we utilize Google BigQuery[1] and the GDELT. BigQuery enables fast, scalable processing of vast datasets like GDELT, while the API facilitates seamless real-time data retrieval. A list of extracted entities is used to query GDELT databases $\mathcal{G}$ for data points within a specific time period to retrieve the most relevant and up-to-date knowledge. Then we employ LLM to summarize the knowledge to obtain. The overall retrieval process can be formalized as:

$$\begin{aligned} \mathcal{K}_i &\leftarrow \mathcal{G}(E_i, t_{\text{start}}, t_{\text{end}}) \\ \hat{\mathcal{K}}_i &\leftarrow \mathcal{M}(\mathcal{K}_i) \end{aligned} \tag{2}$$

where $\mathcal{K}_i$ indicates the knowledge retrieved from the GDELT database. $\hat{\mathcal{K}}_i$ represents the knowledge after being summarized by the LLM. t_{start} and t_{end} represent the start and end times for the query[2].

3.3 Knowledge Recontextualization

The knowledge recontextualization phase involves integrating new knowledge with existing cognitive structures, transforming it into a form suited for new tasks. During this phase, learners process and reorganize newly acquired knowledge to enhance both understanding and application. We begin by extracting relational triples from the original sentence d_i. These relational triples are represented as $T_i = \{\langle e_{i,j}, r_{i,j}, e^{'}_{i,j}\rangle \mid j = 1, 2, ..., l_i\}$, where $e_{i,j}$ and $e^{'}_{i,j}$ are entities, and $r_{i,j}$ denotes the relation between them. l_i is the number of relational triples extracted from d_i. Next, using new knowledge $\hat{\mathcal{K}}_i$ and an LLM $\mathcal{M}$, we update the original triples T_i by generating replacement triples $\hat{T}_i$. The updated sentence d_i^u is then derived by substituting the original triples with $\hat{T}_i$, as shown by:

$$\begin{aligned} \hat{T}_i &\leftarrow \mathcal{M}(T_i, \hat{\mathcal{K}}_i) \\ d_i^u &\leftarrow f(d_i, \hat{T}_i) \end{aligned} \tag{3}$$

where f is the replacement operation.

Furthermore, semantic rewriting is performed while preserving the T_i, resulting in:

$$d_i^s \leftarrow \mathcal{M}(d_i, T_i) \tag{4}$$

We leverage the semantic style of d_i^s combined with the label y_i to construct a **semantic dataset** $\mathcal{D}^s$.

The updated text $\hat{d}_i$ adopts the semantic style of d_i^s, preserving its linguistic characteristics while incorporating the triples of $\hat{T}_i$. Additionally, to maintain classification coherence, the label of $\hat{d}_i$ is kept consistent with that of the original sentence d_i. Formally, this process is represented as:

$$\hat{d}_i \leftarrow \mathcal{M}(d_i, d_i^u, \hat{T}_i, d_i^s) \tag{5}$$

The **updated dataset** $\hat{\mathcal{D}}$ is then formed by combining $\hat{d}_i$ with the corresponding label. This process ensures the systematic integration of new knowledge while maintaining the coherence and adaptability of the transformed content.

3.4 Data Reflection

To evaluate the quality of the generated text, we design an agent to reflect and perform evaluations. This evaluation process employs prompting (Wei et al., 2022) to facilitate step-by-step reasoning. The assessment focuses on two key criteria:

Incorrect Information: Evaluating whether the generated text accurately reflects the facts derived from the provided knowledge. Any discrepancies or inconsistencies are flagged for re-generation.

Label Alignment: Measuring the degree of alignment between the generated text and the corresponding ground truth label, ensuring consistency and relevance to the intended output.

The prompting allows the agent to iteratively reflect on these criteria, providing a rationale for its evaluation. Based on this reflection, the agent determines whether the text required to be regenerated to improve accuracy or alignment. Detailed prompts can be found in Appendix A.1.

3.5 Apply to Existing Datasets

We selected five representative Natural Language Understanding (NLU) tasks from the TweetEval Benchmark (Barbieri et al., 2020) and GLUE Benchmark (Wang, 2018), including Emotion Recognition (Mohammad et al., 2018), Irony Detection (Van Hee et al., 2018), Stance Detection (Mohammad et al., 2016), Microsoft Research

[1] https://cloud.google.com/bigquery

[2] We chose the release date of the latest open-source model as the starting point for retrieval to prevent overlap with the model's training data.

Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), and Recognizing Textual Entailment (RTE) (Wang, 2018), to apply our method for automatic updating and evaluation. Table 1 presents the statistical characteristics of these datasets. Notably, for the MRPC and RTE datasets, we refine the provided sentence pairs during the data reflection phase and ensure the supervision of label accuracy for improved consistency and correctness.

Dataset	Train	Test	Label Space
Emotion	3,257	1,421	joy, optimism, sadness, anger
Irony	2,862	784	irony, not irony
Stance	2,620	1,249	favor, against, neutral
MRPC	4,076	1,587	equivalent, not equivalent
RTE	2,490	277	entailment, not entailment

Table 1: Statistical overview of the five datasets, detailing training and test set sizes along with their corresponding task labels.

3.6 Human Verification on Data Quality

To ensure the reliability of our proposed strategy, we conduct a comprehensive human evaluation with five experienced computational linguistics researchers. All evaluators underwent prior training to ensure consistency in their assessments. The evaluators analyze 50 randomly selected samples based on four key criteria: **Fluency**, **Coherence**, **Factuality**, and **Accuracy**. Following the approach of Ying et al. (2024), Fluency and Coherence are rated on a 3-point scale: 2 (Good), 1 (Acceptable), and 0 (Unsatisfactory). Factuality and Accuracy are rated as 1 (Yes) or 0 (No). Detailed evaluation guidelines can be found in Appendix D.

To assess inter-annotator agreement, we use Fleiss' Kappa Statistic (Fleiss, 1971). As shown in Table 2, the results demonstrates that our method generates high-quality data through proper demonstration and structured workflow. Moreover, the values of κ falling within the range $0.70 < \kappa < 0.85$ indicate substantial agreement among annotators.

Dataset	Fluency	Coherence	Factuality	Accuracy	κ
Emotion	2.99	2.55	0.98	0.94	0.73
Irony	2.97	2.74	0.99	0.97	0.78
Stance	2.99	2.56	0.98	0.96	0.73
MRPC	2.98	2.92	0.98	0.96	0.86
RTE	2.99	2.86	0.96	0.96	0.80

Table 2: The statistics of the updated datasets are presented. κ denotes Fleiss' Kappa (Fleiss, 1971).

4 Experiment

This section first presents the experimental setups, including model configurations and metrics. We then address the following questions to assess the effectiveness of our CoreEval: **Q1:** How does LLM performance change across different tasks after data updates? **Q2:** Does CoreEval outperform existing methods in resisting data contamination? **Q3:** How does the dataset perform under different contamination proportions and types?

4.1 Experiment Setup

Large Language Models. For our experimental investigation, we curated a diverse set of language models comprising eight widely-adopted **open-source LLMs**: Llama3-8B (Dubey et al., 2024), Llama2-13B (Touvron et al., 2023), Ministral-8B (MistralAI, 2024b), Mistral-NeMo-12B (MistralAI, 2024a) (abbreviated as Mistral-12B), Yi1.5-6B (Young et al., 2024), Yi1.5-9B (Young et al., 2024), Qwen2.5-7B (Qwen, 2024), and Qwen2.5-14B (Qwen, 2024)[3]. The experimental evaluation also included three prominent **proprietary LLMs**: ChatGPT, Gemini1.5, and Claude3.5[4].

