Data Contamination Can Cross Language Barriers

Anonymous ACL submission

Abstract

The opacity in developing large language models (LLMs) is raising growing concerns about the potential contamination of public benchmarks in the pre-training data. Existing contamination detection methods are typically based on the text overlap between training and evaluation data, which can be too superficial to reflect deeper forms of contamination. In this paper, we first present a cross-lingual form of contamination that inflates LLMs' performance while evading current detection methods, deliberately injected by overfitting LLMs on the translated versions of benchmark test sets. Then, we propose generalization-based approaches to unmask such deeply concealed contamination. Specifically, we examine the LLM's performance change after modifying the original benchmark by replacing the false answer choices with correct ones from other questions. Contaminated models can hardly generalize to such easier situations, where the false choices can be *not even wrong*, as all choices are correct in their memorization. Experimental results demonstrate that cross-lingual contamination can easily fool existing detection methods, but not ours. In addition, we discuss the potential utilization of cross-lingual contamination in interpreting LLMs' working mechanisms and in post-training LLMs for enhanced multilingual capabilities.

1 Introduction

The pre-training data of current large language models (LLMs) tends to be undisclosed by default, even for those open-sourced models (Meta, 2024; Jiang et al., 2024a). As the scores on popular benchmarks continuously reach new heights, their performance in solving real-world tasks seems inconsistent with the leaderboard (Beeching et al., 2023). Such intransparency in training and inconsistency in user experience has drawn increasing attention to the underlying contamination of public benchmarks in the pre-training data, indicating that some LLMs may simply memorize the answers to difficult questions without a true understanding.

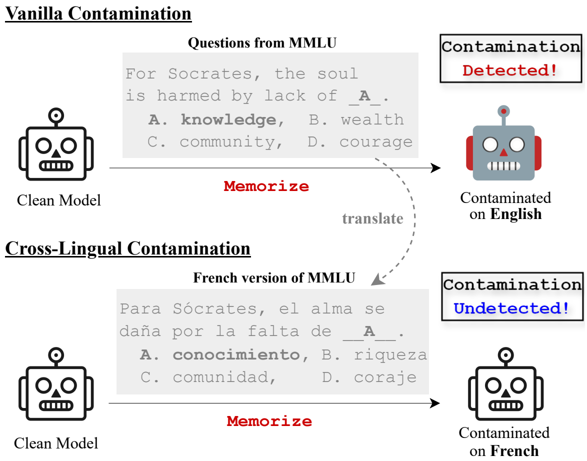

Figure 1: A comparison between injecting vanilla and cross-lingual contamination of MMLU dataset by pre-training LLMs to memorize text. Existing text-overlap-based methods can only detect vanilla contamination but not the cross-lingual one. Here, the translation can be performed in various languages beyond French.

Existing studies often define and detect contamination based on the text overlap or n-gram duplication between pre-training and evaluation data (Chowdhery et al., 2023; Touvron et al., 2023; Jiang et al., 2024b), which only focus on the surface form of the text data without considering the deeper knowledge or semantics in the contamination. We argue that the essence of contamination is not superficial text memorization but the non-generalizable memorization of knowledge or capabilities.

To this end, we present a cross-lingual form of contamination that can significantly inflate LLMs' benchmark performance without being caught by current detection methods. Cross-lingual means the models are contaminated on other languages and then evaluated on English test sets. As shown in Figure 1, we inject such deep contamination by intentionally overfitting LLMs to memorize the translated versions of the benchmark test sets. Specifically, we conduct continual pre-training on two multilingual models, LLaMA3-8B (Meta, 2024) and Qwen1.5-7b (Bai et al., 2023), using translated versions of three popular benchmarks—MMLU (Hendrycks et al., 2020), ARC Challenge (Clark et al., 2018), and MathQA (Amini et al., 2019)—in seven different languages. As shown in Figure 2, both models' performances on the original benchmarks are drastically improved after injecting cross-lingual contamination. Meanwhile, we employ state-of-the-art detection methods based on model completion (Oren et al., 2023; Xu et al., 2024) and LLM judgment (Golchin and Surdeanu, 2023) to test them for contamination. Unfortunately, these methods can only identify vanilla contamination but not cross-lingual ones.

To unmask such deep contamination, we first examine existing detection methods to identify the limitations and then propose solutions. Current methods are predominantly based on text overlap, either checking for string matching between pre-training and evaluation data (Deng et al., 2023; Li, 2023b; OpenAI, 2023; Touvron et al., 2023; Riddell et al., 2024), or comparing the models' output text or likelihood with the evaluation data given controlled prompts (Oren et al., 2023; Xu et al., 2024). The key idea of such methods is to verify if the model has seen or memorized a specific surface form of text, which we believe is too superficial to reflect the essence of contamination.

