An Open-Source Data Contamination Report for Large Language Models

Yucheng Li[1] **, Yunhao Guo**[3] **, Frank Guerin**[1] **, Chenghua Lin**[2]
[1] University of Surrey [2] University of Manchester
[3] Harbin Engineering University
{yucheng.li, f.guerin}@surrey.ac.uk
chenghua.lin@manchester.ac.uk

Abstract

Data contamination in model evaluation has become increasingly prevalent with the growing popularity of large language models. It allows models to "cheat" via memorisation instead of displaying true capabilities. Therefore, contamination analysis has become a crucial part of reliable model evaluation to validate results. However, existing contamination analysis is usually conducted internally by large language model developers and often lacks transparency and completeness. This paper introduces an efficient and affordable method to identify potential data contamination in LLM benchmarks. We also present an extensive data contamination report for over 15 popular large language models across six widely used multiple-choice QA benchmarks. Our experiments reveal varying contamination levels ranging from 1% to 45% across benchmarks, with the contamination degree increasing rapidly over time. Performance analysis of large language models indicates that data contamination can have significant impact on model metrics: inflated accuracy of up to 14% and 7% are observed on contaminated C-Eval and HellaSwag benchmarks, and a small increase is identified on contaminated MMLU. We also find that data contamination has grown rapidly from 2020 to 2023 and that larger models benefit more from contaminated test sets.

1 Introduction

Recent years have seen remarkable progress in large language models (LLMs) pre-trained on massive text corpora scraped from the web. However, many widely used evaluation benchmarks are also constructed from similar web sources, leading to a concerning issue of *data contamination* where examples from test sets are unintentionally included in training data. Contamination enables models to "cheat" via memorisation of test data rather than displaying true generalisation (Marie, 2023), which creates an illusion of progress, distorts model comparisons, and undermines the utility of benchmarks (Jacovi et al., 2023; Sainz et al., 2023).

Contamination analysis therefore becomes a crucial part of reliable LLM evaluation to validate the results. However, as the training data of LLMs is often not openly accessible, existing contamination analysis is mostly conducted internally by LLM developers and thus often lacks transparency and completeness. For instance, OpenAI's contamination study for GPT-4 (OpenAI, 2023) only covered the pre-training data and omitted later finetuning stages. Llama 2 (Touvron et al., 2023b) only reported contamination statistics for two of the 20+ benchmarks used in their evaluation. In addition, the implementation details of their contamination identification remain unclear. Overall, existing internal contamination studies tend to lack sufficient transparency, with minimal sharing of contamination measurements across all evaluation benchmarks, as well as training data details and code to reproduce the results. This prevents the wider research community from fully auditing the credibility of reported metrics and model capabilities.

This paper introduce a practically applicable pipeline that enables the community to identify potential data contamination from their benchmarks. Specifically, our method uses search engines and Common Crawl as a proxy for contamination identification, i.e, if a test example is found verbatim in search engine and Common Crawl, we consider it a "contaminated" sample. This is based on the observation that modern LLMs' pre-training data rely heavily on online resources and Common Crawl (Touvron et al., 2023a). Our method provides a practical and affordable solution for contamination detection that avoids the need for LLMs' training data (inaccessible in most cases) and avoids the expensive local indexing of huge corpora that often requires multi-petabyte storage (details in §4).

With this method, we present an open contam-

Findings of the Association for Computational Linguistics: EACL 2024, pages 528–541
November 12-16, 2024 ©2024 Association for Computational Linguistics

ination analysis for over 15 popular large language models on six common multiple-choice NLP benchmarks, to provide comprehensive measurements of benchmark contamination and its impact on model evaluation. The analysis includes a range of foundation models such as LLaMA (Touvron et al., 2023a), Llama-2, Yi (Yi, 2023), Mistral (Jiang et al., 2023), Baichuan (Yang et al., 2023), and Qwen (Bai et al., 2023) across multiple model sizes (7B, 13B, 30B, 34B, 65B, 70B parameters) as well as instruct-tuned models built on these foundations like Llama-2 Chat and Mistral-Instruct. Six widely used multi-choice benchmarks are assessed: Winogrande (Sakaguchi et al., 2021), AI2_ARC (Clark et al., 2018), CommonsenseQA (Talmor et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021a), and C-Eval (Huang et al., 2023).

Our analysis reveals the following key findings: 1) we detect varying levels of data contamination across benchmarks, with 1% to 45.8% of examples showing verbatim overlap with Common Crawl; 2) by comparing the contamination degree between Common Crawl Dec 2020 to Oct 2023, we find data contamination grows rapidly through time; 3) data contamination can significantly inflate model performance: accuracy increases of 14% and 7% were found on C-Eval and HellaSwag, and a slight increase was found on MMLU; 4) we find larger models tend to benefit more from data contamination than smaller models, perhaps due to their more powerful memorisation capacities; 5) finally, we show our results align well with Llama's original contamination reports, demonstrating the effectiveness of our method. Our data and code can be found in `https://github.com/liyucheng09/Contamination_Detector`.

2 Data Contamination

What is data contamination? Data contamination refers to the phenomenon that examples from the test set are also found in the training data. This might lead to the evaluation failing to accurately reflect models' capabilities, as models can cheat by memorising instead of learning to generalise. There are two primary types of data contamination (Dodge et al., 2021): *input-only contamination* refers to cases where only the input appearing in the pretraining corpus, and *input-and-label contamination* occurs when both inputs and their labels are present. The latter is generally more problematic, as models can directly memorise input-output pairs. But the first may still cause issues as models may learn from the context.

