# Confounders in Instance Variation for the Analysis of Data Contamination

**Behzad Mehrbakhsh**[a,b,c;*], **Darío Garigliotti**[d],
**Fernando Martínez-Plumed**[a,b] and **José Hernández-Orallo**[a,b,c]
[a]UPV - Universitat Politècnica de València
[b]VRAIN - Valencian Research Institute for Artificial Intelligence
[c]ValgrAI - Valencian Graduate School and Research Network of Artificial Intelligence
[d]UiB - University of Bergen

## Abstract

Test contamination is a serious problem for the evaluation of large language models (LLMs) because it leads to the overestimation of their performance and a quick saturation of benchmarks, even before the actual capability is achieved. One strategy to address this issue is the (adversarial) generation of variations, by including different exemplars and different rephrasings of the questions. However, these two interventions can lead to instances that can be more difficult (accumulating on the expected loss of performance by partly removing the contamination) but also to instances that can be less difficult (cancelling the expected loss of performance), which would make contamination undetectable. Understanding these two phenomena in terms of instance difficulty is critical to determine and measure contamination. In this paper we conduct a comprehensive analysis of these two interventions on an addition task with fine-tuned LLAMA-2 models.

## 1 Introduction

Large Language Models (LLMs) (Brown et al., 2020) have transformed Natural Language processing, but face evaluation challenges, especially with publicly available benchmarks. A key issue is *data contamination* (Ravaut et al., 2024), where training data contains test instances. A model trained on this data has seen part of the test set, which can have important effects in the evaluation of the model, leading to inflated performance measures (Yang et al., 2023; Sainz et al., 2023).

Recently, it has been seen that the actual issue of data contamination could be more predominant, as it is not only present by exact copies of the test instances in the training data. Studies such as the one by Yang et al. (2023) show that even rephrased or translated test instances in training data can improve performance, indicating potential contamination. However, the effect of the *difficulty* of these rephrased items has not been yet investigated. For instance, rephrasing can involve more convoluted or unusual expressions, which make the item more difficult for language models. Table 1 shows a series of *templates* that can be used to rephrase the expression behind the task of adding two numbers. The actual *exemplar* (the pair of $[term_1]$ and $[term_2]$) can also be replaced by a different pair to avoid contamination. These two interventions (different rephrasing or exemplar) can have mixed effects: some variations may inadvertently increase the difficulty of the instances, leading to an expected drop in performance, masking contamination, while others may make the instance easier and leading to an overestimation of performance, leading to false positives.

|  | **Template** |
|---|---|
| 1 | Can you please add $[term_1]$ and $[term_2]$ together? |
| 2 | Find the sum of $[term_1]$ and $[term_2]$. |
| 3 | Add up two numbers: $[term_1]$ and $[term_2]$. |
| 4 | Please work out the total of $[term_1]$ and $[term_2]$. |
| 5 | Please determine the numeric sum of $[term_1]$ and $[term_2]$. |
| 6 | Proceed to identify the aggregated total of the numbers $[term_1]$ and $[term_2]$. |
| 7 | Perform an addition operation on the numerical values $[term_1]$ and $[term_2]$. |

Table 1: Various templates created by GPT-4 for the addition task. By instantiating them with different exemplars, we can get different instances, such as ‘Find the sum of 56 and 723’ and ‘Perform an addition operation on the numerical values 35 and 85’. Are these two instances equally difficult?

Understanding these phenomena in terms of instance difficulty is a novel approach for accurately identifying and measuring contamination. In this study, we conduct a thorough analysis of rephrased templates and replaced exemplars using fine-tuned LLAMA-2 (Touvron et al., 2023) models for an

[*] Corresponding author. Email: bmehrba@upv.es .

*Proceedings of the 1st Workshop on Data Contamination (CONDA)*, pages 13–21
August 16, 2024 ©2024 Association for Computational Linguistics

addition task. We generate templates of varying difficulty using GPT-4 (Achiam et al., 2023), fine-tune the models, and assess how these variations affect performance on exemplars of varying number of digits (and hence difficulty).