Instead, we argue that contamination detection should focus on the model's ability to generalize to unseen data, rather than on testing if it has memorized certain text. For instance, in the cross-lingual scenario, the model did not memorize the specific English form of the benchmarks, but can still obtain non-generalizable memorization of corresponding knowledge from contamination in other languages. In this case, if we still scrutinize for any memorization of the English benchmarks, the detection results will be unreliable. Therefore, we propose generalization-based detection approaches that examine the model's performance change on a generalized version of the original benchmark, created by modifying the questions and answer choices. Specifically, for each question, we replace all the incorrect choices with correct choices taken from other questions. Through this manipulation, models that really understand the question should achieve better performance, as some choices can be

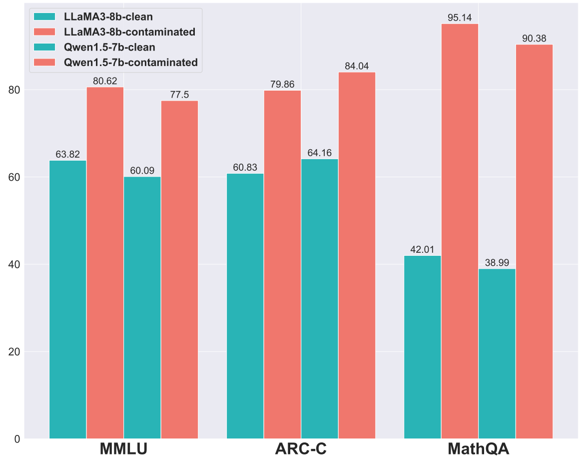

Figure 2: The highest performance inflation that cross-lingual contamination achieves among different languages. Results for all languages are shown in § 3.2

not even wrong to the question, while the contaminated ones can get confused as all choices are memorized as correct. Extensive experimental results prove the effectiveness of our proposed method in detecting cross-lingual contamination.

Additionally, we are curious about why cross-lingual contamination can inflate LLMs' performance and how we can utilize it beyond cheating in evaluation. Hence, we discuss its connections with the interpretability of LLMs and post-training for enhancing LLMs' multilingual capabilities.

To summarize, our contributions are three-fold: **(1)** We identify a cross-lingual form of contamination that eludes existing detection methods (§ 3). **(2)** We propose generalization-based detection methods to unmask such deep contamination (§ 4). **(3)** We discuss the potential impact of cross-lingual contamination on interpreting the working mechanisms of LLMs and on improving their multilingual capabilities via post-training (§ 5). The code, dataset, and checkpoints we use will be publicly released to facilitate related research.

2 Preliminary

In this section, we introduce the definition of contamination and basics for corresponding detection methods (§ 2.1), and our investigation setup (§ 2.2).

2.1 Contamination Definition

While the concept of contamination has been brought up in numerous studies, there is no universally acknowledged strict definition for it.

According to the essence of the concept, we first summarize the most commonly adopted definitions in existing works as memorization-based and highlight their limitations. Then, we propose a generalization-based definition, which forms the basis for our proposed detection methods.

Memorization-Based Most prior studies define contamination based on n-gram duplication between pre-training and evaluation data (Jiang et al., 2024b), which can be summarized as instances where the model has memorized specific pieces of text. Bear this intuition in mind, we can easily understand the essence of existing detection methods and categorize them into two types: **(1) When pre-training data is accessible**, they directly adopt n-gram or text similarity matching between pre-training and evaluation data to examine the duplication that can cause memorization (Radford et al., 2019; Brown et al., 2020; Dodge et al., 2021; Chowdhery et al., 2023; OpenAI, 2023; Touvron et al., 2023; Li, 2023b; Deng et al., 2023; Lee et al., 2023; Gunasekar et al., 2023; Riddell et al., 2024). **(2) When pre-training data is inaccessible**, they prompt the models using a subset of the evaluation data and analyze if the output is a reproduction of specific pieces of text or assess their likelihood, to indirectly determine if certain text memorization exists (Oren et al., 2023; Golchin and Surdeanu, 2023; Li, 2023a; Nasr et al., 2023; Shi et al., 2023; Dong et al., 2024; Xu et al., 2024).

Generalization-Based We suggest that simply testing text memorization can be inadequate to reveal deeper contamination (like the cross-lingual one we identify), where the model is contaminated without memorizing the specific surface form of the text. Therefore, we tend to define contamination as instances where a model acquires non-generalizable knowledge of the evaluation data through various means, such as memorizing the original or transformed (e.g., translated, paraphrased, summarized) forms of the benchmarks.

2.2 Investigation Setup

The primary goals of our investigation are to: **(1)** Verify the feasibility of deep forms of contamination (§ 3). **(2)** Determine whether existing methods can detech such contamination (§ 4.1). **(3)** Propose detection methods capable of identifying such deeply concealed contamination (§ 4.2).

Considering it is unclear whether existing LLMs contain cross-lingual contamination, we intentionally inject such contamination to open-sourced models to obtain contaminated models. Then, we detect such contamination using existing methods and our proposed methods. The detailed investigation configurations are as follows.

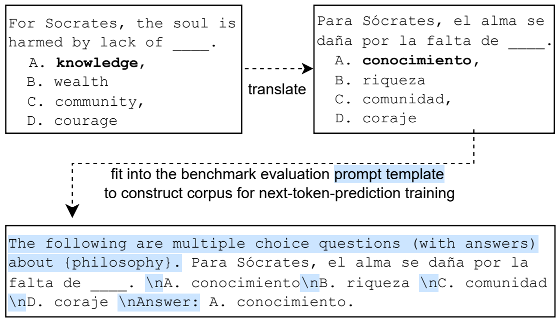

Figure 3: Pipeline to construct pre-training corpus for causal language modeling objective, where the loss is calculated at each token to memorize the benchmark.

Models. To inject cross-lingual contamination, the backbone model should be able to understand different languages. Hence, we employ two multilingual LLMs, LLaMA3-8B (Meta, 2024) and Qwen1.5-7B (Bai et al., 2023), as the backbones.

Datasets. To exhibit the impact of such contamination in evaluation, we adopt three popular benchmarks to inject contamination, MMLU (Hendrycks et al., 2020), ARC Challenge (Clark et al., 2018), and MathQA (Amini et al., 2019), where modern LLMs typically compete with each other.