How common is data contamination? Data contamination appears to be quite widespread across commonly used NLP benchmark datasets based on findings from recent studies. Dodge et al. (2021) and Elazar et al. (2023) audited well-known big language corpora such as C4, The Pile, and RedPajama, revealing contamination rates ranging from 0% to over 50% on GLUE and SuperGLUE benchmarks. The GPT-3 study (Brown et al., 2020) found over 90% of examples in Quac, SQuADv2, and DROP were flagged as contaminated. FLAN (Wei et al., 2021) evaluations identified 7 out of 26 datasets exhibiting a serious contamination ratio of 50% and over. Llama-2 (Touvron et al., 2023a) reported over 16% of MMLU examples are contaminated and about 11% are seriously contaminated (more than 80% token leakage). GPT-4 (OpenAI, 2023) uses academic exams and NLP benchmarks for model evaluation. While 4 out of 34 exams were found to have zero contamination (e.g., Leetcode and Bar Exam), 9 out of 34 showed over 20% of instances marked as dirty examples. Sainz et al. (2024) provide a comprehensive collection of evidence of data contamination in NLP datasets and models, where data contamination is found in hundreds of widely-used datasets, popular pre-training corpora, and state-of-the-art LLMs.

How to identify data contamination? Dodge et al. (2021) take a straightforward approach to detect exact matches between test set examples and the pretraining data after normalising for capitalisation and punctuation. *Exact match* here means the entire input of an evaluation text is found in the training data. The GPT-3 paper (Brown et al., 2020) uses n-gram overlap to identify contamination, treating any examples with 13-gram co-occurrence in both test sets and training data as dirty examples. Llama-2 matches on verbalised and tokenized input to allow a token-level approach to identify contamination. It also involves a "skip-gram budget" to allow slight variants in overlapping. Overall, existing approaches usually use substring matching between evaluation examples and training data to identify data contamination. However, if we have no access to the training data, which is often the case for most recent closed models, it is extremely difficult to reveal contamination by observing models themselves. Pioneering studies propose to identify data contamination by measuring perplexity of test examples (Li, 2023), asking models to reconstruct test examples verbatim (Golchin and Surdeanu, 2023), or examining models' preference on test sample ordering (Oren et al., 2023). In this paper, we use Common Crawl as a proxy for the training data of LLMs, as it often constitutes a significant part of it, thereby avoiding the need to access the full training dataset.

To what extent does data contamination affect model evaluation? While contaminated data can potentially inflate scores, models do not necessarily perform worse on clean subsets or better on dirty subsets across all datasets. The degree of impact likely depends on many factors like the dataset characteristics, model scale, and nature of the pre-training data. For instance, GPT-3 (Brown et al., 2020) showed a marginal 1-2% performance drop on clean subsets for PIQA and ReCoRD, compared to a significant 6% drop on clean set of SQuAD as 94% of its test examples were contaminated. Roberts et al. (2023) found a significant association between a code problem's presence on GitHub and GPT-4's pass rate for that problem. But on other academic tests, GPT-4 showed little performance difference on the clean and contaminated test sets (OpenAI, 2023). Touvron et al. (2023b) reported a 15.3 point and a 9.8 point gap from Llama-2 70B by comparing the its performance on the clean and dirty sets of HellaSwag and MMLU-Humanities.

Mitigating Data Contamination. There exist many efforts to address the issue of data contamination in the evaluation of LLMs. One recent promising attempt is to collect the most up-to-date data from the Internet[1] and dynamically update existing benchmarks or build novel benchmarks automatically with this latest information, e.g., recent news, academic papers, etc (Li et al., 2024b; White et al., 2024; Zhu et al., 2023). By leveraging the most recent information to construct the test set, it not only addresses data contamination, but also avoids potential cheating methods on leaderboards, such as training-on-test-set. In addition, recent studies were proposed to assess LLMs without relying on any specific benchmarks by pair-wise model comparison using crowd-sourcing platforms, e.g., ChatArena[2], or evaluating LLMs with data compression (Li et al., 2024c).

3 Benchmarks for Language Models

Clean and robust benchmarks are important to guide further progress of various models in NLP. Popular benchmarks used to evaluate large language models include:

> **Comprehensive**: MMLU, Big Bench (Srivastava and et al., 2023), AGI Eval (Zhong et al., 2023), C-Eval
>
> **Commonsense reasoning**: PIQA (Bisk et al., 2019), SIQA (Sap et al., 2019), HellaSwag, WinoGrande, ARC, OpenBookQA (Mihaylov et al., 2018), CommonsenseQA
>
> **World knowledge**: NaturalQuestions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017)
>
> **Reading comprehension**: SQuAD (Rajpurkar et al., 2018), QuAC (Choi et al., 2018), BoolQ (Clark et al., 2019)
>
> **Math**: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b)
>
> **Code**: HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021)

The construction of many of these relies heavily on online materials, therefore they are highly prone to data contamination as their source spreads on the Internet. Here we analyse six representative multi-choice QA benchmarks: MMLU, C-Eval, Winogrande, CommonsenseQA, ARC, and HellaSwag. These benchmarks have been selected due to their varied sources and potential susceptibility to data contamination. MMLU, ARC, and C-Eval, which are academic test-based benchmarks, were compiled from online `.docx`/`.pdf` files using techniques like OCR, typically assumed to be less affected by data contamination as such files are often not indexed by online crawlers. However, C-Eval stands out as it is a non-English (Chinese) benchmark, offering an opportunity to assess the impact of non-English benchmarks on language models. Winogrande, uniquely human-authored from scratch, allows examination of whether manually created benchmarks are less prone to data contamination. CommonsenseQA and HellaSwag, both Internet-sourced, differ in their source popularity; while CommonsenseQA is built upon the

[1] `https://huggingface.co/RealTimeData`

[2] `https://lmarena.ai/`

less influential ConceptNet, HellaSwag is sourced from the more popular WikiHow. This selection of benchmarks provides a comprehensive overview of how different sourcing and construction methods might influence the presence and extent of data contamination in language model evaluations.

4 Our Approach

The central goal of data contamination analysis is to categorise test samples as either clean or contaminated and then evaluate models separately on the clean and contaminated samples to assess the impact of contamination on the performance metrics. In this section, we describe our methodology to identify contaminated test samples. The basic idea is to check whether test examples appear verbatim in both search engine and Common Crawl. We use search engine and Common Crawl because they are accessible, affordable and often comprise the majority of pre-training data for large language models, e.g., Common Crawl constitutes over 80% in GPT-3 and LLaMA training data (Brown et al., 2020; Touvron et al., 2023a) and the remainder of pre-training data also relies heavily on online resources.