The main contributions are:

- We investigate how different rephrased and replaced test instances impact the performance of fine-tuned LLAMA-2 models on an addition task, revealing critical insights into the effects of data contamination.
- We study the impact of template difficulty on model performance, highlighting that variations in test instance phrasing can significantly affect evaluation outcomes.
- We evaluate the effects of fine-tuning with easy versus hard templates, showing how template diversity and intrinsic difficulty influence model performance and contamination detection.

The following sections detail our experimental design, methodology, and results.

## 2 Background

Several methods have been used to address data contamination. Prior to big tech companies close sourced their models and the training data, a common approach was trying to look for evaluation instances in the training data. String matching and embedding similarity are two techniques that have been commonly used for this purpose. OpenAI used 8-gram matching of test instances and training dataset for GPT-2 model (Radford et al., 2019). For GPT-3 (Brown et al., 2020) the same approach has been taken and all data points from the evaluation sets that had a 13-gram collision in the pre-training Common Crawl (C4) dataset were removed to tackle contamination.

As contamination can involve minor variations of the examples, calculating cosine similarity between embeddings of test and training items can also be used for finding cases in which the test item has been rephrased or expressed in a different language (Gunasekar et al., 2023) (Riddell et al., 2024).

But string matching and even embedding matching are not able detect rephrased test items effectively in general (Yang et al., 2023). More sophisticated and effective techniques employ embedding similarity search to identify the top-$k$ samples similar to a given test sample and then prompting a powerful LLM such as GPT-4 to determine if any of the $k$ samples are too similar to the test case.

For closed source models where no information regarding the training set is provided, none of the above mentioned methods are applicable. Introducing new contamination-free benchmarks such as LastEval (Li, 2023), WIKIMIA (Shi et al., 2023), KIEval (Yu et al., 2024), LiveCodeBench (Jain et al., 2024), Termite (Ranaldi et al., 2024) might seem a reliable solution for the problem, but as (Balloccu et al., 2024) mentioned, these new benchmarks can get contaminated as soon as they are publicly available or even just when used for evaluating closed source models by the creators of the benchmark themselves for the first time. In addition, building a high quality benchmark is a time consuming process and can not be done overnight.

Consequently, the idea of continuously generating new variation has taken ground. Clean-Eval (Zhu et al., 2023) intends to 'purify' current benchmarks by rephrasing the test items. While a drop in the performance of LLMs on the rephrased data points is considered as a sign of decontamination, the role of difficulty has been neglected in their analysis.

## 3 Methodology

Data contamination occurs when instances from the test set are found in the train set of AI models. For example, if a model is tested on the question "*What is 123 + 456?*" but has seen the same question (and answer) during training, it might simply recall the answer rather than 'compute' it again. Even if rephrased forms of test items like "*Calculate the sum of 123 and 456*" or "*What do you get when you add 456 to 123?*" exist in the training data, the evaluation is still compromised. These rephrased forms can inadvertently aid the model, causing an overestimation of its true capabilities.

On the other hand, testing on the rephrased form of original test items is suggested by the researchers to mitigate the contamination problem. Yet to the best of our knowledge, the role of difficulty of original test items and their variants has not been studied. Also, what matters more, the change in the exemplar or rephrasing the template? For instance, solving "*Find the result of 9876 + 54321*" might naturally be harder than "*Compute 12 + 34*," regardless of rephrasing.