Languages. For cross-lingual contamination, we utilize seven non-English languages that are commonly supported: Chinese, French, German, Italian, Japanese, Korean, and Spanish.

3 Injecting Cross-Lingual Contamination

In this section, we present the injection process of cross-lingual contamination (§ 3.1) and the inflated performance of the contaminated models (§ 3.2).

3.1 Cross-Lingual Contamination

To acquire knowledge from contamination of the evaluation data, we overfit open-sourced LLMs on the translated versions of the benchmark test sets, instead of directly memorizing the original form of text. The process of constructing the training data for contamination is illustrated in Figure 3.

We first translate the benchmark test sets into non-English languages mentioned in § 2.2. Considering the cost and quality balance, we utilize LLaMA3-8B to conduct the translation. The specific prompt template is shown in appendix A.2.

Backbone	Dataset	Clean Model	Vanilla Contaminated	Cross-Lingual Contaminated						
				Chinese	French	German	Italian	Japanese	Korean	Spanish
LLaMA3-8B	MMLU	63.82	98.01	71.12	79.16	65.26	79.89	66.15	68.11	**80.62**
	ARC-C	60.83	91.63	56.22	74.91	61.17	**79.86**	66.29	46.24	73.29
	MathQA	42.01	97.78	86.56	**95.14**	88.17	93.06	84.08	81.71	93.96
Qwen1.5-7B	MMLU	60.09	97.89	67.91	76.13	73.2	75.02	62.34	61.99	**77.5**
	ARC-C	64.16	97.01	**84.04**	69.36	61.17	61.77	62.54	52.55	63.73
	MathQA	38.99	95.61	79.76	**90.38**	89.21	88.1	77.01	77.21	89.48

Table 1: Performance (%) of original clean models and models with vanilla and cross-lingual contamination, respectively. Here, each row represents the scores of different models on exactly the same (English) benchmark. 'Vanilla' indicates the model is contaminated directly on the English version of the benchmark, and the 'Cross-Lingual Contaminated' columns show the scores of models contaminated in a specific non-English language.

Then, we customize the questions and choices to fit in the corresponding prompt templates used for the evaluation of specific benchmarks. In this way, we construct the corpus for continual pre-training of the backbone models through the causal language modeling objective, which stimulates the real-world scenario where specific data contamination is blended into the training corpus. The vanilla contamination is injected in the same way using the original English benchmarks. The training hyperparameters are provided in Table 5.

We inject the contamination for different benchmarks separately, ensuring that each model only contains contamination of one specific benchmark in a single language. Mixing different benchmarks and languages is another way to inject cross-lingual contamination, which we leave for future work.

3.2 Evaluating Contaminated Models

While the contamination is injected in non-English languages, we evaluate these contaminated models on the original English benchmarks to assess their potential impact on misleading the leaderboard.

We report zero-shot accuracy for three types of models: (1) **Clean:** The original backbones with no added contamination. (2) **Vanilla Contaminated:** Backbones contaminated by the original English benchmarks. (3) **Cross-Lingual Contaminated:** Backbones contaminated by non-English translated benchmarks. The evaluation is implemented through LM-Eval framework (Gao et al., 2023) and the results are exhibited in Table 1.

For models with vanilla contamination, their accuracy is close to 100%. This is expected since the models are directly overfitted on these test sets. In the cross-lingual contamination scenario, models are not directly trained on the benchmarks. Surprisingly, the cross-lingual contamination can sneak beyond language barriers and carry over to English.

Regarding models with cross-lingual contamination, their performance, while not reaching 100%, exhibits significant inflation, even though the translation provided by LLaMA3-8B is imperfect. We observe a consistent 5%-10% improvement on the MMLU benchmark across languages, with an even stronger enhancement seen on the MathQA benchmark. The instability of the performance gains shown on ARC-C can be caused by the low-quality translation of the dataset. In addition, we hypothesize that models can more easily memorize factual knowledge (MMLU) and Arabic numbers' operations (MathQA) than reasoning in languages (ARC-C), which is intuitive. One may understand the intricacies of arithmetic or fact retention through repetitive exposure and practice, but reasoning in natural languages, as required in ARC-C tasks, involves a more complex interplay of context, inference, and flexible application of knowledge.

Another interesting finding is the effect of cross-lingual contamination's language category on the contamination effect. We observe that European languages (French, German, Italian, and Spanish) can provide stronger cross-lingual contamination onto English, while Asian languages (Chinese, Japanese, and Korean) provide a lesser effect. This phenomenon could be explained by the closer subword vocabulary shared among these languages, or it might be considered as reflecting a more similar conceptual space among European languages. Since the focus of our paper is to study and prevent contamination in LLM training, we will leave exploration on this end as future work.

4 Detecting Cross-Lingual Contamination

In this section, we conduct detection on the cross-lingual contamination utilizing conventional memorization-based methods (§ 4.1) and our proposed generalization-based approaches (§ 4.2).