We identify data contamination in two steps. First, 1) we use the Bing Search API to check if test examples appear verbatim online, which indicates their potential inclusion in LLMs' pre-training data. Second, 2) we verify if the page containing verbatim test examples we found in step 1) were also indexed in Common Crawl. Here we check the presence of test examples in both search engine and Common Crawl to address a possible false positive issue. And since the second step only involves URL search within Common Crawl instead of string retrieval, it avoids the expensive local indexing of the entire Common Crawl. The *search window* of our contamination detection starts from 2017 and ends at the knowledge cutoff with respect to different LLMs. This is realised via adjusting the *freshness* parameter in the Bing API and using the appropriate indexes of Common Crawl during the identification. Note that we use the release date as the end of the search window for LLMs for which we are not aware of their knowledge cutoff.

To construct the search queries, we verbalise examples accordingly and make sure the question and the correct answer are involved in the queries. For example:

> **Question**: The flaw in Anderson's ACT theory was that some considered it ____.
>
> **Choices**:
>
> A: 'Only applicable to a motor system',
>
> B: 'Untestable and thus, of uncertain scientific value',
>
> C: 'Lacking in definition for its elements',
>
> D: 'Overly complex in explaining the operation of cognition',
>
> **Answer**: B
>
> ***Verbalised Query***: The flaw in Anderson's ACT theory was that some considered it untestable and thus, of uncertain scientific value.

We verbalise this multi-choice question to a query by filling the correct answer in the blank. We do not include other options in the query, because, as discussed in Section 2, the presence of other options does not matter. The question and answer are the key for identifying data contamination. If there is no blank in the question, we simply append the answer after the question to form the query.

To identify overlap between test samples and training data, existing methods often rely on exact string matches. For example, Brown et al. (2020) use N-gram overlap ranging from 8-grams to maximum 13-grams for all evaluation tasks. GPT-4's criterion for contamination is sub-string matching with at least 50 characters (OpenAI, 2023). However, according to our manual analysis, we find the approach of exact string matches often leads to false negative in our pipeline. Touvron et al. (2023b) propose a more fine-grained method that assesses contamination in the token-level and involves a small "skipgram budget" to accommodate slight variations of sequences. However, their exact implementation details remain unclear. We instead simply compute the METEOR (Banerjee and Lavie, 2005) score between matched pages and the queries to quantify the extent of overlap. We consider examples with a METEOR recall score over 0.75 as contaminated cases. This method tolerates minor inserted phrases and word form variations, which greatly mitigates the false negative issue that strict string matching would miss. To avoid potential false positives, we configure our method with two key settings: 1) an order penalty (gamma of 0.8) for METEOR ensures matches respect sequence; 2) matching is constrained to a window

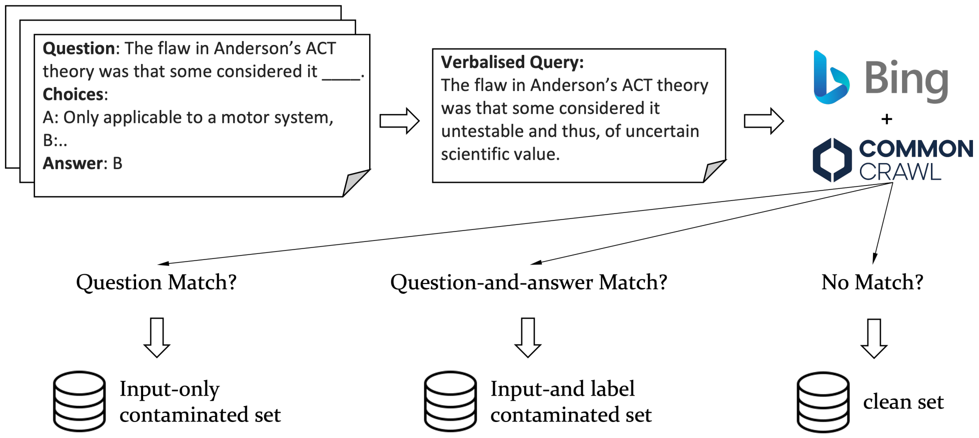

Question: The flaw in Anderson's ACT theory was that some considered it ____.
Choices:
A: Only applicable to a motor system,
B:..
Answer: B
Verbalised Query:
The flaw in Anderson's ACT theory was that some considered it untestable and thus, of uncertain scientific value.
Bing
+
COMMON CRAWL
Question Match?
Question-and-answer Match?
No Match?
Input-only contaminated set
Input-and label contaminated set
clean set

Figure 1: The categorisation of contaminated test samples.

Dataset	Split	#Total	#Online	#Total Contamination	#Input-only Contamination	#Input-and-label Contamination
ARC_c	Test	1172	372	336 (28.7%)	53 (4.5%)	283 (24.1%)
CommonsenseQA	Dev	1221	44	20 (1.6%)	3 (0.2%)	17 (1.4%)
Winogrande	Dev	1267	54	14 (1.1%)	0 (0.0%)	14 (1.1%)
C-Eval	Dev	1346	618	616 (45.8%)	69 (5.1%)	547 (40.6%)
HellaSwag	Dev	10042	1690	1247 (12.4%)	46 (0.4%)	1201 (12.0%)
MMLU	Test	13987	4285	4077 (29.1%)	678 (4.8%)	3399 (24.3%)

Table 1: Data contamination statistics for multi-choice QA benchmarks. Search window: 2020.10-2023.10.

up to 2× the query length, preventing partial or out-of-context matches. We compare our approach with Llama-2's and other contamination detection approaches in Section 7.2.

According to Section 2, here we distinguish two types of data contamination: 1) *input-only contamination* where only question is presented in the matched pages but not answer; 2) *input-and-label contamination* where both question and answer occur in the matched pages. In the upcoming sections, these two types of data contamination are compared and analysed separately.