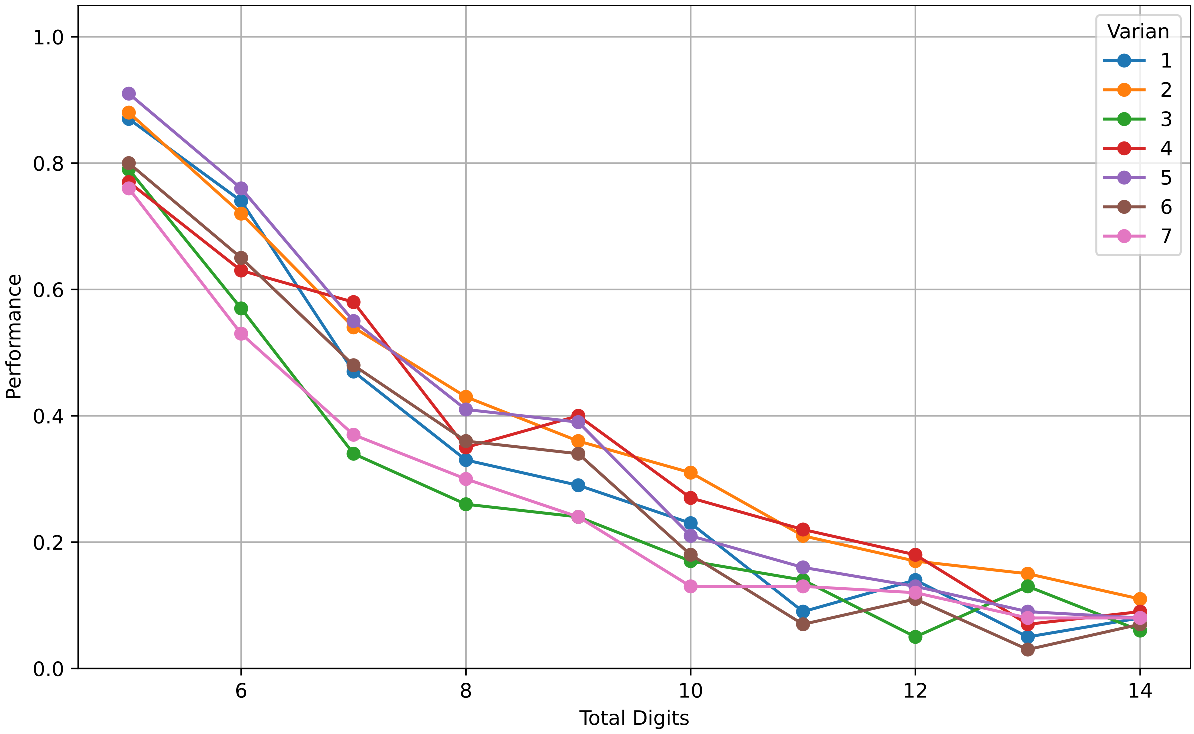

Figure 1: Performance of Llama2 7B chat model on the constructed dataset of addition.

**Dataset Construction** To explore these considerations, we designed a dataset of addition problems varying in complexity. Specifically, we generated 1,000 addition pairs (each different pair is an exemplar) for numbers ranging from five to fourteen digits. Each exemplar's intrinsic difficulty was determined by the sum of the number of digits of the addends. For example, the intrinsic difficulty of "829 + 4531" is 7.

**Instance templates** To produce varied instances, we asked GPT-4 to rephrase each addition problem in ten different ways. After excluding three ambiguous rephrasings, we used the remaining seven clear templates (see table 1). By applying seven templates to one thousand addition pairs, we generated seven thousand instances.

**Model Evaluation and Fine-Tuning** We used the Llama-2 7B chat model to check how different templates and exemplar affect model performance. First, we tested the model on the 7,000 instances (1,000 different exemplars per template) to get a baseline performance, as shown in Figure 1. We see some noticeable effect of the template (#2, orange, being much better than #7, pink), and a very significant influence of the #digits.

We performed three main fine-tuning experiments to explore how template variations affect model performance. In the first experiment, we used 70% of the exemplars (700), each with a different template, keeping a balanced representation of templates in the training data (equal number of exemplars, 100, for each template). The remaining 30% of the exemplars (300) was left for a non-contaminated validation set. Figure 2 shows the data construction process for our fist fine-tuning experiment.

The second experiment focused on the impact of template difficulty. We fine-tuned the model with either the easiest template (template 2) or the hardest template (template 7), based on initial performance evaluation (Figure 1). We then tested the fine-tuned models on all templates to see how this affected performance (Figure 5).