Evaluation Metrics. Inspired by Opitz (2024), we adopted the macro F1-score as the unified evaluation metric across all tasks to ensure consistency in performance assessment. Following Ying et al. (2024), we evaluate the model's performance P using the macro F1-score and subsequently employ performance gain as a metric to assess its resilience to data contamination. This metric quantifies the improvement from test set fine-tuning, with a smaller boost indicating greater resistance to contamination. In the contamination test experiment, we implement two simulation settings. The first involves training solely on the test set and measuring the performance gain $\delta_1 = P_{test} - P_{zero}$ against zero-shot performance where P_{test} denotes performance after fine-tuning on the test set only, and P_{zero} represents the zero-shot performance. The second setting incorporates both training and test sets, comparing the performance gain $\delta_2 = P_{train+test} - P_{train}$. where P_{train} indicates performance after fine-tuning on the training set alone, and $P_{train+test}$ represents performance after fine-tuning with both training and test sets. Detailed

[3] For all aforementioned open-source models, we utilized instruction-tuned versions of the model weights.

[4] In our experiments, we utilized the following model versions: gpt-3.5-turbo-0125 for ChatGPT, gemini-1.5-flash for Gemini1.5, and claude-3-5-haiku-20241022 for Claude3.5.

information about metric δ can be found in Appendix B.

4.2 Performance Test (Q1)

We first evaluate the zero-shot performance of LLMs on both the original and our updated datasets, using zero-shot evaluation as a standard configuration for assessing LLMs capabilities. We analyze how LLMs performance varies across different tasks after data updates. Refer to Appendix C.1 for the inference configurations. To mitigate prompt bias, we average results across multiple prompt templates, with detailed prompts provided in Appendix A.2.

The experimental results, illustrated in Figure 3, reveal the following: 1) While **proprietary models generally outperform most open-source models, the Qwen2.5 series achieves comparable or even superior performance** among open-source models. 2) **Emotion recognition and stance detection tasks substantially decline in performance on our updated dataset relative to the original one**. This decline can be attributed to two factors. First, these tasks may already be contaminated in existing LLMs, leading to decreased performance on our updated dataset, which aligns with prior studies (Aiyappa et al., 2023; Sainz et al., 2024). Second, emotion and stance tasks inherently involve more subjective interpretations and contextual nuances, requiring an understanding of complex, evolving social and cultural contexts. The injection of new knowledge can alter textual patterns, including time-dependent emotional and stance expressions, thereby affecting LLM judgments. This underscores the importance of timely LLM iterations. 3) Proprietary models exhibit a more significant performance drop of 5.42%, compared to 3.62% for open-source models, suggesting that **proprietary models may suffer from more severe data contamination**. The lack of transparency in their training data and model parameters makes detecting and mitigating data contamination in proprietary systems a critical challenge.

4.3 Contamination Test (Q2)

To assess the effectiveness of our method in mitigating the overestimation problem caused by data contamination, we follow prior studies (Zhou et al., 2023; Ying et al., 2024) and simulate data contamination scenarios. Specifically, we introduce test prompts and the test set with ground truth labels, during the training phase to simulate data contamination conditions, enabling a rigorous assessment of our approach's resistance to data leakage.

We conduct contamination simulations on eight open-source models, comparing results across three types of datasets: the original dataset $\mathcal{D}$, semantic dataset $\mathcal{D}^s$, and our updated dataset $\hat{\mathcal{D}}$. Detailed training configurations are provided in Appendix C.2. The results are presented in Table 3, where δ_1 captures both the model's ability to improve task comprehension and its potential to memorize test set information due to contamination. In contrast, δ_2 isolates the effect of training data, making it a more reliable indicator of contamination by attributing performance gains solely to test set exposure. This distinction ensures that δ_2 provides a precise measure of an LLM's resistance to data contamination. Our observations reveal several critical trends regarding data contamination in LLMs:

Performance overestimation intensifies with increasing model size in contaminated settings. For instance, in our simulation using the original dataset, Qwen2.5-7B shows δ_1 and δ_2 values of 12.01 and 4.74, respectively, whereas the larger Qwen2.5-14B model exhibits higher values of 17.45 and 7.19. This trend is consistent across different model series. However, when tested on our updated dataset, these parameter-scale-induced discrepancies are significantly reduced.

Cognitively complex tasks are more sensitive to data contamination. Tasks such as irony detection, stance detection, and RTE, consistently yield higher δ values, suggesting a positive correlation between task cognitive complexity contamination sensitivity. These cognitively demanding tasks may prompt models to rely more on shortcuts like memorization, making them more vulnerable to data contamination compared to simpler tasks like emotion recognition and MPRC.

Our real-world knowledge integration method significantly improves contamination mitigation. While simple data rewriting techniques provide some resistance to data contamination, our method, incorporating real-world real-time knowledge, demonstrates superior performance mitigating overestimation and counteracting the effects of contamination. Notably, it outperforms conventional approaches such as *semt*, highlighting the importance of dynamic knowledge updates in ensuring model robustness.

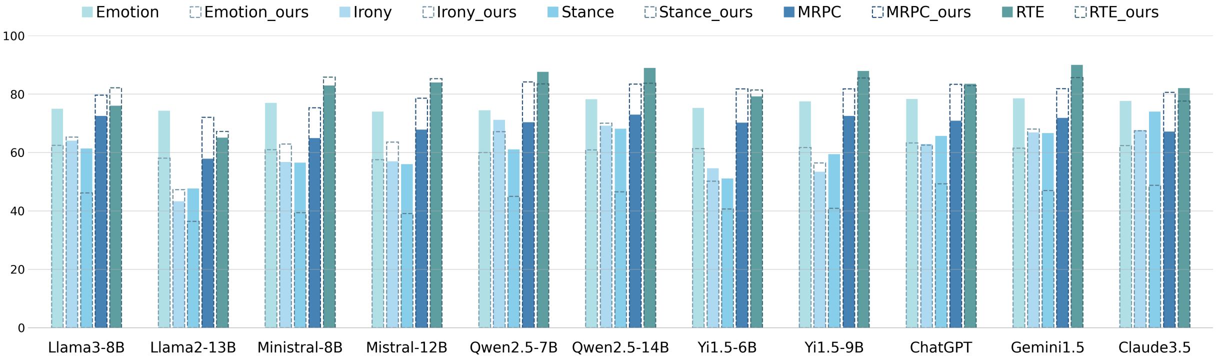


Figure 3: Performance (%) of the eleven involved LLMs (zero-shot) on the original and our updated datasets. We employ various prompt templates and use their average as the final result. Refer to Appendix C.3 for further details.

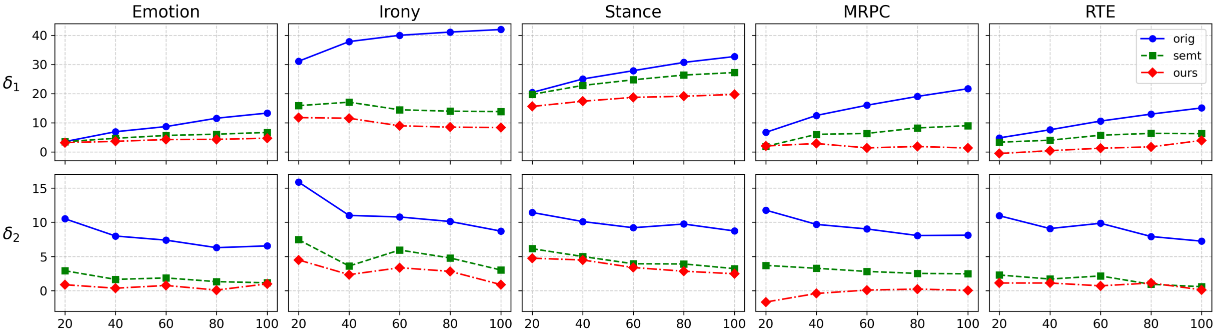


Figure 4: Data contamination resistance (%) of eight open-source models under different data proportions (20%, 40%, 60%, 80%, 100%). The first row shows δ_1 values for the test set-only scenario across the original dataset, semantic dataset, and our updated dataset. The second row presents δ_2 values for the train and test set scenario. The results are the mean values calculated across all eight open-source models.