Backbone	Dataset	Clean Model	Vanilla Contaminated	Cross-Lingual Contaminated Chinese	French	German	Italian	Japanese	Korean	Spanish
		Shared Likelihood (Metric: p-value)								
LLaMA3-8B	MMLU	0.3281	0.3421	0.6827	0.1295	**0.0031**	0.2935	0.5857	0.9351	0.8231
	ARC-C	0.6125	0.6065	0.7327	0.4442	0.3156	0.6110	0.7734	0.6730	0.3446
	MathQA	0.4876	**0.0000001994**	0.4348	0.3102	0.4573	0.1548	0.1983	0.5789	0.6037
Qwen1.5-7B	MMLU	0.7031	0.5866	0.5039	0.2404	0.8566	0.1708	0.3658	0.5688	0.4981
	ARC-C	0.1006	0.1355	0.3740	0.2562	0.3608	0.1302	0.1698	0.4575	0.3258
	MathQA	0.4495	**0.0000006167**	0.2011	0.2934	0.5145	0.4994	0.1355	0.5064	0.5429
		Guided Prompting (Metric: Accuracy (%))								
LLaMA3-8B	MMLU	8.20	4.80	0.80	1.00	5.10	4.70	2.00	1.20	1.40
	ARC-C	1.62	2.39	0.09	1.54	1.28	1.79	0.34	2.13	0.77
	MathQA	0.20	0.13	0.30	0.10	0.23	0.13	0.07	0.10	0.03
Qwen1.5-7B	MMLU	1.30	5.60	0.30	0.60	0.80	1.2	0.4	0.5	0.2
	ARC-C	2.39	0.60	0.00	0.17	0.34	0.09	0.25	0.34	0.26
	MathQA	0.07	0.10	0.03	0.00	0.13	0.10	0.00	0.07	0.03
		N-Gram Accuracy (Metric: Accuracy (%))								
LLaMA3-8B	MMLU	10.02	**73.34**	2.42	2.38	2.32	2.41	3.62	4.83	2.41
	ARC-C	4.91	**70.66**	3.52	3.04	4.32	3.45	3.55	5.32	2.94
	MathQA	8.40	**45.11**	5.15	7.90	8.09	6.89	6.43	5.29	6.85
Qwen1.5-7B	MMLU	8.78	**70.56**	3.27	2.61	2.88	2.51	4.22	5.35	2.56
	ARC-C	22.25	**33.33**	0.36	0.20	0.29	0.22	1.08	0.63	0.19
	MathQA	20.98	**44.31**	8.21	7.05	7.33	8.21	11.96	11.97	8.03

Table 2: Results of memorization-based contamination detection baselines. Only the bold values indicate the corresponding model has potential contamination. **(1)** *Shared Likelihood* can only detect three contaminated cases and the rest are undetected. **(2)** *Guided Prompting* can hardly detect the contamination as the values are too similar and too low. **(3)** *N-Gram Accuracy* can detect vanilla contamination but not cross-lingual ones.

4.1 Memorization-Based

For memorization-based methods defined in § 2.1, we select three typical ones and their detection results are shown in Table 2. We briefly introduce these methods and discuss their results below.

4.1.1 Shared Likelihood

Oren et al. (2023) propose to identify the test set memorization through prompting and statistically analyzing the difference between log probabilities on the original dataset and its shuffled version.

This bias is quantitatively assessed through a permutation test, where the log probabilities assigned by the model to the canonical order are compared against those for various random permutations of the dataset. A significantly higher likelihood for the canonical order compared to the permuted ones implies the model has memorized the original data. The result is delivered by the p-value of the permutation test. A p-value that is smaller than 0.05 suggests a high likelihood of contamination.

We follow the implementation provided by Oren et al. (2023). As shown in Table 2, only the vanilla-contaminated models on MathQA and German-contaminated LLaMA on MMLU are detected. The rest of the contaminated models did not exhibit the expected low p-values. Such discrepancies indicate the limitations of this method in our setting.

4.1.2 Guided Prompting

Golchin and Surdeanu (2023) employ meticulously crafted prompts to guide the model in generating specific text and ask an LLM to judge its similarity to the evaluation data, thereby confirming whether the model has memorized certain pieces of text.

Specifically, one of the four candidate choices is masked and the model is prompted with detailed information to predict it by generation. Then, GPT-3.5/4 is employed to judge if the predicted choice essentially has the same meaning as the original one or not. If a model can correctly predict the masked choice, it indicates the model has memorized the questions with the choices, proving the potential contamination encoded during training.

We utilize GPT-4o (OpenAI, 2024) to judge if the predicted choice is correct and the corresponding prompt is provided in appendix B.2. Based on the prediction accuracy shown in Table 2, it is difficult to determine which model is contaminated, as most values are too low and too similar to tell them apart. Therefore, guided prompting also fails to detect the contamination in our setting.

Backbone	Dataset	Clean Model	Vanilla Contaminated	Cross-Lingual Contaminated						
				Chinese	French	German	Italian	Japanese	Korean	Spanish
LLaMA3-8B	MMLU	63.82	98.01	71.12	79.16	65.26	79.89	66.15	68.11	80.62
	MMLU-g	90.07	81.01	52.71	36.45	29.50	70.82	42.69	47.09	62.78
	difference	**+26.25**	-17.00	-18.41	-42.71	-35.76	-9.07	-23.46	-21.02	-17.84
	ARC-C	60.83	91.63	56.22	74.91	61.17	79.86	66.29	46.24	73.29
	ARC-C-g	73.55	31.74	26.37	40.27	75.00	26.37	26.71	26.79	60.75
	difference	**+12.72**	-59.89	-29.85	-34.64	+13.83	-53.49	-39.58	-19.45	-12.54
	MathQA	42.01	97.78	86.56	95.14	88.17	93.06	84.08	81.71	93.96
	MathQA-g	55.57	98.12	90.81	96.11	90.91	94.40	88.60	87.63	95.54
	difference	**+13.56**	+0.34	+4.25	+0.97	+2.74	+1.34	+4.52	+5.92	+1.58
Qwen1.5-7B	MMLU	60.09	97.89	67.91	76.13	73.20	75.02	62.34	61.99	77.50
	MMLU-g	77.58	80.62	69.51	68.65	68.06	70.05	66.69	63.32	72.88
	difference	**+17.49**	-17.27	1.60	-7.48	-5.14	-4.97	4.35	1.33	-4.62
	ARC-C	64.16	97.01	84.04	69.36	61.17	61.77	62.54	52.55	63.73
	ARC-C-g	85.92	29.61	34.56	26.62	29.18	26.88	24.91	26.45	26.71
	difference	**+21.76**	-67.40	-49.48	-42.74	-31.99	-34.89	-37.63	-26.10	-37.02
	MathQA	38.99	95.61	79.76	90.38	89.21	88.10	77.01	77.21	89.48
	MathQA-g	44.67	95.44	83.37	89.44	89.44	88.67	81.62	80.75	89.37
	difference	**+5.68**	-0.17	+3.61	-0.94	+0.23	+0.57	+4.61	+3.54	-0.11