In the last experiment, we study the role of diversity of contaminating items. We compare the performance of the fine-tuned models when trained on four templates (#1, #2, #3 and #4) with the case only one of these templates is included in training, but repeated four times. (Figure 7)

In all cases we fine-tune Llama2 7B Chat model. Our fine-tuning process used the QLoRA method (Dettmers et al., 2024), implemented through the Huggingface pipeline. The choice of QLoRA allowed us to fit the entire Llama2 7B chat model within the memory constraints of a single NVIDIA

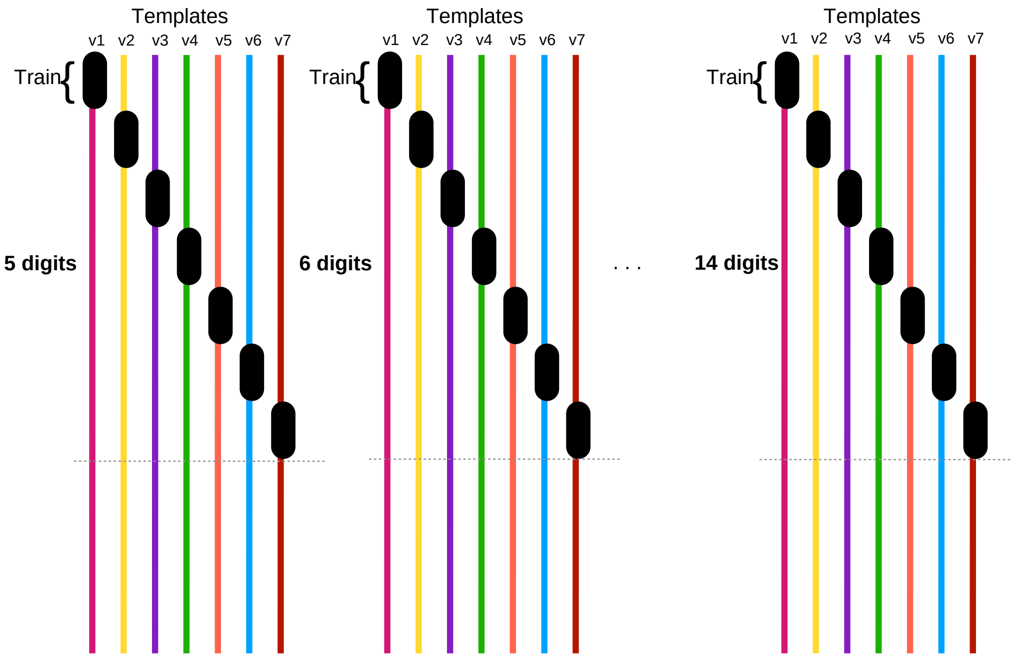

Figure 2: Data split for the first fine-tuning experiment. For each of the 10 digit lengths, 100 addition pairs (exemplars) are randomly generated. 70% are used for contaminating the training data. By applying 7 templates that are demonstrated with different colour in this figure, 490 instances can be created, one in seven (70) appearing in the train (shown in black) and the rest in the test. The 30 % of the original exemplars for this digit length are kept as non-contaminated validation set (below the dashed line) also with seven variations each in the test set (210).

GeForce RTX 3090 GPU with 24 gigabytes of RAM, making fine-tuning feasible for accessible hardware. The learning rate has been set to 1e-3 and batch size of 8 has been used. Through careful calibration, we found that this configuration provides an optimal balance, maximizing model performance while avoiding memory constraint issues. For the first and second experiments we fine-tuned the model for 5 epochs. For the third experiments 1 epoch is as the data is duplicated 4 times.

### 3.1 Research Questions

To guide our analysis, we formulated the following research questions, based on the first intervention (rephrasing):

1. **RQ1:** How does the difficulty of rephrased templates affect the performance of Llama-2 in the presence of potential data contamination?

2. **RQ2:** Does the performance of Llama-2 on contaminated data differ when fine-tuned to templates of different difficulty?

3. **RQ3:** What is the effect of varying the difficulty of the templates used for fine-tuning on the level of data contamination and the subsequent performance evaluation of Llama-2?

All these questions are analysed in the context of the exemplar difficulty as well (# digits), as this is the second intervention that can affect performance, and one intervention can mask the other:

## 4 Results

As shown in previous studies, LLMs are sensitive to prompts, i.e., the way that the request is formulated. Figure 1 shows that even for a simple task such as addition, rephrasing the question influences model performance. We can observe that template 2 in average is the easiest and template 7 is the most difficult version of rephrasing addition among our 7 templates. Consequently, as rephrasing a test item can change its difficulty level, this should be considered when this approach –rephrasing test items– is taken to address contamination. A lower performance of models on the rephrased test items might be simply due to the higher difficulty level of them and may not be a sign of their purity.

