CLEAN–EVAL: Clean Evaluation on Contaminated Large Language Models

Anonymous ACL submission

Abstract

We are currently in an era of fierce competition among various large language models (LLMs) continuously pushing the boundaries of benchmark performance. However, genuinely assessing the capabilities of these LLMs has become a challenging and critical issue due to potential data contamination. In this paper, we propose a novel and valuable method, *Clean-Eval*, which mitigates the issue of data contamination and evaluates the LLMs more cleanly. *Clean-Eval* employs a neural-based model to paraphrase and back-translate the contaminated data into a candidate set, generating expressions with the same meaning but in different surface forms. A semantic detector is then used to filter those generated low-quality samples to narrow down this candidate set. Candidates with moderate BLEURT scores against the original samples are selected as the final evaluation set. According to human assessment, this set is almost semantically equivalent to the original contamination set but expressed differently. We conduct experiments on 20 existing benchmarks across diverse tasks, and results demonstrate that *Clean-Eval* substantially restores the actual evaluation results on contaminated LLMs under both few-shot learning and fine-tuning scenarios. We will later be open-sourced as a website to fairly measure LLMs.

1 Introduction

In recent years, LLMs have made breakthroughs in handling complex and nuanced scenarios, achieved superior performance in some professional and academic benchmarks, and attracted many resources from industry and academia (OpenAI, 2023; Touvron et al., 2023; Golchin and Surdeanu, 2023). This subsequently opens the arms race era of LLMs, and various LLMs are continuously launched, such as GPT-4 (OpenAI, 2023), LLama2 (Touvron et al., 2023) and other LLMs, which have refreshed various evaluation benchmarks continuously.

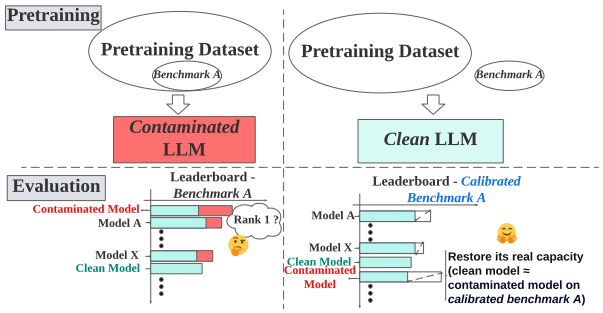
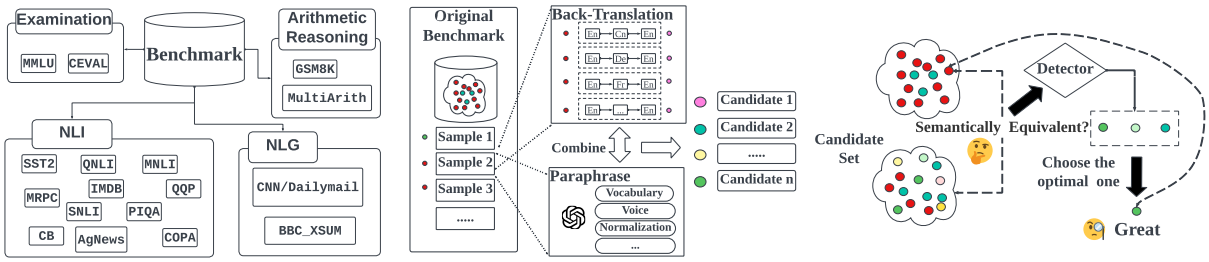


Figure 1: Data contamination happens when Benchmark A is included in the pretraining data, leading to inflated performance metrics like top leaderboard rankings. This can cause a clean model to lag behind the contaminated one. Our goal is to revise Benchmark A, preserving its meaning but changing its surface forms. This aims to re-evaluate the contaminated model, aiming to align its performance closer to that of a clean model.

There is room for doubt regarding the potential overestimation of these benchmark measurements. One reason is that LLMs are trained on data extracted from websites and publicly accessible datasets (OpenAI, 2023; Touvron et al., 2023). Therefore, ensuring no overlap between the pretraining dataset and the evaluated benchmark becomes quite a challenge. This subsequently introduces a significant concern: the risk of data contamination.

Data contamination arises when a model's pretraining data integrates evaluated data, consequently enhancing test performance (Magar and Schwartz, 2022; Golchin and Surdeanu, 2023). Currently, many models opt not to disclose their training sets in technical reports, raising concerns about the potential inclusion of benchmark datasets within their training data. This presents an urgent problem (Wei et al., 2023), as these contaminated models claim highly evaluated results but often lead to poor real-world experiences. We strongly advocate for a cleaner evaluation of LLMs. Unveiling the genuine capabilities of LLMs could sig-



(a) Data Collection. (b) Paraphrase and Back-translation. (c) Semantic Filter.

Figure 2: An overview of our method. We first gather established benchmarks for LLM assessment and then meticulously clean contamination in these benchmarks through LLM-powered paraphrase and multi-language back-translation, employing a semantic detector to filter and select optimal results based on BLEURT scores.

nificantly propel the community of LLMs forward. The most effective resolution involves relabeling a new dataset when developing a new model to assess its capabilities. Unfortunately, this process demands considerable time and labor.