5 Contamination Statistics for Multi-Choice Benchmarks

Our analysis reveals varying levels of data contamination across six multi-choice QA benchmarks, as shown in Table 1. According to the table, we have the following key findings. First, Academic test-based benchmarks like MMLU and C-Eval, despite being collected through methods like OCR, exhibit the highest levels of contamination (29.1% and 45.8%, respectively). This high rate is attributed to the widespread distribution and communication of academic test examples, making them more prone to sharing and discussion. In contrast, benchmarks manually created from scratch like Winogrande demonstrate minimal contamination (1.1%), as they avoided using Internet resources in their benchmark construction. Third, we find significant differences among Internet-sourced benchmarks. For example, CommonsenseQA has low contamination (1.6%) but HellaSwag is much higher (12.4%). This variation might stem from different popularity of the sources: ConceptNet, the source of CommonsenseQA is less popular than WikiHow, the source of HellaSwag. Finally, we find most contamination belongs to *input-and-label* contamination, indicating that models often find the answer alongside the question for contaminated test samples.

We also illustrate how data contamination increases over time, as shown in Figure 2. In the figure, benchmarks such as CommonsenseQA and Winogrande maintain very low rates of contaminated data, with increases of just 0.3% and 0.2% over the past three years. However, benchmarks collected from academic tests like ARC, MMLU, and C-Eval have experienced a substantial increase in contamination, with up to 21% of examples flagged as contaminated during the same period. This shows how test content in academic benchmarks can easily propagate across the Internet,

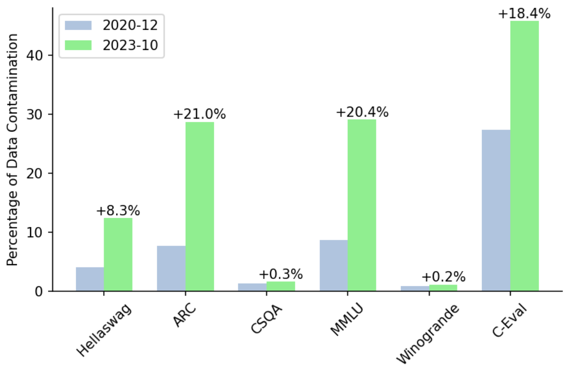


Figure 2: Increase in Data Contamination from the period of *2017-2020* to *2020-2023*. CSQA stands for CommonsenseQA.

which can be a serious issue for academic test based language model benchmarks. We also observe a moderate 8.3% increase for HellaSwag, further demonstrating the increasing risk of data contamination for Internet sourced benchmarks.

In Figure 3, we illustrate where these HellaSwag contaminated test samples come from. We discover that data contamination manifests in a centralised fashion, which means contaminated test samples are not evenly distributed across domains. Instead, they are significantly concentrated in specific domains and rare in others. This finding is meaningful as it reveals the possibility that blocking specific domains during training data collection might alleviate the issue of data contamination. You can find more domain analysis and contamination examples in Appendix A.

6 Impact of Contamination on Model Performance

To assess how data contamination impacts model evaluation, we test popular large language models on contaminated and clean splits of each benchmark. As shown in the previous section, we categorise benchmarks into four subsets: 1) the clean set; 2) not clean set; 3) *input-only contaminated* set; and 4) *input-and-label contaminated* set. Note that 'not clean' = *input-only* + *input-and-label* contamination. Since CommonsenseQA and Winogrande are shown to be just marginally contaminated, we focus on MMLU, C-Eval, HellaSwag and ARC in these experiments. We only report *input-only contaminated* performance for MMLU, as the other benchmarks have too few samples of this type to yield robust results. Following previous implementations (Touvron et al., 2023b; OpenAI, 2023), we use a zero-shot setting for HellaSwag and ARC where only the questions and choices are given in the input, and a 5-shot setting for MMLU and C-Eval where 5 demonstrations are given in the prompts. We employ the third party LLMs evaluation platform *OpenCompass* (OpenCompass, 2023) in our experiments to provide in-context demonstrations, prompts, and metrics computing. We use perplexity to obtain the inference result, i.e., taking the choice with the lowest perplexity as the predicted answer. The results are presented in Table 2. We report model accuracy on the clean set, and the performance difference for not clean, and contaminated sets when they are compared to the clean set. We use ↑ to indicate an advantage against the clean set, and ↓ to indicate an accuracy decrease. For Llama-1,2 series models, we use the search window of 2017-2020 according to their reported training data collection period. For all other models we use an estimated search window of 2017.01-2023.10 as their exact training data collection periods are unknown.

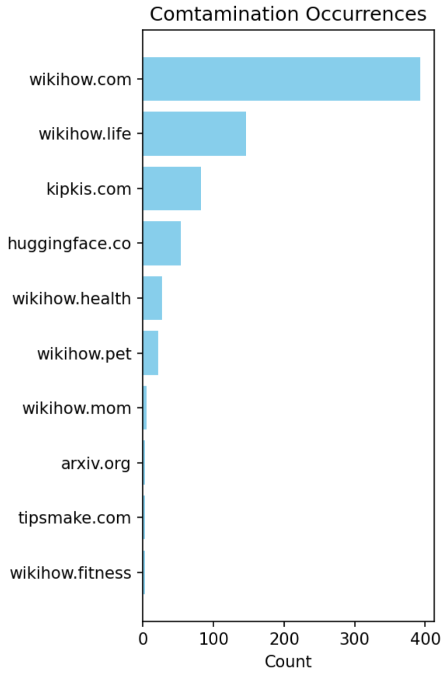


Figure 3: Domain analysis for data contamination in HellaSwag.

English Benchmarks. Based on the table, we find data contamination does not always improve model performance. Instead, the impact depends on both the specific benchmark and model scale. On HellaSwag and ARC benchmarks, many models achieve better metrics on contaminated subsets.