Figure 3 demonstrate this effect more clearly. As it can be seen, there are cases that the performance of the model for one or more templates when tested on the non-contaminated validation

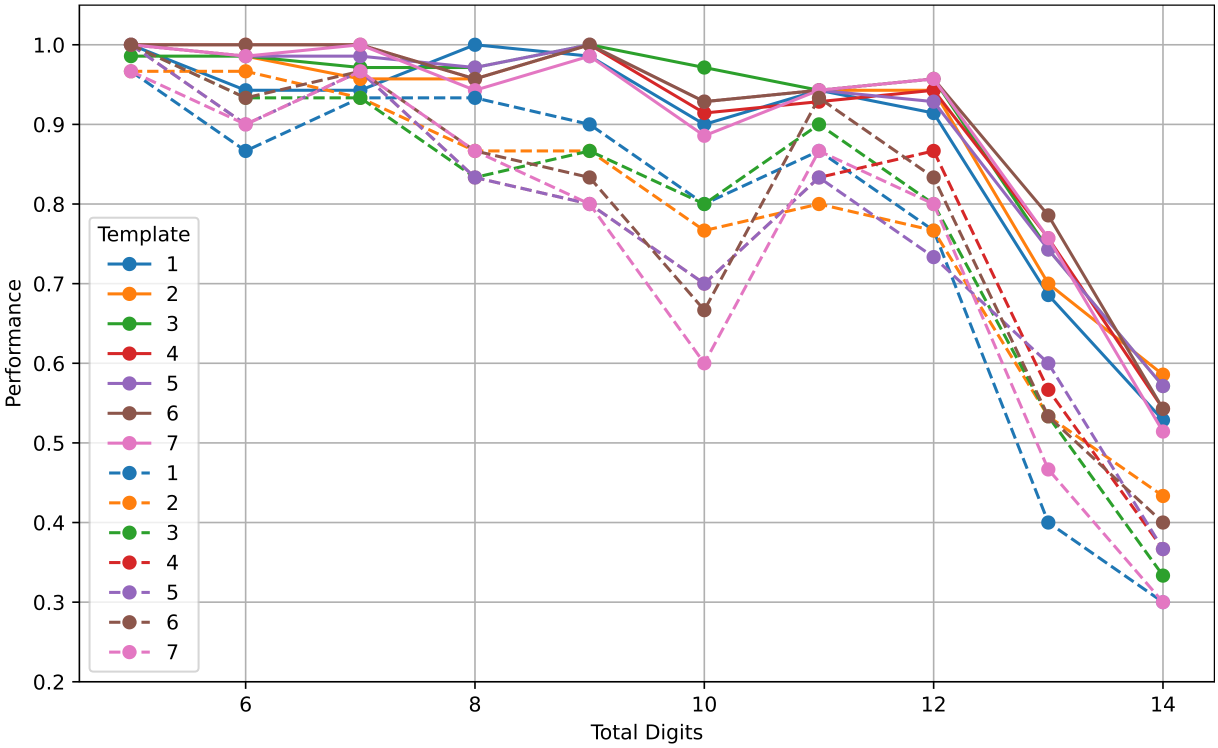

Figure 3: Performance of Llama2 7B chat model fine-tuned on the data described on Figure 2. The dashed lines show the performance of fine-tuned model on the contaminated test set. The test set is contaminated because it contains the exemplar (addition pairs) appeared in the training with the original template and the other 6 templates. The solid lines show the performance of the fine-tune model on the validation set where no exemplar (addition pair) from it exists in the training set.