This paper employs previously proposed benchmarks to create a new benchmark, and our method is called *Clean-Eval*, aiming to mitigate data contamination using LLMs and accurately assess the capabilities of LLMs. Leveraging the exceptional creative capabilities of these models, we perform diverse paraphrasing of contaminated data and back-translate it across multiple language directions. This process results in a pool of calibrated datasets. We effectively filter out low-quality samples by utilizing semantic detectors, and then select the best items based on BLEURT scores derived from comparisons between the calibrated and contaminated data. Finally, We conducted experiments on 20 benchmarks across diverse tasks, and our analysis unveiled noticeable calibrated effects achieved through *Clean-Eval*. Our human evaluation reinforces the method's potential to improve sentence structure, grammar, and linguistic diversity while maintaining core semantics. Acknowledging the challenge of detecting model contamination within specific benchmarks, we propose a new evaluation approach for in-context learning and fine-tuning. Our experiments convincingly demonstrate that processing contaminated data through our method effectively restores the model's genuine performance.

2 Related Work

2.1 Data Contamination

Detecting data contamination is crucial in ensuring the integrity of model training and usage. Researchers and practitioners within the field have dedicated considerable effort to developing methods for identifying and mitigating instances where test data unintentionally becomes part of a model's training dataset Brown et al. (2020); Touvron et al. (2023) .

Model Trainers. Brown et al. (2020) conducted experiments on data contamination, using an n-gram overlap metric to evaluate duplication levels between training and test sets. They subsequently eliminated these duplications from the training dataset. Similarly, Dodge et al. (2021) assessed exact matches, accounting for capitalization and punctuation normalization. This method scrutinized whether entire evaluation text inputs existed within the training data. However, Touvron et al. (2023) critiqued the precision of previous high-order n-gram-based detection methods in determining contamination extent within a sample. Their proposed approach involved token-level contamination identification, allowing for slight variations in overlap positions between evaluation samples and training data. Wei et al. (2023) took a distinctive approach, comparing the language model (LM) loss between the test splits of a dataset and a mimic dataset generated by GPT-4 to correspond to it. A smaller discrepancy value between these sets indicated potential contamination within the model.

Model Users. Carlini et al. (2023) construct a set of prompts using the model's training data. They investigated by supplying prefixes of these prompts to the trained model to assess the model's capacity to complete the remaining portion of the example verbatim. Their study revealed that as the model's capacity, duplicated numbers, and context length increased, the models would be more proficient in memorizing data. Meanwhile, Golchin and Sur-

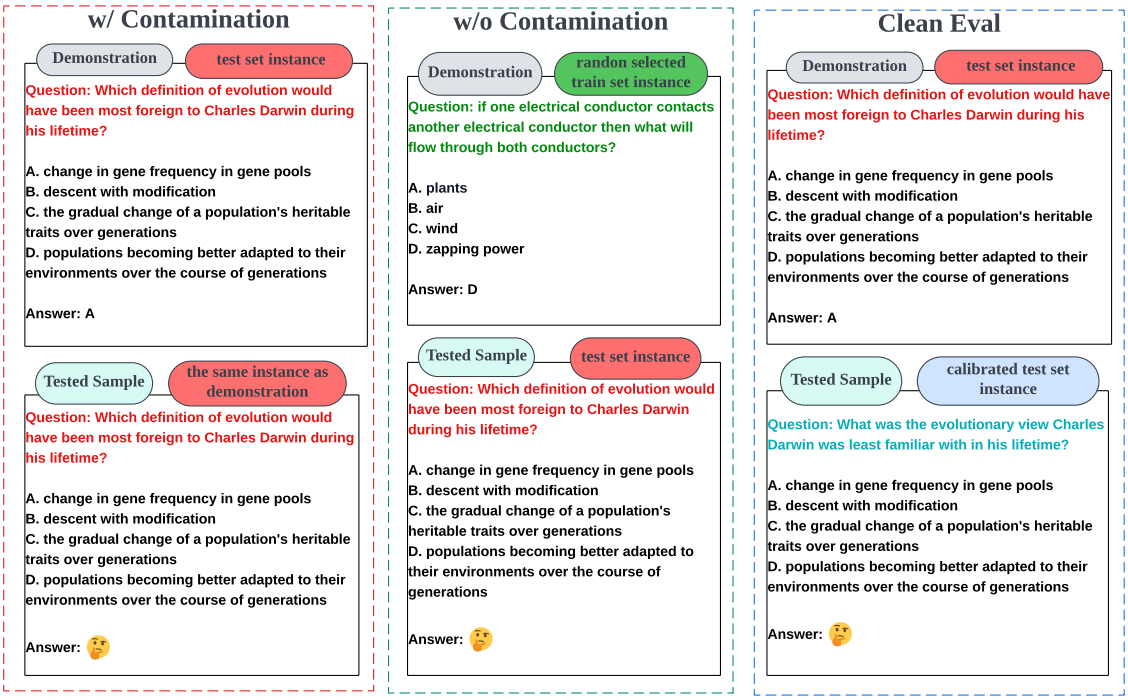


Figure 3: Evaluation setting of in-context learning. Each input comprises a demonstration and a tested sample. In the contamination setting, the demonstration matches the tested sample. In contrast, in the absence of contamination, the demonstration is drawn from a separate split of the dataset, maintaining distinction from the tested sample (e.g., sampled from the train split). In our Clean-Eval setup, the tested sample is a calibrated version of the demonstration, specifically designed to mitigate the effects of contamination.