4.4 Impact of Contamination Proportion (Q3)

In this section, we examine how varying data proportions influence the effects of data contamination. For the *'test set only'* simulated scenario, we sample different proportions of the test set to compute δ_1 and analyze how varying ratios of the test data contamination impact performance overestimation. For the *'training set and test set'* simulated scenario, we vary the proportion of the training set and compute δ_2 by incorporating it with the test set. All training configurations remain consistent with those detailed in Section 4.3. The results are visualized in Figure 4.

δ_1 exhibits an upward trend, reflecting increasing performance overestimation as more test set data is exposed. This is expected, as greater test set contamination amplifies the model's memorization effect, artificially inflating performance.

δ_2 demonstrates a downward trend, aligning with the explanation in Section 4.3. This metric isolates and quantifies performance improvements resulting from test set contamination, independent of enhanced task understanding. When incorporating the training set during the training process, models develop task understanding primarily through training data rather than test data. Therefore, as the proportion of the training set increases, δ_2 effectively filters out the performance gains attributed to task understanding from test data, leading to a more precise measurement of performance overestimation due to contamination by the test data.

Our updated dataset demonstrates stronger resistance to data contamination across both scenarios, significantly reducing performance overestimation regardless of task complexity or the ratio between test and training sets. Further analysis of the mean and variance of δ_1 and δ_2 across different proportions for the original, semantic, and our datasets (outlined in Appendix C.4) reveals that our CoreEval provides more stable metrics across various data proportions compared to both the original and semantic datasets. These findings underscore the critical role of incorporating real-world and real-time knowledge into dataset design to enhance model robustness against data contamination.

		Emotion		Irony		Stance		MRPC		RTE		AVG	
		δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓
Llama3-8B	*orig*	9.37	4.47	30.09	7.07	23.41	6.80	10.98	7.05	20.88	8.79	18.95	6.84
	semt	4.86	1.34	9.66	3.05	20.00	3.14	6.37	2.78	12.20	**0.12**	10.62	2.09
	ours	**3.27**	**1.33**	**2.00**	**1.89**	**11.66**	**2.57**	**0.75**	**0.53**	**4.21**	0.13	**4.38**	**1.29**
Llama2-13B	*orig*	11.83	4.55	52.46	7.97	38.26	6.98	26.98	6.83	26.24	9.02	31.16	7.07
	semt	7.69	1.60	23.15	2.42	32.70	3.26	18.44	2.50	22.63	2.15	20.92	2.38
	ours	**7.41**	**0.57**	**18.23**	**1.12**	**24.50**	**2.56**	**10.09**	**-0.31**	**21.12**	**1.10**	**16.27**	**1.01**
Ministral-8B	*orig*	12.65	6.85	39.66	7.58	30.80	8.51	25.64	8.91	17.08	6.64	25.17	7.70
	semt	6.77	1.58	10.53	1.98	28.47	3.38	11.94	2.84	6.50	-0.72	12.84	1.81
	ours	**4.41**	**0.15**	**2.54**	**0.58**	**20.97**	**2.36**	**3.86**	**0.32**	**4.03**	**-1.46**	**7.16**	**0.39**
Mistral-12B	*orig*	17.41	7.59	40.43	10.59	34.69	9.35	26.51	9.30	13.43	7.49	26.50	8.86
	semt	10.83	**1.44**	8.46	4.27	30.49	3.69	10.54	2.28	2.74	0.11	12.61	2.36
	ours	**7.64**	1.54	**2.92**	**3.40**	**23.35**	**3.19**	**0.61**	**0.43**	**1.45**	**-0.62**	**7.19**	**1.59**
Yi1.5-6B	*orig*	11.42	4.65	39.78	8.45	31.75	8.64	14.96	7.71	19.64	8.80	23.51	7.69
	semt	4.76	**0.60**	20.47	2.62	24.70	2.46	6.92	1.39	11.16	**0.36**	13.60	1.49
	ours	**3.50**	0.84	**16.35**	**0.75**	**18.79**	**2.45**	**-1.00**	**0.21**	**6.76**	1.69	**8.88**	**1.19**
Yi1.5-9B	*orig*	15.03	9.04	44.34	14.13	33.67	11.60	23.87	10.41	9.21	8.08	25.22	10.65
	semt	6.17	1.94	12.86	2.31	25.50	3.79	6.48	1.89	2.53	1.71	10.71	2.33
	ours	**4.50**	**0.51**	**7.59**	**0.55**	**19.66**	**2.00**	**-3.50**	**0.27**	**0.45**	**-0.37**	**5.74**	**0.59**
Qwen2.5-7B	*orig*	6.65	3.44	19.77	4.86	18.51	5.24	8.06	4.16	7.08	6.03	12.01	4.74
	semt	4.93	1.06	10.37	2.77	18.06	2.87	3.21	2.32	**1.74**	**0.72**	7.66	1.95
	ours	**4.72**	**0.61**	**6.82**	**2.31**	**15.25**	**2.32**	**0.04**	**-0.39**	2.31	1.08	**5.83**	**1.19**
Qwen2.5-14B	*orig*	11.53	5.71	27.75	9.78	20.83	8.10	19.95	6.94	7.21	5.43	17.45	7.19
	semt	5.79	1.46	1.46	2.76	17.03	2.87	5.49	1.42	**0.52**	0.12	6.06	1.72
	ours	**4.57**	**0.99**	**-3.57**	**0.93**	**13.98**	**1.20**	**-4.73**	**0.37**	4.38	**0.00**	**2.93**	**0.70**

Table 3: Data contamination resistance (%) of eight open-source models across simulated scenarios. *orig* denotes using original dataset, *semt* denotes using semantic dataset, which involves restating the text while preserving its original meaning, and *ours* denotes using our updated dataset. Following Section 4.2, we use multiple prompt templates to mitigate prompt biases, reporting averaged performance. Best performances are in bold.

	Emotion		Irony		Stance		MRPC		RTE		AVG	
	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓
Llama3-8B	-0.32	-0.01	-0.13	0.12	0.03	-0.06	0.22	0.05	-9.20	-4.77	-1.88	-0.93
Llama2-13B	0.23	0.05	0.28	0.83	0.00	0.05	0.24	0.15	-4.57	-3.72	-0.76	-0.53
Ministral-8B	-0.15	-0.14	-0.14	0.56	0.12	0.18	1.52	-0.06	-15.14	-10.61	-2.76	-2.01
Mistral-12B	0.18	0.09	0.08	-0.52	-0.02	0.40	0.36	0.22	-0.21	0.96	0.08	0.23
Yi1.5-6B	-0.48	0.40	0.30	0.45	0.03	0.23	-0.07	-0.21	-3.29	4.28	-0.70	1.03
Yi1.5-9B	0.02	0.29	-0.20	0.73	-0.29	-0.25	-0.03	0.23	-3.77	-4.08	-0.85	-0.62
Qwen2.5-7B	0.03	0.25	0.04	0.25	0.23	-0.11	0.24	0.25	-2.28	-3.82	-0.35	-0.64
Qwen2.5-14B	-0.06	0.10	-0.05	-0.09	-0.17	-0.19	0.30	0.15	-1.40	-2.71	-0.28	-0.55

Table 4: Data contamination resistance performance (%) of eight open-source models on original datasets under text-only contamination scenarios.