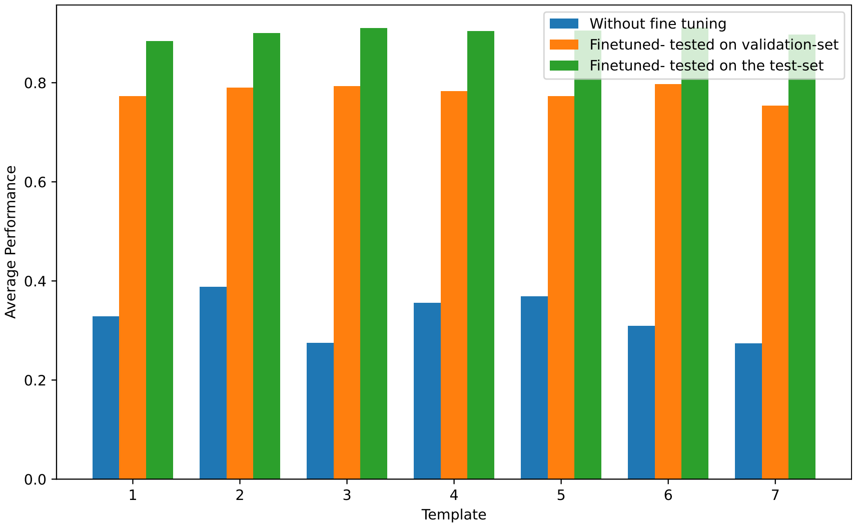

Figure 4: Llama2 7B model average performance comparison when tested on the non-contaminated validation-set, the contaminated test-set, train-set, and test-set excluding training instances

set is higher than some templates when tested on the contaminated test-set. This phenomenon can be seen for 5-digit, 6-digit, 7-digit, and 11-digit additions (Figure 3 clearly demonstrates this when the solid lines cross the dashed lines).

### 4.1 Difficulty of the Contaminating Template

Rephrased items that can potentially be found in the training data might have a different difficulty level compared to the test data points and can influence the contaminating level in different ways. To analyze this, we fine-tuned Llama2 7B chat model

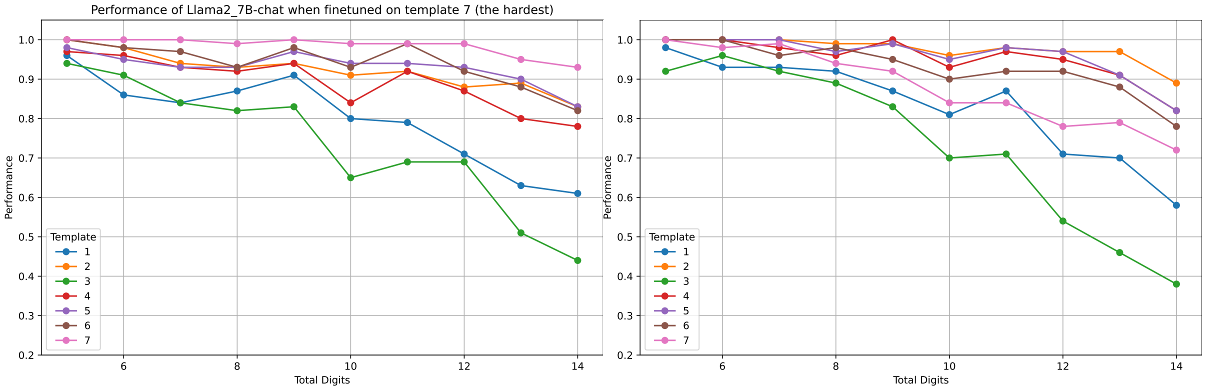

Figure 5: Performance of Llama2 7B chat model when fine-tuned. Left: fine-tuned on the most difficult template #7 (based on the model performance before fine-tuning). Right: fine-tuned on the easiest template #2.