	MMLU				Hellaswag			ARC			Average	
	Clean	Not Clean	I-O Con.	I-L Con.	Clean	Not Clean	I-L Con.	Clean	Not Clean	I-L Con.	Clean	Not Clean
LLaMA 7B	.3427	↓ .0180	↓ .0060	↓ .0204	.6394	↑ .0302	↑ .0333	.3627	↓ .0179	↓ .0460	.4483	↓ .0019
LLaMA 13B	.4652	↓ .0145	↓ .1036	↑ .0034	.7073	↑ .0840	↑ .0836	.3924	↓ .0361	↓ .0591	.5216	↑ .0111
LLaMA 30B	.5690	↓ .0166	↓ .1127	↑ .0027	.7412	↑ .0501	↑ .0497	.4249	↑ .0349	↑ .0418	.5784	↑ .0228
LLaMA 65B	.6364	↓ .0120	↓ .1510	↑ .0160	.7613	↑ .0474	↑ .0478	.4276	↑ .0437	↑ .0391	.6084	↑ .0264
Llama-2 7B	.4310	↑ .0076	↓ .0885	↑ .0270	.6746	↑ .0471	↑ .0436	.3803	↑ .0565	↑ .0364	.4953	↑ .0371
Llama-2 13B	.5647	↓ .0348	↓ .1026	↓ .0212	.8254	↓ .0167	↓ .0254	.4221	↑ .0147	↓ .0054	.6041	↓ .0123
Llama-2 70B	.6884	↑ .0025	↓ .1214	↑ .0275	.7726	↑ .0622	↑ .0729	.4555	↑ .1077	↑ .1112	.6388	↑ .0575
Llama-2 Chat 7B	.4062	↓ .0211	↓ .1248	↓ .0002	.6760	↑ .0845	↑ .0872	.3701	↑ .0773	↑ .1299	.4841	↑ .0469
Llama-2 Chat 13B	.5417	↓ .0319	↓ .1219	↓ .0138	.7341	↑ .0714	↑ .0759	.4334	↑ .1192	↑ .1435	.5697	↑ .0529
Llama-2 Chat 70B	.6324	↓ .0165	↓ .1324	↑ .0068	.7576	↑ .0997	↑ .0765	.4343	↑ .0994	↑ .0272	.6081	↑ .0609
Mistral 7B	.6501	↓ .0210	↓ .1064	↓ .0038	.8533	↓ .0246	↓ .0207	.4720	↑ .0543	↑ .1049	.6585	↑ .0029
Mistral-FT 7B	.5576	↓ .0173	↓ .1087	↑ .0011	.7168	↓ .0477	↓ .0441	.4426	↑ .0574	↑ .1151	.5723	↓ .0025
Yi 6B	.6481	↓ .0094	↓ .0912	↑ .0070	.7628	↓ .0095	↓ .0011	.4380	↑ .0488	↑ .0620	.6163	↑ .0100
Qwen 7B	.5785	↓ .0120	↓ .0917	↑ .0040	.9153	↓ .0009	↑ .0033	.4096	↑ .0509	↑ .0327	.6345	↑ .0127
Baichuan2 7B	.5594	↓ .0274	↓ .1119	↓ .0103	.7494	↓ .0295	↓ .0254	.3710	↓ .0552	↓ .0056	.5599	↓ .0374

Table 2: Model accuracy on the clean set and accuracy difference on not-clean, *input-only* contaminated (denoted as I-O Con.) and *input-and-label* contaminated (denoted as I-L Con.) sets, when compared to the clean set. Significant accuracy inflation (more than 5%) is highlighted with underlines.

	Clean	Not Clean	I-L Contam.
Llama-2 7B	.3135	.3344 ↑	.3364 ↑
Mistral 7B	.4715	.4545 ↓	.4607 ↓
Yi 6B	.6718	.8003 ↑	.8117 ↑
Qwen 7B	.5619	.6169 ↑	.6289 ↑
Baichuan2 7B	.5508	.5649 ↑	.5887 ↑
Average	.4582	.4912 ↑	.5012 ↑

Table 3: Data contamination analysis on C-Eval. I-L Contam. indicates input-and-label contamination.

However, on MMLU tasks we observe no consistent enhancement across models. We also find that larger language models appear more capable of exploiting data contamination to achieve better performance. For instance, LLaMA-2 70B displays increased metrics on most contaminated subsets. In contrast, the 13B LLaMA-2 only outperforms on contaminated ARC. In addition, LLaMA-2 70B achieves a larger advantage on contaminated sets (5%) compared to 3% inflation of the 7B variant. This could be due to the more powerful memorisation capacity in larger language models (Carlini et al., 2022). Finally, we find that input-only contamination does not lead to inflation of metrics. This suggests that contamination has little effect when it does not give away the answer. Surprisingly, we also observe that models demonstrate significantly worse performance on the input-only contaminated set. This may be because the absence of their labels online suggests that no one has provided a solution for these test samples, because they are inherently more challenging. Input-and-label contamination, on the contrary, often leads to more notable accuracy increases, making it the key issue to address for data contamination.

Non-English Benchmark. In Table 3, we present contamination analysis on the non-English benchmark C-Eval. Among the tested models, Llama and Mistral are considered pure English models, while Yi, Qwen, and Baichuan are pretrained as multilingual language models. We find the pure English models, Llama and Mistral, do not exhibit notable performance increases on C-Eval's contaminated subsets. However, the multilingual large language models all demonstrate significant performance advantages on dirty subsets. Yi 6B even achieves a 14% higher accuracy score on the input-and-label contaminated set, proving the potential for serious distortion of evaluation results.