deanu (2023) introduced an approach involving the development of guided instructions that include the initial segment of a data instance and its corresponding partition name. These guided instructions are subsequently utilized to induce the model to generate the second part of the data, based on a provided prompt. Rouge, BLEURT, and GPT4 auto evaluation determine whether the model had data contamination. Furthermore, Li (2023) analyzed six prominent multi-choice QA benchmarks, quantifying their overlap with the training dataset already known of Llama to detect potential data contamination.

2.2 Existing Benchmark

Many benchmarks have been proposed, including MMLU (Li et al., 2023a), CEVAL (Huang et al., 2023), etc., to measure the capability of LLMs comprehensively. However, labeling these benchmarks is time-consuming and laborious, and ensuring no overlap with the training set of LLM is often challenging. There is also work to reformulate existing benchmarks to build new ones. Li et al. (2023b) propose ReForm-Eval to reformulate existing benchmarks into unified large vision-language model compatible formats.

Nevertheless, based on our knowledge, there is no proposed solution to the problem of data contamination causing excessive model evaluation performance. In this paper, we propose an effective method to mitigate this problem. Experiments demonstrate that our methods work in evaluating both closed and open LLMs.

3 Clean-Eval

The framework of our method is shown in Figure 2. Our methodology comprises three primary stages. Initially, we concentrate on gathering established benchmarks to assess LLMs. In the subsequent phase, we meticulously cleaned contamination in the collected benchmarks. This involves paraphrasing samples using the creative capacities of the LLMs and performing multi-language back-translation on the contaminated data. In the final phase, we use the semantic detector to filter the outcomes of the contamination cleanup, eliminating subpar results and selecting the ultimate results based on the BLEURT score.

text-davinci-003 | In-context Learning | [Accuracy]									
	AG News	**QQP**	**QNLI**	**RTE**	**MNLI**	**WNLI**	**SNLI**	**IMDB**	**PIQA**
w/ Contamination	53.67	95.00	90.67	96.39	80.00	95.78	94.00	95.67	86.33
w/o Contamination	40.67	83.33	80.00	84.12	71.00	54.93	73.67	89.00	80.33
Clean-Eval	53.00 ↓	79.00 ↓	82.00 ↓	76.90 ↓	71.67↓	71.83 ↓	62.00 ↓	85.33 ↓	75.33 ↓
	MultiArith	**MRPC**	**GSM8K**	**COPA**	**CB**	**BOOLQ**	**SST2**	**MMLU**	**CEVAL**
w/ Contamination	65.00	93.67	64.33	92.00	98.21	87.33	90.67	73.67	66.33
w/o Contamination	35.00	68.33	12.33	90.00	82.14	81.33	80.00	59.00	41.00
Clean-Eval	60.00↓	65.67↓	50.67 ↓	75.00↓	91.07 ↓	83.67 ↓	78.00↓	57.00 ↓	38.33 ↓
Llama2 | Fine-Tuning | [Accuracy]									
	AG News	**QQP**	**QNLI**	**RTE**	**MNLI**	**WNLI**	**SNLI**	**IMDB**	**PIQA**
w/ Contamination	54.00	99.00	98.00	99.27	99.67	63.38	99.00	97.33	100.00
w/o Contamination	31.67	84.00	85.67	80.51	72.00	47.89	82.00	94.00	74.33
Clean-Eval	51.34 ↓	81.00 ↓	79.00 ↓	67.87↓	73.67 ↓	60.56 ↓	68.37 ↓	95.33 ↓	78.67↓
	MultiArith	**MRPC**	**GSM8K**	**COPA**	**CB**	**BOOLQ**	**SST2**	**MMLU**	**CEVAL**
w/ Contamination	36.11	96.33	50.67	100.00	85.71	99.33	99.99	82.67	87.33
w/o Contamination	16.11	79.33	7.00	89.00	58.93	73.33	94.67	37.33	30.00
Clean-Eval	22.78 ↓	60.33 ↓	26.33↓	76.00 ↓	71.43 ↓	91.33↓	90.67 ↓	25.00 ↓	85.00 ↓

Table 1: Natural language understanding tasks. The symbol ↓ indicates a decrease in performance compared to the contamination setting. The optimal candidate is chosen according to the lowest BLEURT score.

3.1 Back-translation

Back-translation (BT) involves retranslating content from the target language into its source language using literal terms (Sennrich et al., 2016). In this process, slight differences can be introduced, such as replacing synonyms. Therefore, we translate the raw data into various language orientations and then revert to the original language to compose our candidate set of contamination cleanup data. In this process, we aim to achieve a distinct expression from the original sample while preserving the semantics.