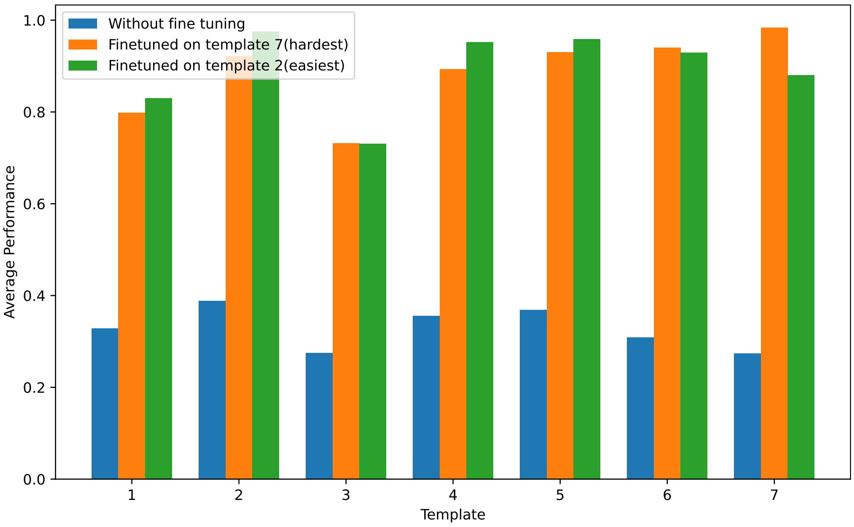

Figure 6: Average performance of Llama2 7B models before and after fine-tuning on templates #2 and #7.

on the most difficult and on the easiest template and measure their performance of the fine-tuned models on all templates variations. As can be seen in Figure 1, for the Llama2 7B chat model, template 2 is the easiest, as on average the performance for this template is higher than those of the others. Template 7 on the other hand, is the most difficult for this model (before fine-tuning).

Figure 5 shows the performance of the fine-tuned models. It is noticeable that performance of the fine-tuned model on templates that have not been used in fine-tuning is affected differently when fine-tuned on templates 2 or 7. Humans are not highly sensitive to the way a question is formulated and in those cases that they are, it is expected that if they learn the hardest rephrased form of a task, they can easily cover the easier variations too. Figure 6 shows that this does not hold for LLMs. Fine-tuning on the easiest template shows better performance on 3 other templates (templates 1, 3, 4 and 5) however, for the case which we fine-tuning on the hardest template the resulting model performs better only on one other template (number 6).

### 4.2 Diversity of the Contaminating Template

A training set that contains one rephrased item from the test set that is repeated $k$ times has been contaminated in a different way compared to a training set that has $k$ variations of a test item. In the latter case, the diversity of the contaminating template is higher. But does higher diversity cause higher contamination effect in term of performance boost? In order to find out the response to this question, we fine-tuned Llama2 7B model in two different scenarios. In the first scenario, we fine-tune the

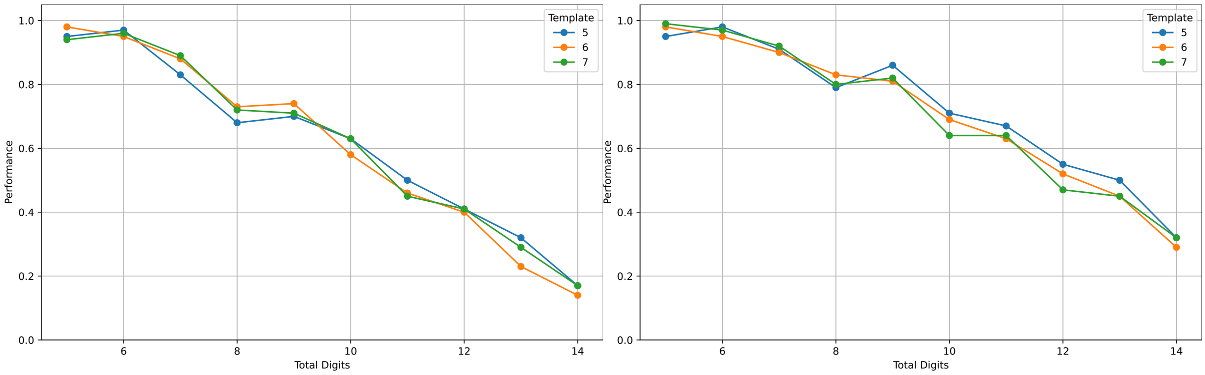

Figure 7: On the left the performance of Llama2 7B chat when fine-tuned one epoch on templates 1, 2, 3 and 4. On the right, performance of the model when fine tuned on a random template between 1, 2, 3 or 4 for four epochs.

model on all instances of templates 1, 2, 3 and 4 and then tested on the remaining templates (5, 6 and 7). The result can be seen in Figure 7 left. In the second scenario, while we use the same test set, the training set contains one random template between 1, 2, 3 and 4 that is repeated 4 times (Figure 7 right). A close look at these figures reveals that if we fix the number of times a rephrased form of a test instance is found in the training set, less diversity has a higher contaminating effect. In our experiments the average performance of the model fine.tuned on diverse templates is 61.4 while we get 71.0 for the less diverse counterpart. This could be due to the fact that in the diverse cases, the model's weights are updated to learn the template and the task which in our case id the addition, but in a less diverse scenario, the model can concentrate on the task as the template wrapping the main task is not changing.