What is the threshold of overlap for a test example to affect model prediction? We illustrate how the METEOR score, which measures sentence similarity, correlates with model performance on test samples. The METEOR metric measures the similarity between two sentences. For instance, a test sample with a METEOR score of 0.8 indicates high equivalence between that test case and sentences in training data. In Figure 4, we group test samples by METEOR score and present the accuracy achieved on those groups by Llama-2 70B across four benchmarks. On ARC, HellaSwag, and C-Eval, a general upwards accuracy trend emerges as METEOR rises, indicating that models attain higher metrics when more verbatim overlapping samples exist in the training data. In

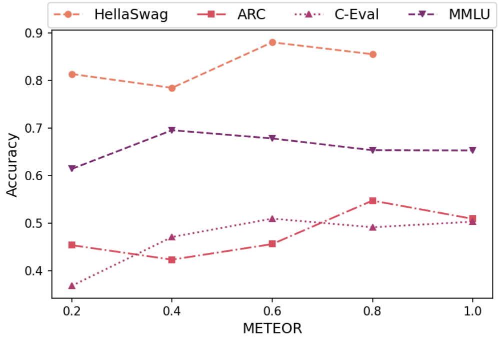


Figure 4: Accuracy of Llama-2 70B for test examples with different METEOR score.

Method	Contam. (%)	Acc. Inflation (%)
	HellaSwag	
Ground Truth	8.4%	7.42%
Ours	8.3%	7.29%
minK-20%	not-applicable	14.29%
	MMLU	
Ground Truth	11%	2.00%
Ours	9.7%	2.75%
minK-20%	not-applicable	11.54%

Table 4: Comparison against to ground truth (Touvron et al., 2023b) and minK-20% (Shi et al., 2023)

essence, substantial text duplication enables exploitation through memorisation, inflating model scores.

7 Discussion

7.1 Existing Methods to Mitigate Data Contamination

Several techniques have previously been proposed to mitigate the data contamination issue in language model evaluation. Our findings provide some novel insights on the effectiveness of these approaches.

Blocklisting benchmark sources. Blocking sources of benchmarks in training data collection is a common way to avoid data contamination. In our paper, we further demonstrate the feasibility of this method. As shown in Figure 3, the distribution of data contamination is very centralised, so blocking only a small set of domains can significantly alleviate the issue of data contamination. However, we also find blocklisted links quickly expire but content spreads, making the blocklist ineffective over time. For instance, we test the contamination blocklist in the first release of MMLU[3], and we found the given blocklist only avoids 1.5% of contaminated cases we detected in §5. If we adopt a more aggressive method that skips all domains in the blocklist, it still just avoids 21% of contaminated cases. This suggests content used in MMLU spreads rapidly, which emphasises the necessity to update the blocklists regularly.

Avoid using data that appears with its solution on the Internet (Jacovi et al., 2023). According to our results, avoiding the presence of answers is a feasible method and can indeed prevent memorising exact answers. As shown in Table 2, we found input-only contamination typically does not lead to metrics inflation compared to input-and-label contamination. This suggests that as long as the contamination does not reveal the answer, it is unlikely that the model can achieve an unfair advantage. A better solution is to completely avoid using online resources in benchmark construction. Winogrande is a good example that is barely affected by data contamination since its test examples were developed with fresh, human-authored content.

Protecting test data from automatic crawlers via encryption and forbidding further distribution (Jacovi et al., 2023). Forbidding further distribution of benchmarks can indeed prevent data contamination to some extent. This was proven in our Figure 3, where some contaminated cases are from huggingface.co, a dataset sharing platform. However, forbidding further distribution of the test data also significantly limits the popularity of benchmarks. For example, benchmarks such as HellaSwag and C-Eval make their test sets nonpublic to avoid potential data contamination issues. However, this also makes popular third party model evaluation platforms turn to using their validation sets instead of the test sets, as the platform hosts can access the answers in the validation sets to conduct the assessment (OpenCompass, 2023). Actually, most researchers tend to evaluate their models on publicly available splits rather than restricted ones, even if the latter have lower contamination risk. Therefore, benchmarks should consider balancing robustness against ease of adoption by the community.

7.2 Comparison to Ground Truth and Other Methods

The data contamination analysis in the original Llama-2 paper is quite incomplete, presenting results for only HellaSwag and MMLU benchmarks. However, we can still compare our results to theirs (considered as ground truth) to show the effective-

[3] https://people.eecs.berkeley.edu/~hendrycks/data.tar

ness of our method. We also include minK (Shi et al., 2023), a recent SOTA approach for data contamination detection, in our comparison. The optimal setting reported in Shi et al. (2023) was used here, that considers top 20% probability in their detection process (thus minK-20%). As shown in Table 4, our method exhibits accuracy in contamination identification and achieves results very similar to the ground truth. Specifically, our results show less than 1% error on accuracy and less than 2% error in the percentage of contamination compared to the ground truth. Compared to minK, our method not only achieves a more accurate result in accuracy inflation, but can also provide the percentage of contamination for a given benchmark, which is not applicable to the minK approach.

8 Conclusion

This paper conducted an extensive data contamination analysis for popular large language models on six multi-choice QA benchmarks. We identified varying levels of test set contamination, ranging from 1% to 47% across benchmarks. We also found that data contamination can lead to increased metrics: data contamination in ARC and HellaSwag generally allows models to achieve significantly higher accuracy, but contamination in MMLU has less of an impact on model's performance, although it did increase performance on a specific sub-set of MMLU. Our findings offer a transparent perspective on data contamination, emphasising its significance as an urgent issue within the evaluation community.

9 Limitation

The use of search APIs in our method will cost around $10 per 1,000 queries with Bing. We spent about $110 in total for querying the entire MMLU. But this number includes trying different settings. So one can expect to spend much less in their own experiments. Nevertheless, the cost of search APIs is still much more affordable compared to hosting the entire Common Crawl locally, which would require dealing with multi-petabyte data. Another possible limitation is the restriction on lengthy queries by search engines, which prevents the analysis of benchmarks with long input passages, such as reading comprehension. Finally, LLM developers may use training data that does not appear on the Internet, such as user-generated data, which is out of the scope of our method. However, we argue that this would hardly lead to new data contamination, as it is unlikely that users include NLP benchmark examples in their generated data.