4.5 Impact of Contamination Types (Q3)

In this section, we further extend our investigation by implementing a text-only contamination test, drawing upon the methodologies proposed by Li et al. (2024c) and Jiang et al. (2024b). Diverging from previous simulation scenarios that involved the exposure of both test labels and texts during the training phase, this specific experimental setup exclusively leaks the textual content of the evaluation samples. Detailed training configurations are elaborated in Appendix C.2, and the comprehensive results are presented in Table 4.

The experimental findings indicate that the δ_1 and δ_2 values, when measured on the original datasets under these text-only contamination conditions, are predominantly negative across eight distinct open-source models. This observation suggests that text-only contamination, without label leakage, does not contribute to performance overestimation, consistent with the prior research by Li et al. (2024c). Conversely, the substantial performance improvements observed in Table 3, where test sets including ground truth labels and test prompts are contaminated, highlight the critical need for targeted mitigation strategies to address this type of data contamination.

5 Conclusion

In this paper, we introduce CoreEval, an automatic contamination-resilient evaluation framework incorporating real-time real-world knowledge. We further propose a structured workflow engineered to guarantee the timeliness and reliability of LLM evaluations. Extensive experiments across various NLP tasks demonstrate CoreEval's robust effectiveness in mitigating data contamination. CoreEval is developed to be broadly applicable across NLP tasks, delivering efficient contamination-resilient evaluation while ensuring high data quality with minimal human intervention, thus facilitating fairer and more timely LLM assessment.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China 62176076, Natural Science Foundation of Guang Dong 2023A1515012922, the Shenzhen Foundational Research Funding JCYJ20220818102415032, the Major Key Project of PCL2023A09, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005 and CIPSC-SMP-ZHIPU Large Model Cross-Disciplinary Fund ZPCG20241119405.

Limitations

Our proposed CoreEval framework updates text based on up-to-date and real-world knowledge. Although we have implemented data reflection and iteration processes to minimize inaccuracies, there is a possibility of generating a minimal amount of hallucinated data. Given our manual evaluation scores for the quality of updated data, the impact of such minimal hallucinated data on the evaluation of LLMs for most NLP tasks is negligible. Furthermore, in this study, CoreEval is applied only to classification tasks. In the future, we plan to extend its application to more complex tasks such as question answering and summarization.

Ethics Statement

The datasets used in this study are sourced from open-access datasets, ensuring compliance with data accessibility standards. We have taken measures to remove any information related to user privacy from these datasets to protect individual identities and maintain confidentiality. The real-world knowledge required for updates is sourced from GDELT. While updating the data, there is a possibility of introducing references to relevant individuals or events. We have made every effort to ensure that these references are accurate and respectful.

References

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on chatgpt? In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 47–54.

Sourav Banerjee, Ayushi Agarwal, and Eishkaran Singh. 2024. The vulnerability of language model benchmarks: Do they accurately reflect true llm performance? *arXiv preprint arXiv:2412.03597*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Jerome S Bruner. 2009. *The process of education*. Harvard university press.

Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230.

Jasper Dekoninck, Mark Niklas Müller, Maximilian Baader, Marc Fischer, and Martin Vechev. 2024a. Evading data contamination detection for language models is (too) easy. *arXiv preprint arXiv:2402.02823*.

Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. 2024b. Constat: Performance-based contamination detection in large language models. *arXiv preprint arXiv:2405.16281*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in covid-19 tweets. In *Proceedings of the 59th annual meeting of the association for computational* *linguistics and the 11th international joint conference on natural language processing (long papers)*, volume 1.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53.

Minhao Jiang, Ken Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024a. Does data contamination make a difference? insights from intentionally contaminating pre-training data for language models. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024b. Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2024a. Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18582–18590.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024b. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18600–18607.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024c. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541.

Junxia Ma, Changjiang Wang, Hanwen Xing, Dongming Zhao, and Yazhou Zhang. 2024. Chain of stance: Stance detection with large language models. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 82–94. Springer.

MistralAI. 2024a. Mistral nemo: our new best small model.

MistralAI. 2024b. Un ministral, des ministraux: Introducing the world's best edge models.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Juri Opitz. 2024. A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *Transactions of the Association for Computational Linguistics*, 12:820–836.

Prakrit Pathak and Prashant Singh Rana. 2024. Comparative analysis of pretrained models for text classification, generation and summarization: A detailed analysis. In *International Conference on Pattern Recognition*, pages 151–166. Springer.

Qwen. 2024. Qwen2.5: A party of foundation models.

Anna Rogers, Alexey Romanov, Anna Rumshisky, Svitlana Volkova, Mikhail Gronas, and Alex Gribov. 2018. Rusentiment: An enriched sentiment analysis dataset for social media in russian. In *Proceedings of the 27th international conference on computational linguistics*, pages 755–763.

Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2013. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold.

Oscar Sainz, Iker García-Ferrero, Alon Jacovi, Jon Ander Campos, Yanai Elazar, Eneko Agirre, Yoav Goldberg, Wei-Lin Chen, Jenny Chim, Leshem Choshen, et al. 2024. Data contamination report from the 2024 conda shared task. *arXiv preprint arXiv:2407.21530*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, et al. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*.

Xiaobao Wu, Liangming Pan, Yuxi Xie, Ruiwen Zhou, Shuai Zhao, Yubo Ma, Mingzhe Du, Rui Mao, Anh Tuan Luu, and William Yang Wang. 2024. Antileak-bench: Preventing data contamination by automatically constructing benchmarks with updated real-world knowledge. *arXiv preprint arXiv:2412.13670*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and YAN Shuicheng. 2024. Automating dataset updates towards reliable and timely evaluation of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

A Various Prompt Templates

A.1 Prompt of CoreEval Framework

Figure 5 presents the prompts in the process of Real-World Knowledge Attainment. The workflows of Knowledge contextualization are shown in Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, and Figure 11. Ultimately, Figure 12 and Figure 13 demonstrate the prompts of data reflection.

A.2 Prompt of Contamination Test

To address potential result bias stemming from task sensitivity to prompts, we employed three prompt templates for each task. The performance metrics were then averaged across these prompt variations to obtain the final results. The comprehensive set of prompt templates utilized for all five tasks is detailed in Table 5, 6, 7, 8, and 9, which present the complete prompt formulations for each task-specific evaluation.

B Data Contamination Resistance Indicators

Data contamination, which refers to the inflated performance of a model on a specific dataset or benchmark due to the leakage of test data, can distort the true evaluation and assessment of a LLM's capabilities. (Zhou et al., 2023; Dekoninck et al., 2024b) Therefore, mitigating the overestimation of performance caused by data contamination is key to addressing this issue. The degree of spurious performance growth following data contamination becomes the primary metric for evaluating data contamination mitigation efforts.

However, precisely determining whether a model has been contaminated by certain datasets remains challenging in practice. Previous studies have simulated data contamination by directly training models on test sets of specific datasets (Ying et al., 2024; Li et al., 2024c; Jiang et al., 2024b; Zhou et al., 2023). The mitigation effectiveness is then quantified by measuring the performance gap between the contaminated model before and after data updates. In our work, we similarly introduce δ_1, which measures the performance difference between the model's evaluation results after training solely on the test set and its zero-shot performance (i.e., performance without any training) as one of the indicators for evaluating data contamination mitigation.