3.2 Paraphrase

LLMs have showcased significant potential across diverse professional domains, particularly in creative writing (Touvron et al., 2023). Harnessing their creative prowess, we utilize LLMs to generate multiple paraphrases of raw data, purposefully introducing variations. Specifically, we leverage the text-davinci-003 version of GPT-3 to generate these paraphrases. For instance, a typical prompt in our approach was: *Please paraphrase this sentence in three different ways.*

3.3 Filter

However, these candidate sets might need to be further examined to ensure their quality. As shown in Figure 2c, we use a semantic detector to judge whether the content in the candidate set is semantically similar to the original content to narrow the set of candidate sets further and select the candidate according to the BLEURT score as the final result.[1] In Appendix B.3, the BLEURT scores of each instance on various benchmarks are presented, with scores typically ranging from 0.4 to 0.9. Our analysis indicates that the lowest BLEURT score serves as an effective indicator for restoring the true capabilities of LLMs.

With these essential steps, we have achieved greater efficiency in harnessing existing datasets, mitigated data contamination concerns, and furnished high-calibrated new data suitable for evaluating model performance.

4 Evaluation Setting

Nearly all LLMs operate with proprietary training datasets, making it challenging to ascertain whether

[1]This detector is optional. Removing the detector saves computational and token costs, but can potentially degrade the quality of the selected candidates.

	CNN/Daily-Mail			BBC-XSUM		
	Rouge-1	**Rouge-2**	**Rouge-L**	**Rouge-1**	**Rouge-2**	**Rouge-L**
w/ Contamination	23.38	9.45	21.69	33.64	18.56	29.2
w/o Contamination	21.18	7.18	19.57	22.97	7.78	19.08
Clean-Eval	23.14 ↓	9.35 ↓	21.41 ↓	33.09↓	17.92↓	28.90 ↓

Table 2: ICL experiments and metrics in Rouge. ↓ is compared to the contamination dataset. The optimal candidate is chosen according to the lowest BLEURT score.

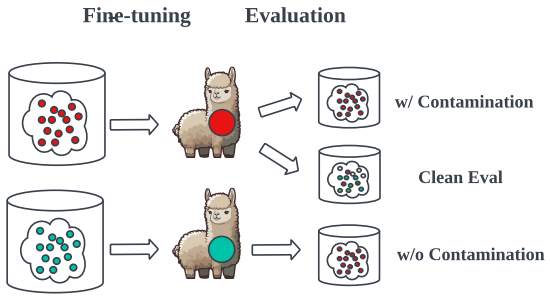


Figure 4: Evaluation setting of fine-tuning. We fine-tuned two models using datasets labeled red and green. When evaluated on the red dataset, these two models are categorized as contaminated and uncontaminated. Testing a model's performance on the red dataset processed by *Clean-Eval* is attributed to the Clean-Eval setting.

the data being tested is free from contamination. To address this issue, we introduce an experimental framework for simulating data contamination.

4.1 In-context Learning

In-context learning (ICL) involves presenting a task demonstration to the model as a part of a natural language prompt. According to Brown et al. (2020), LLMs are classified as few-shot learners. Due to restricted access to the GPT-3 model and its variability, we execute ICL on these models to assess the efficacy of *Clean-Eval*. Within the ICL scenario, we propose and compare three evaluation settings: contamination, without contamination, and clean evaluation for any given benchmark.

Each input comprises a demonstration and a tested sample, with different evaluation settings contingent upon their constitution. The demonstration matching the tested sample, as depicted on the left side of Figure 3, constitutes the **contamination setting**. When the demonstration and tested sample originate from different dataset splits (center of Figure 3), it is categorized as the **without contamination setting**. In contrast, when the tested sample is the demonstration processed by *Clean-Eval* (right side of Figure 3), it represents the **Clean-Eval setting**.

4.2 Fine-tuning

Fine-tuning entails further optimization adjustments for a specific task or dataset using a pre-trained LLM. Illustrated in Figure 4, we fine-tune two models using distinct splits of a dataset.

Each instance within a benchmark is formatted as an instruction for fine-tuning the model. When the evaluation data mirrors the fine-tuned data, it's categorized as **the contamination setting**. If the evaluation and fine-tuned data originate from different splits of the same dataset, it falls under the **without contamination setting**. Lastly, when the evaluation data is fine-tuned data processed by *Clean-Eval*, it represents our **Clean-Eval setting**.