Table 3: Generalization-based contamination detection results. Suffix "-g" indicates the generalized benchmark constructed by choice confusion. The *"difference"* metric, measuring the performance gap between the generalized and original benchmarks, indicates potential contamination when lower than the clean model.

4.1.3 N-Gram Accuracy

Similar to masking out the choice, Xu et al. (2024) examine the model's memorization by removing the entire answer part of the generation benchmarks and verifying if the model's generated output matches the removed answer text.

Since the benchmarks we adopt in this paper are all multiple-choice typed, we combine all choices to form the "answer" and check if the model will automatically generate the choices given a normal question from the benchmark. Then, we use this constructed "answer" to calculate the N-gram accuracy as defined in (Xu et al., 2024). The key idea is still to verify if the model has memorized the text. More details are provided in appendix B.3.

From the results shown in Table 2, we observe that the accuracy of models injected with vanilla contamination is much higher than the corresponding clean model, suggesting the presence of contamination. Meanwhile, models with cross-lingual contamination present consistently lower n-gram accuracy than the clean model, indicating that such contamination cannot be detected by this method.

4.2 Generalization-Based

As there can be countless transformations of the evaluation data, detecting duplication of a specific surface form becomes unfeasible. Based on our definition in § 2.1, we propose generalization-based methods that detect contamination by evaluating the models' ability to generalize to unseen data.

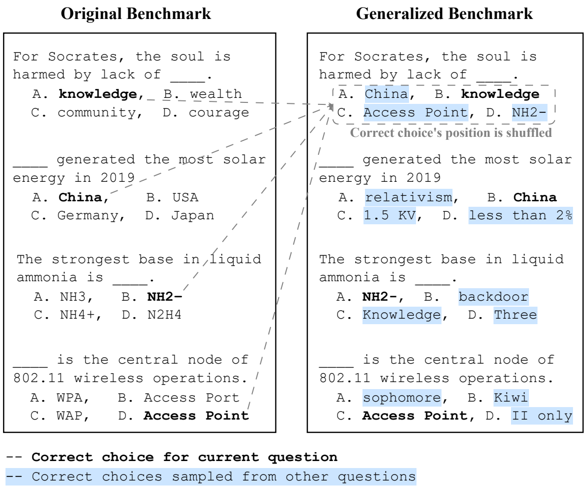

Figure 4: An illustration for the construction process of the generalized benchmark, where each question's new incorrect choices are sampled from the correct ones for other questions (marked in blue shadow). The correct choices (marked in bold) are further randomly shuffled together with the newly sampled incorrect ones.

4.2.1 Constructing Generalized Benchmark

The key idea of our proposed method is to test whether a model achieving high performance on a specific benchmark can further excel when faced with an easier variant of the same benchmark.

As illustrated in Figure 4, we replace the false choices of the current question with correct ones from other questions to create the generalized version of the benchmark. In addition, we shuffle the choices to ensure the model cannot simply predict the correct answer via the answer order shortcut.

In this case, the newly sampled false choices can be *not even wrong* to the current question, making it much easier to answer and thereby yield a significant performance gain for models that genuinely understand the question. However, if a model is contaminated, it may get confused as the newly sampled false choices are still "correct" according to its memorization during pre-training. This confusion can lead to little performance gain or even a drop in performance. Therefore, we refer to our proposed method as **choice confusion**.

4.2.2 Measuring Contamination

We calculate the difference in the same model's performance between the generalized and original versions of the benchmark and use it as the metric to assess the potential contamination.

As shown in Table 3, all clean models show remarkable improvements. While models with either vanilla or cross-lingual contamination exhibit minimal improvement compared with that of the clean model, or a significant decline in performance in most cases, indicating contamination detected.

We observe that the metric relates to datasets. For MMLU and ARC-C, contaminated models tend to experience a performance drop. However, for MathQA, most of them exhibit a slight increase. We assume this is because most of the choices are Arabic numbers, making it difficult for the model to memorize all the correct answers without the question, and therefore it becomes less confusing.

4.2.3 Evaluating Real-World LLMs

Existing memorization-based methods can only detect limited types of contamination, as they assume the model memorizes text in specific forms.

Though inspired by cross-lingual contamination, our proposed generalization-based detection method is not limited to this specific form and can be applied to any scenario where the model is injected with non-generalizable knowledge.