Furthermore, we argue that the performance improvements of LLMs directly exposed to test set data may stem from two sources: enhanced task understanding through exposure to task-specific data, and direct memorization effects from test set contamination. To isolate the latter effect, we propose δ_2, which compares the performance difference between models trained on both train and test sets versus those trained exclusively on the train set. δ_2 effectively eliminates the task-understanding gains from the train set while capturing the additional benefits derived from test set inclusion in training (i.e., the primary impact of data contamination), thereby providing a more accurate reflection of data contamination's contribution to model performance.

The substantial difference between these two indicators, as demonstrated in Table 3, effectively validates this observation. Moreover, the declining trend of δ_2 with increasing train set proportions, as illustrated in Figure 4, confirms that this indicator successfully isolates the impact of data contamination by removing the contribution of improved task understanding.

C Experimant Detail

C.1 Inference Configuration in Performance Test

For proprietary models, we set the temperature to 1.0, top-p to 1.0, max tokens to 1024, and fixed the seed to ensure experimental reproducibility. For open-source models, we load model weights in bf16 format, set the temperature to 1.0, top-p to 1.0, max tokens to 512, and apply greedy decoding to guarantee reproducibility.

C.2 Training Configuration in Contamination Test

Due to computational resource constraints, we applied LoRA fine-tuning (Hu et al., 2021) to eight open-source models. The LoRA hyperparameters were configured with a rank of 16, alpha of 32, dropout of 0.1, learning rate of 1e-4, and 3 epochs. For the RTE task, we set the training batch size to 2 and maximum sequence length to 512. For all other tasks, the maximum sequence length was set to 400, while the training batch size was adjusted according to model size. Specifically, Llama3-8B, Qwen2.5-7B, Mistral-8B, and Yi1.5-6B were trained with a batch size of 8; Yi1.5-9B, Llama2-13B, and Mistral-12B with a batch size of 3; and Qwen2.5-14B with a batch size of 2. For text-only contamination simulated scenarios, we configured the LoRA hyperparameters with a rank of 16, alpha of 32, dropout of 0.1, training batch size of 1, maximum sequence length of 1024, and 3 epochs. The learning rate was set to 1e-3 for the RTE task and 1e-5 for other tasks.

During inference, we employed a greedy decoding strategy by setting do_sample to False and num_sample to 1, thereby ensuring the reproducibility of our experimental results.

C.3 Experimental Result of Performance Test

We employed a greedy decoding strategy by setting do_sample to False and num_sample to 1, thereby ensuring the reproducibility of our experimental results. The detailed results of the original dataset and our updated dataset are presented in Table 10.

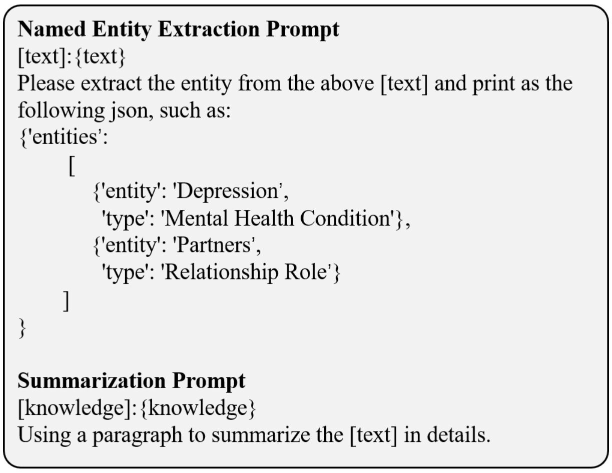
C.4 Experimental Result of Data Proportion Analysis

Table 11 presents the detailed experimental results of our data proportion analysis, encompassing the performance of eight open-source models across five tasks. The evaluation was conducted using varying proportions (20%, 40%, 60%, 80%, and 100%) of both test and training sets, along with the average performance across all five tasks.

Table 12 illustrates the standard deviations in data contamination resistance performance under varying data proportions for three datasets: the original, semantic, and our proposed updated dataset. The analysis reveals that our updated dataset consistently achieves lower variance compared to its counterparts. This reduced variability substantiates that our dataset yields more stable and robust evaluation metrics across different degrees of data contamination.

D Guideline of Human Evaluation

Table 13 outlines the guidelines for human evaluation. Before presenting annotators with the final evaluation materials, we conduct a training session, providing them with this form and comprehensive instructions. This helps ensure they fully grasp the evaluation process, the significance of each metric, and the corresponding scoring standards.



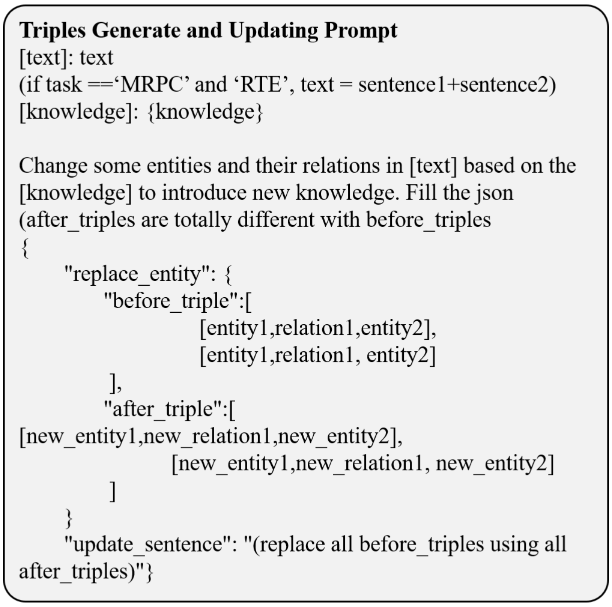
Named Entity Extraction Prompt

```
[text]:{text}
Please extract the entity from the above [text] and print as the
following json, such as:
{'entities':
      [
         {'entity': 'Depression',
          'type': 'Mental Health Condition'},
         {'entity': 'Partners',
          'type': 'Relationship Role'}
      ]
}
```

Summarization Prompt

[knowledge]:{knowledge}
Using a paragraph to summarize the [text] in details.

Figure 5: Prompt in Real-World Knowledge Attainment



Triples Generate and Updating Prompt

[text]: text
(if task ==‘MRPC’ and ‘RTE’, text = sentence1+sentence2)
[knowledge]: {knowledge}

```
Change some entities and their relations in [text] based on the
[knowledge] to introduce new knowledge. Fill the json
(after_triples are totally different with before_triples
{
      "replace_entity": {
           "before_triple":[
                        [entity1,relation1,entity2],
                        [entity1,relation1, entity2]
            ],
           "after_triple":[
[new_entity1,new_relation1,new_entity2],
                     [new_entity1,new_relation1, new_entity2]
            ]
      }
      "update_sentence": "(replace all before_triples using all
after_triples)"}
```

Figure 6: Prompt of triples generation and updating.

Emotion Recognition Prompt
[text]: {text}
[relation]:{replace_entity}

1.Now you are a [sentence] writer expert, your objective it to write **only one** really complex and difficult [semantic _sentence].
2. [semantic _sentence] and [sentence] express the same emotion.
3.The length of [semantic _sentence] is len(text)

Please change the [text] structure without changing the [relation] and the meaning, fill the json:{
"semantic_sentence": "reason": "(how to keep the emotion of [semantic _sentence] and [sentence] same)"}

Irony Detection Prompt
[text]: {text}
[relation]:{replace_entity}

1.Now you are a [sentence] writer expert, your objective it to write **only one** really complex and difficult [semantic_sentence].
2. [semantic_sentence] and [sentence] express the same emotion.
3.The length of [semantic_sentence] is len(text)

Please change the [text] structure without changing the [relation] and the meaning, fill the json:{
"semantic _sentence": "reason": "(how to keep the emotion of [semantic_sentence] and [sentence] same)"}

Stance Detection Prompt
[text]: {text}
[target]: {target}
[relation]:{replace_entity}

1. Now you are a [sentence] writer expert, your objective it to write **only one** really complex and difficult [semantic_sentence].
2. Please change the [sentence] structure without changing the [relation] and the meaning.
3. The stance (favor, against, none) of [semantic_sentence] and [sentence] for [target] are same.
4. The length of [semantic_sentence] is len(text)

Fill the json:{"semantic_sentence":"", "reason":""}

Figure 7: Prompt of semantic rewriting for emotion recognition, irony detection, and stance detection tasks.