5 Experiments

5.1 Datasets

We have meticulously curated 20 datasets, spanning a wide array of tasks. These tasks encompass text implication, problem pair matching, natural language reasoning, semantic similarity, sentiment analysis, common sense reasoning, text classification, mathematical reasoning, examinations, and even some natural language generation tasks. This classification provides valuable insights into the performance of various task types concerning data contamination. Below is the comprehensive list of datasets we have utilized:

- **Nature Language Inference.** GLUE dataset (Wang et al., 2019b) that includes QNLI, MNLI, SNLI, WNLI, RTE, QQP, MRPC, SST2; IMDB (Maas et al., 2011); BOOLQ (Clark et al., 2019); Super-GLUE dataset (Wang et al., 2019a) that includes COPA, CB; Ag News (Zhang et al., 2015).
- **Nature Language Generation.** CNN_Dailymail (See et al., 2017),

Data	Method	Rouge-1	Rouge-2	Rouge-L	BLEURT	Equivalence
QNLI	Back-translation	54.08	29.80	50.47	63.44	100.00
	Paraphrase	48.50	26.02	43.28	63.19	100.00
	Clean-Eval	**46.85**	**22.90**	**42.53**	**60.21**	100.00
SST2	Back-translation	52.35	32.05	51.01	59.94	100.00
	Paraphrase	30.39	9.98	27.77	42.96	100.00
	Clean-Eval	**26.66**	**7.55**	**23.64**	**40.90**	100.00
MMLU	Back-translation	52.85	30.15	48.68	57.91	100.00
	Paraphrase	45.71	23.10	40.79	55.46	100.00
	Clean-Eval	**42.42**	**19.70**	**38.32**	**51.99**	100.00

Table 3: The difference between the sample processed with different methods and the original sample. We choose the lowest BLEURT score as our optimal candidate. As all generated samples undergo semantic detection, their semantic equivalence consistently reaches 100%.

BBC_XSUM (Narayan et al., 2018).

- **Arithmetic Reasoning.** GSM8K (Cobbe et al., 2021), MultiArith
- **Examination.** MMLU (Hendrycks et al., 2021), CEVAL (Huang et al., 2023).

5.2 Metrics

ROUGE & BLEURT. To measure the degree of overlap between a generated instance and a reference, we utilize both ROUGE (Lin, 2004) and BLEURT scores (Sellam et al., 2020). ROUGE evaluates lexical similarity, focusing on shared words and phrases, while BLEURT assesses the semantic relevance and fluency of the generated sequence concerning the reference instance.

Equivalence. We employed the text-davinci-003 model (Brown et al., 2020) to assess equivalence before and after the processing of contaminated data by *Clean-Eval*. Details of the prompt designs can be found in Appendix B.1.

5.3 Contamination Cleanup.

Models. We employ the text-davinci-003 model (Brown et al., 2020) for paraphrasing, back-translation, and semantic detection purposes. Additionally, we utilize the BLEURT-20 model (Sellam et al., 2020) to compute BLEURT scores and then select the optimal candidate.

Process. Given the diversity in format and content across datasets, our processing criteria vary accordingly. Resource constraints prevent comprehensive processing of every dataset aspect within our method, *Clean-eval*. For instance, while we thoroughly handle all contents in datasets like SNLI paired datasets, our focus narrows to questions alone in question-options-answer or question-answer datasets. Additionally, our analysis is limited to the initial three sentences or less in dealing with lengthy text. Furthermore, all generated samples undergo semantic detection. If they fail this detection, the original sample is output.

Results. The results are shown in Table 3. Following our *Clean-Eval* method, the surface form of the newly generated sample notably differs from the original sample, particularly in terms of n-gram variations. However, the presence of the semantic detector ensures the quality and fidelity of the generated results, assuring their reliability despite these surface-level alterations.

5.4 In-context Learning

Model. We use the text-davinci-003 model (Brown et al., 2020) to conduct ICL experiments.

Implementation Details. Each tested use case is provided with task-specific instructions. For instance, one instance attributed to CNN/Dailymail would receive a prompt such as "The task is to summarize this article:". Detailed designs for all prompts are in Appendix B.2.

Results and Analysis The results displayed in Table 1 and Table 2 consistently showcase superior performance across all tasks in the presence of data contamination, surpassing both the no-contamination and Clean-Eval settings. This emphasizes a distinct performance advantage influenced by data contamination. Notably, the model demonstrates robust generalization across simpler tasks like RTE, IMDB, and QQP, evident from its strong performance even in the absence of con-

tamination. However, when contamination occurs in these tasks, the model sustains a near-optimal performance level.

The Clean-Eval setting is reliable, revealing the model's genuine capability. Many datasets exhibit performance levels close to those without contamination. Yet, a performance gap between the no-contamination and Clean-Eval settings still exists, especially in more intricate tasks involving mathematical reasoning, such as GSM8K and MultiArith. The model's reduced performance in the no-contamination setting might stem from a lack of chain of thought, leading to performance degradation. Moreover, as depicted in Table 2, our approach effectively mitigates data contamination, even when limiting processing to the first three sentences or fewer in an article. All results indicate that employing our *Clean-Eval* method results in a gradual performance decline, aligning more closely with the no-contamination setting.

5.5 Fine-tuning

Model. For fine-tuning, we employ the LLama2-7b-chat model (Touvron et al., 2023).