## 5 Conclusions

Our study explored the effects of data contamination and instance variation on the performance of large language models (LLMs), specifically using fine-tuned LLAMA-2 models on an addition task. Key findings from this investigation answered our research questions and highlighted important insights.

Regarding template difficulty (**RQ1**), we show that the performance of LLMs is significantly influenced by the template or rephrasing of test items. Easier templates, such as Template 2, consistently yielded higher performance, while more complex templates, like Template 7, posed greater challenges. When analysing the impact of fine-tuning on contaminated data (**RQ2**), we observed that performance on non-contaminated templates varies after fine-tuning. Contrary to expectations, some templates showed higher performance on non-contaminated validation sets compared to contaminated test sets, indicating that contamination and difficulty levels are deeply intertwined. Finally, focusing on template diversity and contamination (**RQ3**), we saw that fine-tuning on an easier templates improved performance on other templates more consistently than fine-tuning on harder templates. This challenges the assumption that learning from more difficult examples would generalize better to simpler variations. Additionally, lower template diversity in the training set amplified the contamination effect, suggesting that less diverse contamination scenarios have a stronger influence on model performance.

These findings highlight the relevance of considering both the difficulty and diversity of rephrased instances when evaluating LLM performance. In particular, the change of rephrasing and exemplars can have confounding effects masking contamination or suggesting contamination where there is not. Our results suggest that addressing data contamination effectively requires more nuanced strategies that accommodate these factors. For future work, we are investigating the effects of contamination and instance variation across more complex and diverse NLP tasks.

## 6 Acknowledgments

We thank the anonymous reviewers for their comments.

This work was funded by ValGRAI, the Norwegian Research Council grant 329745 Machine Teaching for Explainable AI, CIPROM/2022/6 (FASSLOW) and IDIFEDER/2021/05 (CLUSTERIA) funded by Generalitat Valenciana, the EC H2020-EU grant agreement No. 952215 (TAILOR), US DARPA HR00112120007 (RECoG-AI) and Spanish grant PID2021-122830OB-C42 (SFERA) funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe".

In compliance with the recommendations of the Science paper about reporting of evaluation results in AI (Burnell et al., 2023), we include all the results at the instance level (`https://github.com/Behzadmeh/ACL-CONDA-2024`).

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D Ullman, Fernando Martinez-Plumed, Joshua B Tenenbaum, Danaja Rutar, Lucy G Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, Douwe Kiela, Murray Shanahan, Ellen M. Voorhees, Anthony G. Cohn, Joel Z Leibo, and Jose Hernandez-Orallo. 2023. Rethink reporting of evaluation results in AI. *Science*, 380(6641):136–138.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Yucheng Li. 2023. An open source data contamination report for llama series models. *arXiv preprint arXiv:2310.17589*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Federico Ranaldi, Elena Sofia Ruzzetti, Dario Onorati, Leonardo Ranaldi, Cristina Giannone, Andrea Favalli, Raniero Romagnoli, and Fabio Massimo Zanzotto. 2024. Investigating the impact of data contamination of large language models in text-to-sql translation. *arXiv preprint arXiv:2402.08100*.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq R. Joty. 2024. How much are llms contaminated? a comprehensive survey and the llmsanitize library. *ArXiv*, abs/2404.00699.

Martin Riddell, Ansong Ni, and Arman Cohan. 2024. Quantifying contamination in evaluating code generation capabilities of language models. *arXiv preprint arXiv:2403.04811*.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

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*.

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*.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. Kieval: A knowledge-grounded interactive evaluation framework for large language models. *arXiv preprint arXiv:2402.15043*.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yunze Song, Yumeng Zhang, Hanxu Hu, Yiran Wei, Rui Wang, and Hongyuan Lu. 2023. Clean-eval: Clean evaluation on contaminated large language models. *arXiv preprint arXiv:2311.09154*.