MRPC Prompt
# Change sentence 1
[sentence1:{sentence1}
[sentence2]: {sentence2}
[relation]:{replace_entity}

1.Now you are a [sentence1] writer expert, your objective it to write **only one** really complex and difficult [rephrase_sentence1] based on the [sentence1].
2. Ensure that [relation] and the relation remain intact, and preserve the original tone and meaning
3.The length of [rephrase_sentence1] is len(sentence1)

Fill the json:{'rephrase_sentence1':",'reason':'the relation (semantically_equivalent or not_semantically_equivalent) between [sentence2] and [rephrase_sentence1]'}

# Change sentence 2
[sentence1]: {rephrase_sentence1}
[sentence2]: {sentence2}
[relation]:{replace_entity}

1.Now you are a [sentence2] writer expert, your objective it to write **only one** really complex and difficult [rephrase_sentence2] based on the given [sentence2].
2. Ensure that [relation] and the relation remain intact, and preserve the original tone and meaning.
3. The length of [rephrase_sentence2] is len(sentence2)

Fill the json:{'rephrase_sentence2':",'reason':'the relation (semantically_equivalent or not_semantically_equivalent) between [rephrase_sentence2] and [sentence1]'}"

RTE Prompt
# Change hypothesis
[premise]:{premise}
[hypothesis]: {hypothesis}
[relation]:{replace_entity}

1.Now you are a [premise] writer expert, your objective it to write **only one** really complex and difficult [promise] based on the given [hypothesis]
2. Ensure that [relation] and the relation remain intact, and preserve the original tone and meaning while rephrasing the structure of [premise].
3. The length of [rephrase_ premise] is len(premise)
Fill the json:{'rephrase_premise':", 'reason':''}"

# Change premise
[premise]:{rephrase_ premise}
[hypothesis]: {hypothesis}
[relation]:{replace_entity}

1.Now you are a [hypothesis] writer expert, your objective it to write **only one** really complex and difficult [hypothesis] based on the given [premise]
2. Ensure that [relation] and the relation remain intact, and preserve the original tone and meaning while rephrasing the structure of [premise].
3. The length of [rephrase_ hypothesis] is len(hypothesis)
Fill the json:{'rephrase_ hypothesis ':", 'reason':''}"

Figure 8: Prompt of semantic rewriting for MRPC and RTE tasks.

Emotion Recognition Prompt
[style_sentence]: {semantic_result}
[original_sentence]:{text}
[sentence]:{update_sentence}
[relation]:{replace_entity}

1.Now you are a [sentence] writer expert, your objective it to write **only one** really complex and difficult [our_sentence].
2. The structure of [our_sentence] is similar to [style_sentence].
3. Do not change the [relation]
4. The emotion of [our_sentence] and [original_sentence] are same.
5. The length of [our_sentence] is len(style_sentence)
Fill the json:{'our_sentence':'','reason':‘’}

Irony Detection Prompt
[sentence]:{update_sentence}
[style_sentence]: {'semantic_result}
[original_sentence]:{text}
[relation]:{relation}

1. Now you are a [knowledge] writer expert, your objective it to write **only one** really complex and difficult [our_sentence].\n" \
2. The structure of [our_sentence] is similar to [style_sentence].
3. Do not change the [relation]
4. The emotion (irony or none_irony) of [our_sentence] and [original_sentence] are same.
5. The length of [our_sentence] is len(original_sentence)

Fill the json:{'our_sentence':'','reason':'(how to keep the emotion of (irony or none_irony) [our_sentence] and [text] same)'}"

Stance Detection Prompt
[sentence]:{update_sentence}
[style_sentence]: {semantic_result}
[original_sentence]:{text}
[relation]:{relation}
[target]: {target}

1. Now you are a [sentence] writer expert, your objective it to write **only one** really complex and difficult [our_sentence].\n" \
2. The structure of [our_sentence] is similar to [style_sentence]. \n" \
3. Do not change the [relation];\n" \
4. Think the stance (favor, against, none) of [original_sentence] for [target], and make sure the stance of [our_sentence] is the same
5. The length of [our_sentence] is str(original_sentence).

Fill the json:{'our_sentence':'','reason':'talk about the stance (favor, against, none) of [text] and [our_sentence] for [target]}

Figure 9: Prompt of updated sentence for emotion recognition, irony detection, and stance detection tasks.

MRPC Prompt

```
# Change sentence1
[sentence1]:{update_sentence1}
[original_sentence1]:{sentence1}
[sentence2]:{semantic_sentence2}
[original_sentence2]:{sentence2}
[style_sentence1]: {semantic_sentence1}
[knowledge]:{knowledge}
[relation]:{relation}

Now you are a sentence1 rewriter expert, your objective it to write **only one** really complex and difficult [our_sentence1].
1. The style and structure of [our_sentence1] are similar to [style_sentence1].
2. Keep [relation] in [our_sentence1].
3. The relation between[our_sentence1] and [sentence2] is similar to the relation between [original_sentence1] and [sentence2]
4. The length of [our_sentence1] is len(original_sentence1)

Fill the [our_sentence1] and the reason in the following json: {"our_sentence1":"","sentence1_reason":"the relation (semantically_equivalent or not_semantically_equivalent) between [our_sentence1] and [sentence2]"}’

# Change sentence2
[sentence1]:{our_sentence1}
[original_sentence1]:{sentence1}
[sentence2]:{update_sentence2}
[original_sentence2]:{sentence2}
[style_sentence2]: {semantic_sentence2}
[relation]:{relation}

Now you are a [sentence2] rewriter expert, your objective it to write **only one** really complex and difficult [our_sentence2] based on the given [sentence1]
1. The style and structure of [our_sentence2] are similar to [style_sentence2].
2. Keep [relation] in [our_sentence2].
3. The relation between[our_sentence2] and [sentence1] is similar to the relation between[original_sentence2] and [original_sentence1]
4. The length of [our_sentence2] is len(original_sentence2)

Fill the [our_sentence2] and the reason in the following json: {"our_sentence2":"","sentence2_reason":"sentence1_reason":"the relation (semantically_equivalent or not_semantically_equivalent) between [our_sentence2] and [sentence1]"}'
```

Figure 10: Prompt of semantic rewriting for MRPC task.

RTE Prompt
# Change premise
[style_ premise]:{semantic_premise}
[premise]: {update_premise}
[original_premise]:{premise}
[hypothesis]:{semantic_hypothesis}
[original_hypothesis]:{original_hypothesis}
[relation]:{relation}

Now you are a premise rewriter expert, your objective it to write **only one** really complex and difficult [our_premise].
1. The style, structure of [our_premise] are similar to [style_premise].
2. Keep [relation] in [our_premise].
3. The relation between[our_ premise] and [hypothesis] is similar to the relation between[original_premise] and [original_hypothesis]
4. The length of [our_premise] is len(original_premise)

Fill the [our_premise] and the reason in the following json: {"our_premise":"","premise_reason":""}

# Change sentence2
[style_hypothesis]:{semantic_hypothesis}
[premise]: {our_premise}
[original_premise]: {original_premise}
[hypothesis]:{hypothesis}
[original_hypothesis]:{hypothesis}
[relation]:{relation}

Now you are a premise rewriter expert, your objective it to write **only one** really complex and difficult [our_hypothesis].
1. The style, structure of [our_ hypothesis] are similar to [style_hypothesis].
2. Keep [relation] in [our_hypothesis].
3. The relation between[our_hypothesis] and [premise] is similar to the relation between[original_hypothesis] and [original_premise]
4. The length of [our_hypothesis] is len(original_hypothesis)

Fill the [our_hypothesis] and the reason in the following json: {"our_hypothesis":""," hypothesis_reason":""}

Figure 11: Prompt of semantic rewriting for RTE task.