Implementation Details As model parameters grow in size, achieving full fine-tuning becomes increasingly challenging. In such scenarios, we resort to LoRA for fine-tuning (Hu et al., 2021). Additional experiment settings are detailed in Appendix A. Our process commences by transforming original data into instructional data, followed by single-instruction fine-tuning. Considering the extensive array of datasets, conducting exhaustive fine-tuning for each model to attain optimal performance would be impractical and time-consuming. Thus, we fine-tune the model for approximately 40 epochs before assessing its performance.

Results and Analysis When the model undergoes fine-tuning and subsequent performance testing using the same dataset, it achieves notably higher accuracy, even reaching 100% on specific datasets. However, this performance dips when evaluated on a different dataset split. A significant performance gap exists between the uncontaminated and contaminated dataset settings, particularly in challenging tasks like MultiArith, GSM8k, MMLU, and CEVAL. Notably, when tested under a Clean-Eval setting, the model's performance aligns closely with that of the uncontaminated data.

6 Analysis

6.1 Ablation Study

In Table 3, we conducted an ablation study comparing three methods, including back-translation, paraphrase, and *Clean-Eval*. Back-translation consistently yields higher Rouge and BLEURT scores than other methods across three datasets. This suggests that back-translation is effective in maintaining lexical and sentence structure from the original text. Paraphrase introduces variations in content expression, showcasing the ability to offer alternative ways of expressing the same semantic content. *Clean-Eval*, which combines paraphrase and back-translation, emerges as a comprehensive approach. It maintains semantic equivalence, as indicated by the Equivalence score, and enhances the diversity of content expression.

6.1.1 BLEURT Score

In this part, we explored whether the selection based on the BLEURT score impacts the model performance.

	Score	QNLI	SST2	MMLU
BT	lowest	**7.33**	**6.00**	**14.67**
	median	-5.99	6.00	3.33
	highest	-10.67	6.00	4.01
Para	lowest	-8.67	**6.00**	8.01
	median	**4.01**	4.66	5.33
	highest	0.01	6.00	**8.67**

Table 4: In ICL experiments, we assess the performance gap using various BLEURT scores. This gap represents the difference in performance between the model tested in the Clean-Eval setting versus the no-contamination setting and the model tested in the contamination setting versus the Clean-Eval setting. A higher value signifies that Clean-Eval approaches performance levels akin to those in the no-contamination setting.

Results. Table 4 illustrates that paraphrasing exhibits variability across three datasets. However, back-translation demonstrates the potential to bring the performance of the model closer to that of the no-contamination setting when choosing the lowest BLEURT score. Hence, to restore the large model's capabilities, selecting the best candidate based on the lowest BLEURT score might be a viable strategy.

6.1.2 Combination sequence

We conducted a comparison to evaluate the performance impact of the sequence of paraphrasing and back-translation.

Order	QNLI	SST2	MMLU
Para + BT	**10.67**	**6.00**	**14.67**
BT + Para	10.67	6.00	12.67

Table 5: Performance gap with different combination orders of paraphrase and back-translation.

Results. From Table 5, we can see that while QNLI and SST2 tasks are less sensitive to method order, the MMLU task shows slight differences. Therefore, we can tailor the order based on task requirements and we choose to first paraphrase and then back-translation in *Clean-Eval*.

6.1.3 Equivalence Detector

Continuous use of back translation would end up with a string that differs markedly from that which you started (Way, 2013). A combination of paraphrase and back-translation might also cause this problem.

Method	QNLI	SST2	MMLU
BT	74.17	86.33	73.33
Para	91.67	82.67	73.33
Clean-Eval	72.17 ↓	56.34↓	60.34↓

Table 6: Model performance on the calibrated dataset without equivalence detector.

Results. As we can see from Table 6, across all three datasets, the paraphrasing method demonstrates relatively high performance, especially in QNLI and SST2. In the absence of a semantic detector, results generated through *Clean-Eval* exhibit a general decline in performance. This suggests the possibility of introducing semantic errors or inaccuracies during the generation process and the importance of semantic detectors.

6.2 Human Evaluation

We performed human evaluations of the generated output to assess potential changes after our method *Clean-Eval*.

Results. Human evaluation results on the SST2 dataset indicate that 97% of instances maintain semantic equivalence with the original ones. This suggests the model largely preserves the intended meaning of the questions, showcasing the effectiveness of the generated output in retaining input semantics.

7 Case Study

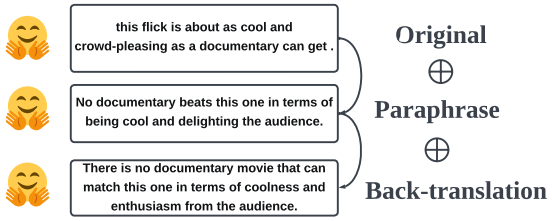


Figure 5: A case study from SST2 dataset.

In this case, the paraphrased sentence successfully conveys the essence of the original while introducing some variation. The transformation maintains a positive sentiment, emphasizing the documentary's coolness and appeal to the audience. Back-translation aims to ensure that the paraphrased sentence retains its intended meaning. The back-translated sentence aligns well with the paraphrased version. The key elements, such as the documentary's uniqueness, coolness, and audience appeal, are preserved. The combined approach of paraphrasing and back-translation proves effective in enhancing the original sentence. The paraphrased version introduces a nuanced expression, and the subsequent back-translation successfully captures the intended meaning. The final output maintains a positive tone and successfully communicates the documentary's appeal.