We employ our proposed method to detect potential contamination in several trending LLMs in the real world. The results in Table 4 indicate that Phi2 can be inadvertently contaminated on MMLU and ARC-C benchmarks. Similarly, the math expert LLM Abel-7B may unintentionally acquire contamination from the MathQA benchmark data. Model details are provided in appendix B.4

	llama2 7b	mistral 7b	phi2 2.7b	phi3 3.8b	abel 7b	glm4 9b	qwen2 7b
MMLU	44.88	57.29	23.83	67.27	47.08	67.36	69.05
-g	72.87	82.71	25.02	85.29	68.37	84.91	89.23
diff	+27.99	+25.42	**+1.20**	+18.02	+21.29	+17.55	+20.18
ARC-C	36.18	64.08	42.92	80.20	50.34	86.35	84.81
-g	44.71	85.75	47.27	92.15	66.04	91.81	95.22
diff	+8.53	+21.67	**+4.35**	+11.95	+15.70	+5.46	+10.41
MathQA	28.71	36.88	31.32	41.14	34.30	43.05	44.36
-g	36.18	45.77	38.70	49.06	35.71	56.04	49.03
diff	+7.47	+8.89	+7.38	+7.92	**+1.41**	+12.99	+4.67

Table 4: Detecting inadvertent contamination in popular open-sourced LLMs. Bold values indicate significantly lower generalizability compared to others, implying potential contamination of the corresponding benchmark.

5 Beyond Contamination

Can cross-lingual contamination only be utilized for cheating on benchmarks? In this section, we further discuss two potential scenarios where cross-lingual contamination can serve as a good starting point: interpreting the working mechanisms of LLMs (§ 5.1) and improving LLMs' unbalanced multilingual capabilities (§ 5.2).

5.1 How Do LLMs Think Across Languages?

From Table 1, we observe that the performance of the **same** backbone model can vary significantly when continually pre-trained on the **same** benchmark data in different languages. This is intriguing as we are injecting the **same** amount of knowledge.

Our hypothesis is that the knowledge in a model can be fixed, and language acts as an interface. Due to the uneven distribution of languages in the training corpus, the model's ability to understand and generate text can vary across different languages, which can be regarded as interfaces with varying qualities. In this case, despite the model having the same underlying knowledge, its performance can vary significantly, depending on the quality of the interfaces through which it is adopted.

Wendler et al. (2024) propose a similar idea that LLMs operate in "input", "concept", and "output" spaces when processing non-English. The input and output spaces here are similar to the language interfaces in our assumption. Huang et al. (2024) enhance LLMs' multilingual ability by feeding LLMs the encoded representation instead of the text of non-English inputs, which is also consistent with our hypothesis of language interfaces.

Therefore, we believe cross-lingual contamination can be a promising starting point for exploring the interpretability of multilingual LLMs.

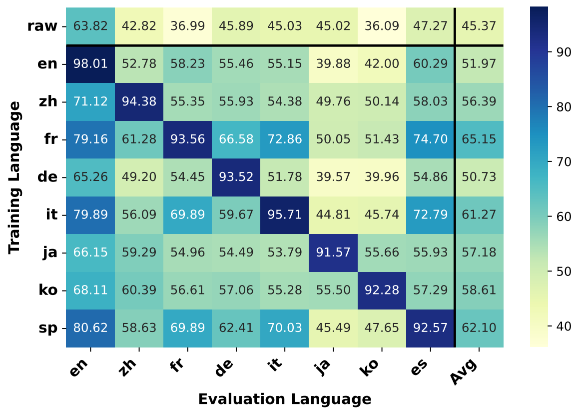

Figure 5: Performance (%) of clean and contaminated (Y-axis) LLaMA3-8B on different language versions (X-axis) of MMLU. Here, the first row "**raw**" represents the clean model's performance. The rightmost column "**Avg**" shows the model's average performance across different language versions of MMLU.

5.2 How to Localize LLMs for Non-English?

Considering a scenario where the budget is limited and we want a model with the best overall multilingual performance, in which single language should we conduct the continual pre-training?

As noted in § 3.2, contamination in non-English languages can improve performance on the English benchmark. We further extend the evaluation to non-English languages to assess the impact of contamination on multilingual performance.

Figure 5 shows that contaminating in French achieves the best average performance, indicating that French could be the best choice for continual pre-training. Surprisingly, English only scored 51.97, ranking second last in all languages.

Hence, investigating cross-lingual contamination can provide valuable perspectives for enhancing the unbalanced multilingual capabilities of LLMs.

6 Related Work

6.1 Contamination Detection

There has been a series of works for contamination detection. Mainly, they rely on a hypothesis that the test set is left in the training corpus in its original form. Hence it is possible to detect contamination by examining the perplexity of the test set (Jiang et al., 2024b), or by asking the model to generate candidate choices and compare the similarity between the generated choice and original choice (Golchin and Surdeanu, 2023), or by checking if the order of questions/choices would have an impact on model performance (Oren et al., 2023).

However, these methods, while valuable, have certain limitations. The common assumption may not hold as simple paraphrasing can alter the training distribution, potentially evading the perplexity/n-gram check (Jiang et al., 2024b). Similarly, the wrong choices in multiple-choice benchmarks can be resampled and replaced to evade generation-style detection (Golchin and Surdeanu, 2023), and sequence order sensitivity (Oren et al., 2023) can be alleviated via in-sample shuffling.