Incorrect Information Prompt
# For emotion recognition, irony detection
[text]:{text}
[label]:{label}

Is the emotion conveyed by [text] [label]?

# For stance detection
[text]:{text}
[label]:{label}
[target]:{target}

Is the stance of [text] for [target] [label]?

# For MRPC and RTE
[sentence1]:{sentence1}
[sentence2]:{sentence2}
[label]:{label}

Are [sentence1] and [sentence2] [label]?

Figure 12: Prompt in Incorrect Information of Data Reflection

Label Alignment Prompt
[knowledge]:{knowledge}
[text]:{text}

Whether the [text] accurately represents the facts derived from the provided [knowledge].

Figure 13: Prompt in Label Alignment of Data Reflection

Variant	**Emotion Recognition Prompt**

Prompt #1	Emotion detection is the task of identifying the emotional tone expressed in a given text. The possible emotions include joy, anger, sadness, and optimism.
	The answer must be one and only one from the options: "joy", "anger", "sadness", and "optimism". No other responses are acceptable.
	If none of the options seem to apply, select the one that is closest to the emotional tone of the text.
	
	text: {text}
	Question: What is the primary emotion expressed in the text? Please select and only select the correct answer from "joy", "anger", "sadness", and "optimism". The response must strictly adhere to the following JSON FORMAT:
	{
	"emotion": "joy" | "anger" | "sadness" | "optimism"
	}

Prompt #2	Analyze the emotional tone of the following text. The emotions to choose from are joy, anger, sadness, and optimism.
	text: {text}
	Provide the detected emotion in JSON FORMAT:
	{
	"emotion": "joy" | "anger" | "sadness" | "optimism"
	}

Prompt #3	Emotion detection is the task of identifying the emotional tone expressed in a given text. The possible emotions include joy, anger, sadness, and optimism. **The answer must be one and only one from the options: "joy", "anger", "sadness", and "optimism". No other responses are acceptable.** If none of the options seem to apply, select the one that is closest to the emotional tone of the text. text: {text} Question: What is the primary emotion expressed in the text? Please select and only select the correct answer from "joy", "anger", "sadness", and "optimism".

Table 5: Prompt templates for Emotion Recognition task.

Variant	Irony Detection Prompt

Prompt #1	Irony detection is the task of identifying whether a given text contains irony. Irony is when the literal meaning of the text is opposite or significantly different from the intended meaning, often used to convey criticism, sarcasm, or humor. The possible labels are "irony" and "not irony".
	text: {text}
	Question: Does the text contain irony? Irony can be characterized by:
	- A contrast between what is said and what is meant (verbal irony).
	- A situation where the expected outcome is different from the actual result (situational irony).
	- Exaggeration or sarcasm used to convey a hidden message or criticism.
	Please select the correct answer from "irony" and "not irony".
	Answer this question with JSON FORMAT:
	{
	"irony detection": "irony" | "not irony"
	}

Prompt #2	Analyze the following text to determine whether it contains irony. Irony is when there is a discrepancy between what is said and what is meant, or when the outcome of a situation is unexpected or opposite to what was anticipated. The labels to choose from are "irony" and "not irony".
	text: {text}
	Please provide your answer in JSON FORMAT:
	{
	"irony detection": "irony" | "not irony"
	}

Prompt #3	Irony detection is the task of identifying whether a given text contains irony. Irony often involves: - Saying one thing while meaning another (verbal irony). - A situation where the outcome is surprising or opposite to what was expected (situational irony). - Sarcasm or exaggerated statements to make a point or criticism. text: {text} Question: Does the text contain irony? Please select the correct answer from "irony" and "not irony".

Table 6: Prompt templates for Irony Detection task.

Variant	Stance Detection Prompt

Prompt #1	Stance detection is to determine the attitude or tendency towards a certain target through a given sentence, including favor, against and neutral.
	text: {text}
	Question: What is the attitude of the text toward "{target}"? Please select the correct answer from "favor", "against" and "neutral".
	Answer this question with JSON FORMAT:
	{
	"stance": "favor" | "against" | "neutral"
	}

Prompt #2	Given a text, determine the sentiment towards the specified target: {target}. Possible answers are "favor", "against", or "neutral".
	text: {text}
	Please provide your answer in JSON FORMAT:
	{
	"stance": "favor" | "against" | "neutral"
	}

Prompt #3	Stance detection is to determine the attitude or tendency towards a certain target through a given sentence, including favor, against and neutral. text: {text} Question: What is the attitude of the text toward "{target}"? Please select the correct answer from "favor", "against" and "neutral".

Table 7: Prompt templates for Stance Detection task.

Variant	MRPC Prompt

Prompt #1	The task is to determine whether a given sentence pair is semantically equivalent. The possible labels are "semantically equivalent" and "not semantically equivalent".
	sentence1: {sentence1}
	sentence2: {sentence2}
	Question: Are the sentences semantically equivalent? Please select the correct answer from "semantically equivalent" and "not semantically equivalent".
	Answer this question with JSON FORMAT:
	{
	"mrpc": "semantically equivalent" | "not semantically equivalent"
	}

Prompt #2	Given a pair of sentences, determine whether they are semantically equivalent. The possible labels are "semantically equivalent" and "not semantically equivalent".
	sentence1: {sentence1}
	sentence2: {sentence2}
	Please provide your answer in JSON FORMAT:
	{
	"mrpc": "semantically equivalent" | "not semantically equivalent"
	}

Prompt #3	The task is to determine whether a given sentence pair is semantically equivalent. The possible labels are "semantically equivalent" and "not semantically equivalent".
	sentence1: {sentence1}
	sentence2: {sentence2}
	Question: Are the sentences semantically equivalent? Please select the correct answer from "semantically equivalent" and "not semantically equivalent".

Table 8: Prompt templates for MRPC (Microsoft Research Paraphrase Corpus) task.

Variant	RTE Prompt

Prompt #1	Recognizing Textual Entailment (RTE) is the task of determining whether a given premise entails a hypothesis. The possible labels are "entailment" and "not entailment".
	premise: {premise}
	hypothesis: {hypothesis}
	Question: Does the premise entail the hypothesis? Please select the correct answer from "entailment" and "not entailment".
	Answer this question with JSON FORMAT:
	{
	"rte": "entailment" | "not entailment"
	}

Prompt #2	Given a premise and a hypothesis, determine whether the premise entails the hypothesis. The possible labels are "entailment" and "not entailment".
	premise: {premise}
	hypothesis: {hypothesis}
	Please provide your answer in JSON FORMAT:
	{
	"rte": "entailment" | "not entailment"
	}

Prompt #3	Recognizing Textual Entailment (RTE) is the task of determining whether a given premise entails a hypothesis. The possible labels are "entailment" and "not entailment".
	premise: {premise}
	hypothesis: {hypothesis}
	Question: Does the premise entail the hypothesis? Please select the correct answer from "entailment" and "not entailment".

Table 9: Prompt templates for RTE (Recognizing Textual Entailment) task.