8 Conclusion

Data contamination is an urgent problem for the development of LLMs society. Downloading and trying contaminated models can be a waste of time for both researchers and developers. To save their time, this paper intends to mitigate the issue of data contamination in LLMs through the introduction of the *Clean-Eval* method. This approach leverages existing datasets to create a new evaluation dataset, effectively mitigating the impact of contamination. Experimental results demonstrate the method's success in accurately assessing model capabilities and restoring real performance. *Clean-Eval* holds promise in enhancing transparency and reliability in the evaluation of LLMs.

Limitations

Datasets. This paper focuses on two mainstream models. In the absence of knowledge regarding their training data, our selected benchmark, mimicking the no-contamination setting, likely overlaps with their existing training data. Consequently, performance testing on these benchmarks could yield inflated performance metrics. Moreover, due to resource constraints, we sampled approximately 300 instances for each benchmark. However, despite this limited number, randomness in sampling aims to ensure these instances represent the entire dataset.

Fine-tuning. Given the extensive collection of benchmarks, conducting exhaustive fine-tuning to maximize model performance becomes impractical. Instead, we fine-tune the model using a consistent experimental setup for approximately 40 epochs. Our goal is to illustrate that models affected by contamination exhibit higher performance. Furthermore, evaluating benchmarks processed by our method *Clean-Eval* aims to mitigate this performance inflation and restore the true capabilities of the LLMs.

Ethic Statement

This paper will not pose any ethical problems. The datasets used in this paper have already been used in previous articles.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus.

Shahriar Golchin and Mihai Surdeanu. 2023. Time travel in llms: Tracing data contamination in large language models.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese.

Yucheng Li. 2023. An open source data contamination report for llama series models.

Zejun Li, Ye Wang, Mengfei Du, Qingwen Liu, Binhao Wu, Jiwen Zhang, Chengxing Zhou, Zhihao Fan, Jie Fu, Jingjing Chen, Xuanjing Huang, and Zhongyu Wei. 2023b. Reform-eval: Evaluating large vision language models via unified re-formulation of task-oriented benchmarks.

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.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Andy Way. 2013. Emerging use-cases for machine translation. In *Proceedings of Translating and the Computer 35*.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. 2023. Skywork: A more open bilingual foundation model.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

A Experiment Settings

We conducted fine-tuning of the Llama2-7b-chat version on 2 RTX4090 GPUs, each with 24GB of memory. The model was fine-tuned according to specific instructions, utilizing the following prompt:

```
[INST] <<SYS>>\n"
"You are a helpful, respectful, and honest assistant."
"<</SYS>>\n\n{0} [/INST]\n{1}</s>"]
```

To optimize memory usage and enable deployment on smaller devices, we loaded our Llama2-7b-chat model in 4-bit precision, effectively reducing memory consumption. Employing a bfloat16 compute data type alongside nested quantization further contributed to memory efficiency. Additionally, we leveraged LoRA with a 16-dimensional updated matrix and scaling set at 64. A batch size of 16 was chosen for shorter instructions, while longer instructions used a batch size of 4. The initial learning rate was set to 2e-4, coupled with the paged_adamw_8bit optimizer for training.

B Prompt Design

B.1 Method prompt

Our paraphrasing, back-translation, and equivalence detector prompts are shown in Table 7.

B.2 Instruction for Each Dataset

Our prompts for each benchmark are shown in Table 8.

Method	Prompt Design
Paraphrase	Please paraphrase the following sentence without changing the meaning in 3 ways, and then return as a list.
Back-translation	Please translate the following sentence into *[language]* without changing the meaning.
Equivalence Detector	Please determine whether the following sentences are equivalent.