Future attempts will include scanning lengthy input examples via sequence chunking, and developing perplexity-based approaches to detect contaminated examples without requiring full passage matching.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

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.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

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.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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. *arXiv preprint arXiv:1905.10044*.

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*, abs/1803.05457.

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.

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*.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. 2023. What's in my big data? *arXiv preprint arXiv:2310.20707*.

Shahriar Golchin and Mihai Surdeanu. 2023. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding.

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. *arXiv preprint arXiv:2305.08322*.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.

Mintong Kang, Nezihe Merve Gürel, Ning Yu, Dawn Song, and Bo Li. 2024. C-rag: Certified generation risks for retrieval-augmented language models. *arXiv preprint arXiv:2402.03181*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023a. Compressing context to enhance inference efficiency of large language models.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024a. Finding challenging metaphors that confuse pretrained language models. *arXiv preprint arXiv:2401.16012*.

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

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024c. Evaluating large language models for generalization and robustness via data compression. *arXiv preprint arXiv:2402.00861*.

Yucheng Li, Chenghua Lin, and Frank Guerin. 2022. Cm-gen: A neural framework for chinese metaphor generation with explicit context modelling. In *Proceedings of the 29th international conference on computational linguistics*, pages 6468–6479.

Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin, and Loïc Barrault. 2023b. Framebert: Conceptual metaphor detection with frame embedding learning. *arXiv preprint arXiv:2302.04834*.

Yucheng Li, Yan Yang, Qinmin Hu, Chengcai Chen, and Liang He. 2021. An argument extraction decoder in open information extraction. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 313–326. Springer.

Yufei Li, Zexin Li, Yingfan Gao, and Cong Liu. 2023c. White-box multi-objective adversarial attack on dialogue generation. *arXiv preprint arXiv:2305.03655*.

Yufei Li, Zexin Li, Wei Yang, and Cong Liu. 2023d. Rt-lm: Uncertainty-aware resource management for real-time inference of language models. *arXiv preprint arXiv:2309.06619*.

Yufei Li, Xiao Yu, Yanchi Liu, Haifeng Chen, and Cong Liu. 2023e. Uncertainty-aware bootstrap learning for joint extraction on distantly-supervised data. *arXiv preprint arXiv:2305.03827*.

Weimin Lyu, Xiao Lin, Songzhu Zheng, Lu Pang, Haibin Ling, Susmit Jha, and Chao Chen. 2024. Task-agnostic detector for insertion-based backdoor attacks. *arXiv preprint arXiv:2403.17155*.

Weimin Lyu, Songzhu Zheng, Haibin Ling, and Chao Chen. 2023. Backdoor attacks against transformers with attention enhancement. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

Benjamin Marie. 2023. The decontaminated evaluation of gpt-4. Accessed: 2023-07-28.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

OpenAI. 2023. Gpt-4 technical report.

OpenCompass. 2023. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. 2023. Data contamination through the lens of time. *arXiv preprint arXiv:2310.10628*.

Oscar Sainz, Jon 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. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

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*.

Zezheng Song, Jiaxin Yuan, and Haizhao Yang. 2024. Fmint: Bridging human designed and data pretrained models for differential equation foundation model. *arXiv preprint arXiv:2404.14688*.

Aarohi Srivastava and et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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

Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, et al. 2023. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. *arXiv preprint arXiv:2310.02954*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Yi. 2023. A series of large language models trained from scratch by developers at 01-ai. https://github.com/01-ai/Yi.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations*.

A More Information about Contamination in Multi-Choice QA Benchmarks

To provide a straightforward impression, we provide some example of data contamination from the MMLU benchmark as shown in Figure 7. In Figure 7 (a), the METEOR recall score between the test question and matched example was 0.9275, well above the 0.8 contamination threshold, indicating a clear leakage of this test example in the training data of Llama models. While minor formatting differences exist, the near-complete overlap constitutes concerning *input-and-label* contamination that allows models to memorise rather than generalise. However, in Figure 7 (b) we find no answer choices and the correct answer in that page, which makes it a *input-only* contamination case. While *input-only contamination* poses a lower risk for direct label leakage, it can still allow models unfair advantage if exposed to the questions during training.

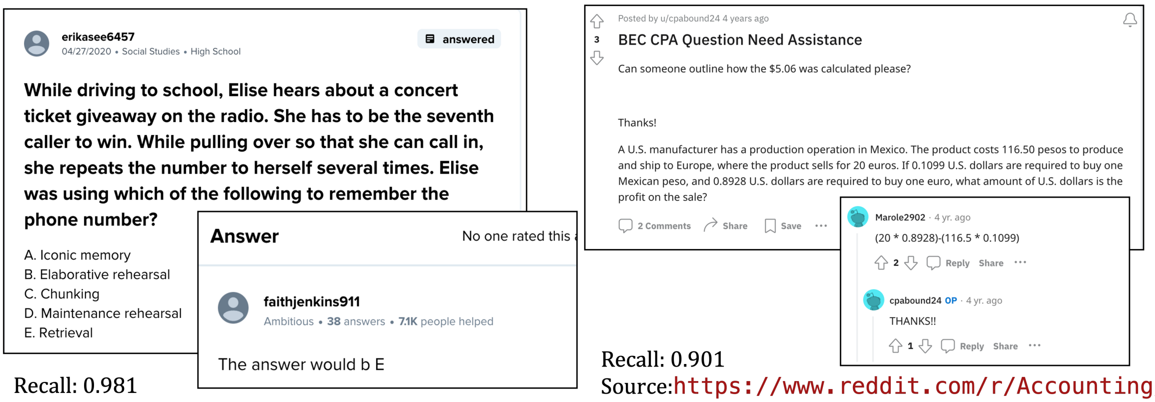
We also present the domain visualisation for contaminated test sample in ARC (see Figure 5) and MMLU (see Figure 6).

B More Results

In Table 5, we present more detailed statistics of Llama models' performance on different categorises of MMLU benchmark.