6.2 Cross-Lingual Language Modeling

Model's cross-lingual transferability has been extensively explored in recent years, particularly with the advent of Transformer models like BERT (Devlin et al., 2018) and GPT2 (Radford et al., 2019). These models have been demonstrated to effectively leverage shared linguistic features across languages, enhancing their performance on cross-lingual tasks without the need for extensive language-specific training data. For instance, studies such as XLM-R (Conneau et al., 2020), which uses a transformer-based architecture to learn language-agnostic representations, show significant improvements in cross-lingual classification tasks. Similarly, Wu and Dredze (2019) investigated the transferability of monolingual models to other languages by fine-tuning on small amounts of target language data, revealing that even limited adaptation can yield substantial gains in model performance across diverse language settings.

7 Conclusions and Future Work

In this paper, we identify a cross-lingual form of data contamination that can significantly inflate LLMs' benchmark performance while evading current detection approaches. To detect such deeply concealed contamination, we suggest a generalization-based definition of contamination and propose to detect contamination by examining the model's generalizability. With extensive experiments, we confirm that data contamination can cross language barriers. We also demonstrate that our proposed generalization-based method is able to detect not only cross-lingual but also other undisclosed contamination. In the future, we will extend our generalization-based detection approach to other potential forms of contamination. We will also explore how such cross-lingual contamination can benefit the interpretability of LLMs and the enhancement of multilingual capabilities.

Limitations

Although we conducted extensive experiments on both the injection and detection of cross-lingual contamination, the investigation of this work has some limitations: **(1)** The injection of cross-lingual contamination is only based on 7B LLMs. Whether such cross-lingual contamination universally works on other sizes of LLMs is unclear. **(2)** The benchmarks we select are all multiple-choice questions-answering, which limits the detection of contamination on other forms of benchmarks. We select the multiple-choice datasets as they are among the most widely adopted benchmarks for LLMs evaluation. **(3)** The contamination for different benchmarks and languages is injected separately, which may not reflect the real-world scenarios where multiple benchmarks and languages are blended. The main reason for not including such a multilingual and multi-benchmark mixture is the constraint on computation resources, as we employ full-parameter continual pre-training instead of parameter-efficient fine-tuning. We encourage future works to tackle these limitations and provide stronger detection methods to uncover the potential undisclosed contamination in the wild.

Ethical Considerations

We discuss the ethical considerations and broader impact of our work here: (1) **Intended Use.** We identify cross-lingual contamination to remind the community of the risk of such deeply concealed contamination. Our proposed detection method is to inspire future works to unmask other undisclosed contamination. (2) **Misuse Risks.** The experimental results and findings in this paper **should not** be used for offensive arguments or interpreted as implying misconduct of other works.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

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. In *Proceedings of NeurIPS*, volume 33, pages 1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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. *arXiv preprint arXiv:2104.08758*.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv:2402.15938*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Shahriar Golchin and Mihai Surdeanu. 2023. Time travel in llms: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*.

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*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024. Mindmerger: Efficient boosting llm reasoning in non-english languages. *arXiv preprint arXiv:2405.17386*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

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*.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Yucheng Li. 2023a. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation. *arXiv preprint arXiv:2309.10677*.

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

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

OpenAI. 2023. Gpt-4 technical report.

OpenAI. 2024. Hello gpt4o.

Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving test set contamination for black-box language models. In *The Twelfth International Conference on Learning Representations*.

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.

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

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*, abs/2307.09288.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*.

Appendices

A Details for Contamination Injection

In the experiments of injecting cross-lingual contamination, we adopt three widely adopted public benchmarks and translate their test sets into different languages for continual pre-training on two open-sourced multilingual LLMs.

A.1 Benchmark Test Sets

The benchmark datasets we use are all in the form of multiple-choice, which are licensed and intended for research use. Their details are as follows.

MMLU[1](Hendrycks et al., 2020) is a benchmark for measuring models' language understanding ability with questions in various domains, such as biology, engineering, and computer science. The test set contains around $14k$ questions in total.

ARC-Challenge[2](Clark et al., 2018) is a dataset specially designed for the evaluation of reasoning ability. Its test set consists of $2.59k$ data samples.

MathQA[3](Amini et al., 2019) is a professional mathematical question-answering dataset of which the choices are mostly Arabic numbers. There are around $2.99k$ questions in the test set.

A.2 Translation Prompt

The quality of translation is critical for our experiments. Therefore, considering both cost and quality, we utilized LLaMA3[4] to conduct the translations. The prompt template is shown below.

```
"Help me translate the following text into native <language>:
     <text>. do not use direct translation. Output your
     translation only without any explanations or notes!
     Output your translation only without any explanations
     or notes! Output your translation only without any
     explanations or notes!"
```

A.3 Continual Pre-Training

We employ continual pre-training to contaminate two multilingual LLMs (LLaMA3-8B and Qwen1.5-7B) with the original English and translated versions of benchmark test sets. The training hyperparameters are shown in Table 5. The experiment is conducted on Nvidia Tesla A100 GPUs.

[1]https://huggingface.co/datasets/hails/mmlu_no_train

[2]https://huggingface.co/datasets/allenai/ai2_arc

[3]https://huggingface.co/datasets/allenai/math_qa

[4]https://huggingface.co/meta-llama/Meta-Llama-3-8B-instruct

Batch Size	16
Learning Rate	5×10^{-5}
Optimizer	AdaFactor
Epochs	36

Table 5: Hyperparameters for continual pre-training

B Details for Contamination Detection

For contamination detection, we implement three baselines along with our proposed generalization-based method (choice confusion). The experiments of contamination detection are conducted on Nvidia RTX886 A6000 GPUs.

B.1 Shared Likelihood

Our implementation is largely based on the original codebase[5] provided by Golchin and Surdeanu (2023). To ensure a fair evaluation, we first try to reproduce the results in Golchin and Surdeanu (2023) and then adapt the code to our scenario. Due to the randomness of the permutation test and the selection of parameters in the original implementation, our reproduced results are slightly different than those in the paper but consistent in general.

