Investigating Data Contamination in Modern Benchmarks for Large Language Models

Anonymous ACL submission

Abstract

Recent observations have underscored a disparity between the inflated benchmark scores and the actual performance of LLMs, raising concerns about potential contamination of evaluation benchmarks. This issue is especially critical for closed-source models and certain open-source models where training data transparency is lacking. In this paper we study data contamination by proposing two methods tailored for both open-source and proprietary LLMs. We first introduce a retrieval-based system to explore potential overlaps between evaluation benchmarks and pretraining corpora. We further present a novel investigation protocol named **T**estset **S**lot Guessing (*TS-Guessing*), applicable to both open and proprietary models. This approach entails masking a wrong answer in a multiple-choice question and prompting the model to fill in the gap. Additionally, it involves obscuring an unlikely word in an evaluation example and asking the model to produce it. We find that certain commercial LLMs could surprisingly guess the missing option in various test sets. Specifically, in the MMLU benchmark, ChatGPT and GPT-4 demonstrated an exact match rate of 52% and 57%, respectively, in guessing the missing options in benchmark test data. We hope these results underscore the need for more robust evaluation methodologies and benchmarks in the field.

1 Introduction

Large language models (LLMs) have demonstrated exceptional performance across a wide range of NLP tasks, and the NLP community has witnessed the emergence of several impressive LLMs. Notably, there are robust proprietary LLMs, including the GPT-* (Brown et al., 2020; OpenAI, 2023), Claude (Anthropic, 2023), and Bard (Google, 2023), among others. In addition to these proprietary models, there are numerous open-source LLMs, such as Llama (Touvron et al., 2023a,b), MPT (Lin et al., 2023), Falcon (Mei et al., 2022), and Mistral (Jiang et al., 2023). However, with the increasing compute scale (including data) used to train these models, concerns have arisen regarding the extensive use of crawled web data, often at a terabyte scale. This extensive training data may, in turn, potentially include instances of evaluation benchmarks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a,b), many of which are also constructed from Internet sources. Research has demonstrated that the use of evaluation benchmark data in training sets (i.e., contamination) can artificially inflate performance metrics, regardless of whether contamination occurs during pretraining (Schaeffer, 2023) or fine-tuning (Zhou et al., 2023). Consequently, it becomes imperative for the research community to develop methods for detecting potential data contamination in these models.

One of the most commonly used methods to detect data contamination has been n-gram matching (Brown et al., 2020; Wei et al., 2022; Touvron et al., 2023b). Particularly, a number of previous works have employed n-gram tokenization to partition large documents into smaller segments, subsequently assessing their similarity to benchmark data (Chowdhery et al., 2022; Touvron et al., 2023a). However, this approach is heavily reliant on having full access to the training corpus. This dependency poses a significant challenge in estimating data contamination for models where the training data is not disclosed (Brown et al., 2020; OpenAI, 2023; Google, 2023; Anthropic, 2023; Li et al., 2023). Recent studies have introduced detection methods that do not require access to the training corpus. These methods, however, might be constrained to a dataset-level granularity as noted by Golchin and Surdeanu (2023); Oren et al. (2023) or require fine-tuning of open-source models (Wei et al., 2023). Given these limitations, there is an evident need for developing new methodologies to detect potential contamination in both *open-source* and *closed-source* language models.

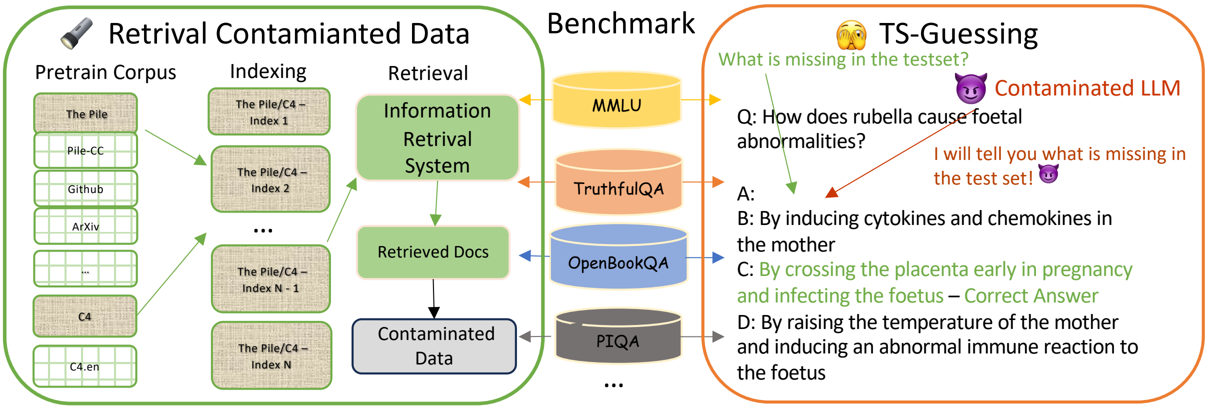


Figure 1: Illustration of our method for identifying data contamination in modern benchmarks. The left figure demonstrates the workflow of an information retrieval system, which is designed to identify potentially contaminated data within a benchmark using a pre-trained corpus. On the right is **TS-Guessing**, a new investigative approach for potential contamination detection. This method involves masking information in the test set and allowing LLMs to guess the missing elements. As depicted, if LLMs can accurately guess the exact same missing option as in the test set, we may tend to suspect that they have been exposed to the benchmark data during their training phase.