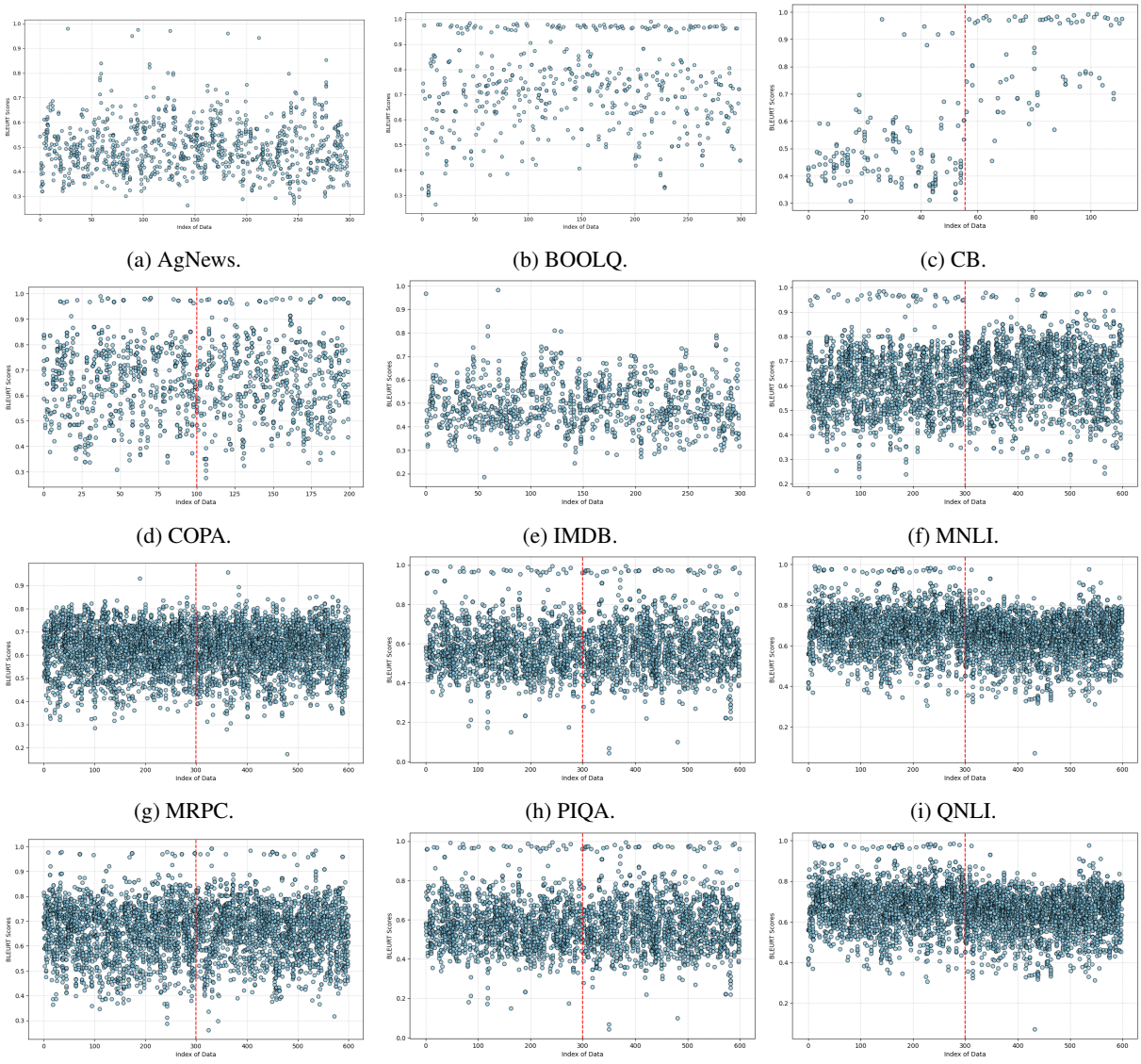
Table 7: Prompt designs of each method.

Dataset	Prompt Design
RTE	The task is to determine whether a pair of sentences are entailed by each other. Just return entailment or not_entailment.
QQP, MRPC	The task is to determine whether a pair of questions are semantically equivalent. Just return equivalent or not_equivalent.
QNLI	The task is to determine whether the context sentence contains the answer to the question. Just return entailment or not_entailment.
MNLI, CB	The task is to predict whether the premise entails the hypothesis, contradicts the hypothesis, or neither. Just return entailment, contradiction, or neutral.
WNLI	The task is to predict if the sentence with the pronoun substituted is entailed by the original sentence. Just return entailment or not_entailment.
SNLI	The task is to determine whether a pair of sentences are entailed, contradicted, or neutral to each other. Just return entailment, contradiction, or neutral.
IMDB	The task is to determine whether the sentiment of the text is positive or negative. Just return positive or negative.
PIQA	The task is to select the best solution to the question. Just return the solution1 or solution2.
COPA	Given a premise, choose one of the following two choices that express the sample["question"] relationship. Just return choice1 or choice2.
BOOLQ	The task is to answer true or false given the question. Just return true or false.
SST2	The task is to determine whether the sentiment of the sentence is positive or negative. Just return positive or negative.
AG News	The task is to classify the article into sports, world, business, or sci/tech. Just return sports, world, business, or sci/tech.
GSM8K, MultiArith	The task is to answer a given mathematical question. Just directly return the final number answer.
MMLU, CEVAL	Please select the best answer from the options according to the question. Just return one answer with A, B, C, or D.
CNN_Dailymail, BBC_XSUM	Please summarize this article.

Table 8: Prompt designs of each benchmark.

B.3 BLEURT Score

Figure 6 illustrates the BLEURT score of each instance from selected benchmarks compared to the original instance.



(a) AgNews. (b) BOOLQ. (c) CB.

(d) COPA. (e) IMDB. (f) MNLI.

(g) MRPC. (h) PIQA. (i) QNLI.

(j) QQP. (k) PIQA. (l) QNLI.

Figure 6: The BLEURT score of each instance from selected benchmarks compared to the original instance. The graph featuring the red line represents a paired dataset, depicting one instance on either side of this demarcation