B.2 Guided Prompting

We adopt GPT-4o (OpenAI, 2024) with in-context examples to judge if the model's predicted choice essentially has the same meaning as the correct one. The specific prompt template is shown below.

```
"<question>
Compare the following two sentences and determine if they
     have the same meaning. Answer with "true" if they do
     and "false" if they do not. No Explanation needed, do
     not repeat question.

Example1:
<example1>
Sentence 1: The sky is blue.
Sentence 2: The sky is clear.
Answer: false
</example1>

Example2:
<example2>
Sentence 1: She is a doctor.
Sentence 2: She practices medicine.
Answer: true
</example2>

Now, compare these sentences:

<class>
Sentence 1: [{i[0]}]
Sentence 2: [{i[1]}]

Do the two sentences have the same meaning? Answer with
     "true" if they do and "false" if they do not
Your Answer:
</class>
</question>"
```

[5]https://github.com/tatsu-lab/test_set_contamination

B.3 N-Gram Accuracy

We adopt a similar approach to that used by Xu et al. (2024). Instead of calculating the n-gram accuracy on the combined text of the question and answer, we focus on the question and choices. We identify five equally spaced indices within the combined tokens. For each index, we provide the model with the prefix text preceding the index and then determine the n-gram accuracy of the generated text. The n-gram accuracy is expected to be higher if the model is contaminated, as then the generated tokens will be more similar to the tokens within the dataset. The pseudocode for the n-gram accuracy calculation process is shown as follows.

```
# Create combined question and choice text
format_text = f"{question}{choice}"
tokens = tokenizer.tokenize(format_text)
# Find indexes for prefix texts
starting_points = np.linspace(2, len(tokens), num=5)

correct_n_grams = 0
total_n_grams = 0
for idx in starting_points:
    # Generate text based on prefix text
    gens = model.generate(tokens[:idx])
    total_n_grams += 1
    # Compare generated and original n gram tokens
    if gens[0, -n:] == tokens[idx:idx + n]):
        correct_n_grams += 1
# Calculate n-gram accuracy
n_gram_accuracy = correct_n_grams / total_n_grams
```

B.4 Choice Confusion

We utilize the LM-Eval[6] framework to evaluate different models on the original and translated versions of benchmarks to ensure fair comparisons.

The experiments of contamination detection are not limited to detecting the cross-lingual contamination injected by us intentionally. We also detect other undisclosed contamination in real-world popular multi-lingual LLMs, including LLaMA2-7B[7], Mistral-7B[8], Phi2-2.7B[9], Phi3-3.8B[10], Abel-7B[11], GLM4-9B[12], Qwen2-7B[13].

In the LM-Eval framework, the specific yaml templates we use for MMLU, ARC-Challenge, and MathQA are provided as follows.

```
# MMLU Template
task: custom_mmlu_name
dataset_path: custom_mmlu_datapath
test_split: test
fewshot_config:
  sampler: first_n
output_type: multiple_choice
doc_to_text: "{{question.strip()}}\nA. {{choices[0]}}\nB.
       {{choices[1]}}\nC. {{choices[2]}}\nD.
       {{choices[3]}}\nAnswer:"
doc_to_choice: ["A", "B", "C", "D"]
doc_to_target: answer
metric_list:
  - metric: acc
    aggregation: mean
    higher_is_better: true
metadata:
  version: 0.0
```

```
# ARC-Challenge Template
group:
  - ai2_arc
task: custom_arc_name
dataset_path: custom_arc_datapath
output_type: multiple_choice
test_split: test
doc_to_text: "Question: {{question}}\nChoices:
       {{choices.text}}\nOptions:{{choices.label}}\nAnswer:"
doc_to_choice: "{{choices.label}}"
doc_to_target: "{{choices.label.index(answerKey)}}"
should_decontaminate: true
doc_to_decontamination_query: "Question:
       {{question}}\nAnswer:"
metric_list:
  - metric: acc
    aggregation: mean
    higher_is_better: true
  - metric: acc_norm
    aggregation: mean
    higher_is_better: true
metadata:
  version: 1.0
```

```
#MathQA Template
task: custom_mathqa_name
dataset_path: custom_mathqa_datapath
output_type: multiple_choice
test_split: test
doc_to_text: "Question: {{Problem}}\nAnswer:"
doc_to_target: "{{['a', 'b', 'c', 'd', 'e'].index(correct)}}"
doc_to_choice: !function utils.doc_to_choice
should_decontaminate: true
doc_to_decontamination_query: "Question: {{Problem}}\nAnswer:"
metric_list:
  - metric: acc
    aggregation: mean
    higher_is_better: true
  - metric: acc_norm
    aggregation: mean
    higher_is_better: true
metadata:
  version: 1.0
```

There are mainly 5 hyperparameters: `Model Path`, `Task`, `Batch Size`, `Max Batch Size`, `N shot`. `Model Path` and `Task` will be set as custom paths and names, and we set `Batch Size` and `Max Batch Size` to 2 and `N shot` as 0.

[6] https://github.com/EleutherAI/lm-evaluation-harness
[7] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[9] https://huggingface.co/microsoft/phi-2
[10] https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
[11] https://huggingface.co/GAIR/Abel-7B-002
[12] https://huggingface.co/THUDM/glm-4-9b-chat
[13] https://huggingface.co/Qwen/Qwen2-7B-Instruct