Model		Emotion	Irony	Stance	MRPC	RTE	AVG
Llama3-8B	*orig*	74.91	64.02	61.34	72.47	75.98	69.74
	ours	62.42	65.27	46.19	79.62	82.19	67.14
Llama2-13B	*orig*	74.25	43.29	47.68	57.86	65.02	57.62
	ours	58.06	47.24	36.41	72.06	67.20	56.19
Mistral-8B	*orig*	76.94	56.72	56.52	64.90	82.92	67.60
	ours	60.95	62.92	39.38	75.29	85.84	64.88
Mistral-12B	*orig*	73.96	56.95	55.95	67.82	83.91	67.72
	ours	57.55	63.53	39.09	78.61	85.30	64.82
Qwen2.5-7B	*orig*	74.43	71.15	61.02	70.30	87.59	72.90
	ours	60.01	67.13	44.93	84.13	83.48	67.94
Qwen2.5-14B	*orig*	78.19	69.12	68.08	72.95	88.93	75.45
	ours	60.85	70.05	46.55	83.40	83.69	68.91
Yi1.5-6B	*orig*	75.25	54.57	51.03	70.21	79.15	66.04
	ours	61.30	50.14	40.66	81.81	81.43	63.07
Yi1.5-9B	*orig*	77.48	53.40	59.36	72.48	87.89	70.12
	ours	61.73	56.45	40.91	81.75	85.54	65.28
ChatGPT	*orig*	78.32	62.93	65.67	70.86	83.53	72.26
	ours	63.23	62.54	49.29	83.36	82.88	68.26
Gemini1.5	*orig*	78.49	66.82	66.58	71.80	89.99	74.74
	ours	61.50	68.03	46.98	81.83	85.67	68.80
Claude3.5	*orig*	77.59	67.68	73.97	67.16	82.00	73.68
	ours	62.36	67.45	48.76	80.62	77.60	67.36

Table 10: Performance (%) of the eleven involved LLMs (zero-shot) on the original and our updated datasets. We utilize macro F1-score as the unified evaluation metric.

Proportion(%)		Emotion		Irony		Stance		MRPC		RTE		AVG	
		δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓	δ_1 ↓	δ_2 ↓
20	*orig*	3.52	10.51	31.09	15.87	20.39	11.43	6.69	11.77	4.80	10.95	13.30	12.10
	semt	3.48	2.92	15.86	7.45	19.73	6.14	**1.87**	3.70	3.32	2.30	8.85	4.50
	ours	**3.18**	**0.88**	**11.77**	**4.50**	**15.58**	**4.75**	2.03	**-1.65**	**-0.57**	**1.13**	**6.40**	**1.92**
40	*orig*	6.91	7.99	37.84	11.00	25.00	10.10	12.49	9.69	7.58	9.08	17.96	9.57
	semt	4.67	1.66	17.05	3.61	22.80	5.00	6.00	3.28	4.00	1.71	10.90	3.05
	ours	**3.61**	**0.37**	**11.53**	**2.35**	**17.38**	**4.49**	**2.86**	**-0.40**	**0.41**	**1.13**	**7.16**	**1.59**
60	*orig*	8.68	7.39	40.00	10.77	27.86	9.20	16.00	9.02	10.57	9.85	20.62	9.25
	semt	5.63	1.87	14.44	5.95	24.73	3.94	6.34	2.82	5.72	2.16	11.37	3.35
	ours	**4.23**	**0.77**	**8.95**	**3.36**	**18.70**	**3.39**	**1.37**	**0.10**	**1.26**	**0.72**	**6.90**	**1.67**
80	*orig*	11.55	6.29	41.11	10.11	30.70	9.74	19.03	8.06	12.96	7.92	23.07	8.42
	semt	6.07	1.33	13.96	4.78	26.36	3.91	8.22	2.53	6.35	**0.94**	12.19	2.70
	ours	**4.29**	**0.10**	**8.48**	**2.83**	**19.08**	**2.85**	**1.85**	**0.24**	**1.72**	1.13	**7.09**	**1.43**
100	*orig*	13.31	6.55	41.98	8.71	32.71	8.73	21.68	8.11	15.08	7.24	24.95	7.87
	semt	6.67	1.16	13.82	3.03	27.21	3.23	8.99	2.47	6.26	0.59	12.59	2.10
	ours	**4.70**	**1.01**	**8.34**	**0.90**	**19.72**	**2.48**	**1.32**	**0.06**	**3.93**	**0.13**	**7.60**	**0.92**

Table 11: Data contamination resistance performance (%) of eight open-source models across simulated scenarios under different data proportions (20%, 40%, 60%, 80%, 100%). The results are the mean values calculated across all eight open-source models. *orig* denote using original dataset, *semt* denote using semantic dataset, and *ours* denote using our updated dataset. We employ multiple prompt templates to avoid prompt-sensitive biases, and use their averaged performance as the final results. The best scores are in bold.

	Emotion		Irony		Stance		MRPC		RTE		AVG	
	$\sigma(\delta_1)$	$\sigma(\delta_2)$	$\sigma(\delta_1)$	$\sigma(\delta_2)$	$\sigma(\delta_1)$	$\sigma(\delta_2)$	$\sigma(\delta_1)$	$\sigma(\delta_2)$	$\sigma(\delta_1)$	$\sigma(\delta_2)$	$\sigma(\delta_1)$	$\sigma(\delta_2)$
orig	3.85	1.69	4.37	2.71	4.85	1.03	5.85	1.52	4.11	1.49	4.61	1.69
semt	1.25	0.69	**1.39**	1.79	3.00	1.14	2.77	**0.52**	**1.39**	0.75	1.96	0.98
ours	**0.60**	**0.38**	1.69	**1.32**	**1.64**	**1.00**	**0.63**	0.77	1.69	**0.44**	**1.25**	**0.78**

Table 12: Standard deviations of data contamination resistance performance (%) across different data proportions (20%, 40%, 60%, 80%, 100%). The results are the mean values calculated across all eight open-source models. The best scores are in bold.

Guideline of Human Evaluation	
(1) Fluency	
Definition	Assess whether the language of the sentence is fluent, without grammatical or spelling errors. The scoring range is 1-3.
Score	1 point: The text contains multiple grammatical and/or spelling errors, significantly impacting the readability and understanding. 2 points: The text contains a few grammatical or spelling errors, slightly affecting readability, but the overall meaning of the text is understandable. 3 points: The text is grammatically and orthographically correct, expressing fluently and naturally, easy to understand.
(2) Coherence	
Definition	Assess whether the question is logically clear and articulated explicitly. The scoring range is 1-3.
Score	1 point: The sentence lacks logical structure, is expressed in a disorganized manner, making it difficult for readers to understand. 2 points: The sentence has a basic logical structure, with a relatively clear theme or argument, but the expression may not be direct enough or some parts may be slightly vague, affecting overall clarity. 3 points: The question or answer has a clear structure, is logically coherent, expressed directly and clearly, easy to understand, and effectively conveys the theme or argument.
(3) Factuality	
Definition	Score based on whether [text] contains multiple factual errors, generally conforms to facts but contains minor errors or inaccuracies, or is entirely based on facts with all provided information being accurate. The scoring range is 0-1
Score	0 point: The text contains multiple factual errors or significant inaccuracies, making the information misleading or incorrect. 1 point: The text is entirely factually accurate, with all provided information verified as correct.
(4) Accuracy	
Definition	Score the category accuracy considering if the label category matches the content, ranging from 0-1.
Score	0 point: The assigned label does not match the content or is misleading. 1 point: The assigned label accurately reflects the content.

Table 13: Guideline of human evaluation for data quality.