Model	MMLU			MMLU-Humanities			MMLU-STEM			MMLU-Social-Science			MMLU-Other		
	Clean	I-O Con.	I-L Con.	Clean	I-O Con.	I-L Con.	Clean	I-O Con.	I-L Con.	Clean	I-O Con.	I-L Con.	Clean	I-O Con.	I-L Con.
Llama 7B	34.27	33.67	32.23	33.69	25.76	34.22	30.79	33.04	30.67	37.40	38.10	31.59	35.64	35.23	33.60
Llama 13B	46.52	36.15	46.86	43.79	43.94	53.38	37.78	27.73	37.37	55.55	49.52	51.33	50.31	41.48	49.53
Llama 30B	56.90	45.63	57.17	55.02	59.09	64.36	46.10	36.28	47.84	65.91	58.10	63.18	61.84	51.14	57.09
Llama 65B	63.64	48.54	65.25	63.71	56.06	74.73	52.58	41.59	54.09	72.08	59.05	74.17	67.30	52.84	62.21
Llama 2 7B	43.10	34.26	45.80	41.90	45.45	55.57	34.38	26.55	36.82	49.74	45.71	49.20	47.30	38.07	46.29
Llama 2 13B	56.47	46.21	54.35	55.73	59.09	60.91	44.27	37.17	44.17	64.22	60.95	61.69	62.64	50.00	54.39
Llama 2 70B	68.84	56.71	71.59	65.78	74.24	79.28	57.18	45.43	61.52	81.12	67.62	80.15	73.13	65.34	68.96

Table 5: Llama series models' performance (accuracy) across different categories of MMLU. *I-O Con.* and *I-L Con.* indicate *input-only* contamination and *input-and-label* contamination respectively.



erikasee6457
04/27/2020 • Social Studies • High School

answered

While driving to school, Elise hears about a concert ticket giveaway on the radio. She has to be the seventh caller to win. While pulling over so that she can call in, she repeats the number to herself several times. Elise was using which of the following to remember the phone number?

A. Iconic memory
B. Elaborative rehearsal
C. Chunking
D. Maintenance rehearsal
E. Retrieval

Answer No one rated this

faithjenkins911
Ambitious • 38 answers • 7.1K people helped

The answer would b E

Recall: 0.981
Source: https://answers-learning.com/social-studies/question15992997

(a)

Posted by u/cpabound24 4 years ago

3

BEC CPA Question Need Assistance

Can someone outline how the $5.06 was calculated please?

Thanks!

A U.S. manufacturer has a production operation in Mexico. The product costs 116.50 pesos to produce and ship to Europe, where the product sells for 20 euros. If 0.1099 U.S. dollars are required to buy one Mexican peso, and 0.8928 U.S. dollars are required to buy one euro, what amount of U.S. dollars is the profit on the sale?

2 Comments Share Save ···

Marole2902 · 4 yr. ago
(20 * 0.8928)-(116.5 * 0.1099)
2 Reply Share ···

cpabound24 OP · 4 yr. ago
THANKS!!
1 Reply Share ···

Recall: 0.901
Source: https://www.reddit.com/r/Accounting/comments/cguwma/bec_cpa_question_need_assistance/

(b)

Figure 7: An example of *input-and-label* (a) and *input-only* (b) contamination from MMLU.

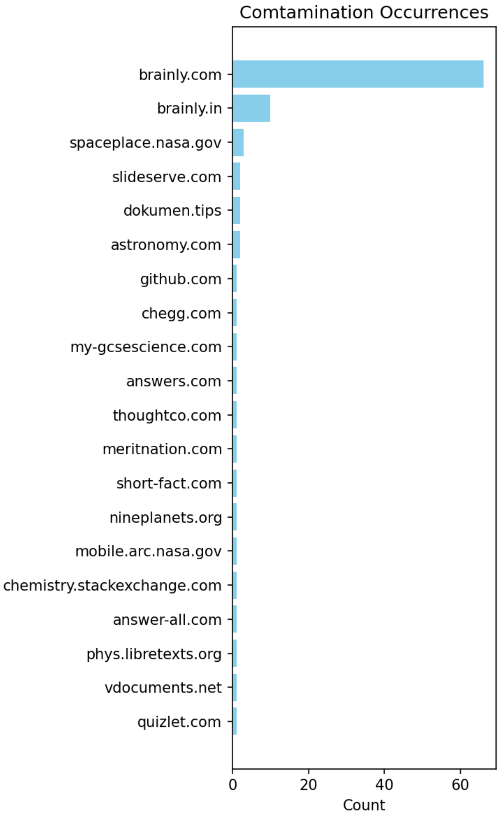


Figure 5: Domain analysis for data contamination in ARC.

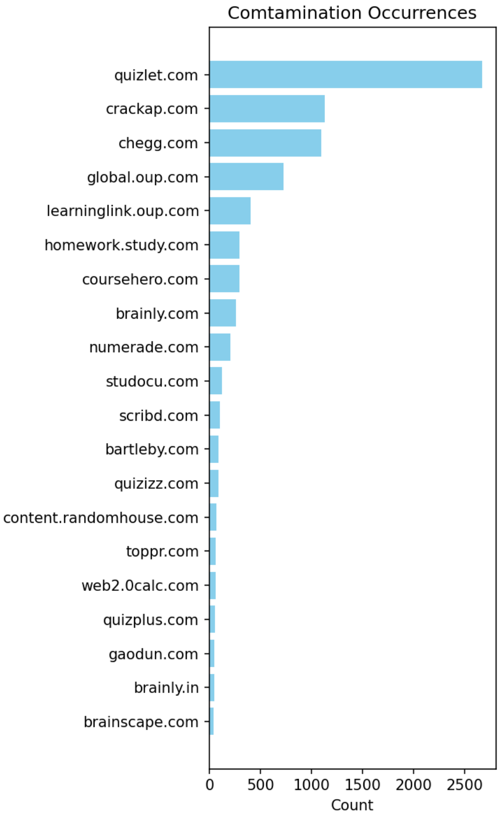


Figure 6: Domain analysis for data contamination in MMLU.