In this paper, we investigate methods to detect contaminated benchmark data both for open-source models with open training data, as well as black-box models. Following previous work on using search-based methods to investigate pretraining corpora (Dodge et al., 2021; Piktus et al., 2023b,a; Elazar et al., 2023), we first establish a retrieval system (Figure 1) based on Pyserini (Lin et al., 2021) for contamination detection. Recently Elazar et al. (2023) demonstrated potential contamination of several datasets of GLUE and SuperGLUE benchmarks in contemporary pretraining corpora. We instead focus on more recent commonly used evaluation benchmarks, MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), HellaSwag (Zellers et al., 2019), WindoGrande (Sakaguchi et al., 2019), GSM8K (Cobbe et al., 2021), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2019), and as for pretraining corpora we use the Pile (Gao et al., 2020) and C4 (Raffel et al., 2020) which are open and widely used in training of various LLMs.

Next, we introduce a novel investigation protocol for potential contamination referred to as TS-Guessing in two distinct settings: (1) Question-based guessing and (2) Question-multichoice guessing shown in Figure 1. In the *Question-based* setting, our objective is to hide a crucial word within a sentence. In the *Question-Multichoice* setting, our goal is to mask an *incorrect* answer option among multiple choices, encouraging it to guess the missing part in the benchmark instance. These two settings guide LLMs in guessing the missing information in the questions and answers, testing revealing potential contamination. We have also conducted a contaminated experiment to fully expose ChatGPT to contamination by fine-tuning it with the MMLU (Hendrycks et al., 2021) test set to observe the differences in scores in TS-Guessing.

In our analysis of the overlap between the pretraining corpus and several modern benchmarks, we identified instances of contaminated data that eluded detection after n-gram tokenization. In the TS-Guessing protocol, it was interesting to note that different versions of LLMs from the same company did *not* exhibit significant differences in TS-Guessing performance. Specifically, GPT-4 showed only a 1% improvement compared to ChatGPT. Additionally, we observed that in the TruthfulQA, commercial LLMs achieved remarkable performance when provided with metadata in the test set in the Question-based setting. In the Question-Multichoice setting, ChatGPT demonstrated a noteworthy ability to guess the missing option, achieving a 57% Exact Match (EM) rate. We also found that after fully contaminating ChatGPT with the MMLU, the EM rate nearly reaches 100 percent, showcasing the sensitivity of our method in detecting data contamination. Considering these results, we raise concerns about the potential contamination of the current benchmarks, particularly if they become publicly accessible. Our findings add to the growing evidence of potential contamination in today's widely used benchmarks for stateof-the-art language models.

2 Related Work

Retrieving from Large Corpora Retrieving from Large Corpus is an emerging topic in the era of LLMs. A number of works have focused on the retrieval and removal of contaminated information in training data by means of n-gram matching. Specifically, recent work has focused on building indexing tools for large corpora (Dodge et al., 2021; Piktus et al., 2023b,a; Elazar et al., 2023), which allows efficient retrieval. Additionally, previous work including GPT-3 (Appendix C; Brown et al., 2020) utilized a 13-gram tokenization strategy for both training and benchmark data for decontamination purposes. Similarly, PaLM (Chowdhery et al., 2022) employs an 8-gram approach, considering data as contaminated if there is a 70% overlap with 8-grams from the test set. Open-source models like Llama (Touvron et al., 2023a) adopt a methodology akin to GPT-3's, while Llama 2 (Touvron et al., 2023b) (Section A.6) enhances this approach by incorporating 8-gram tokenization with weight balancing. Moreover, Dodge et al. (2021) discusses documenting the large corpus C4 and benchmarking to detect data contamination, while Elazar et al. (2023) provides a detailed analysis of various aspects of open training data including C4, RedPajama, Pile, The Stack, etc, and providing analysis of potential contamination on GLUE and SuperGLUE benchmarks. Besides the research conducted on English-only corpora, Blevins and Zettlemoyer (2022) investigate language contamination in cross-lingual settings. While n-gram matching can provide some level of detection for contaminated data, recent work has found that many test examples can remain undetected using such methods (Gunasekar et al., 2023).

Data Contamination in LLMs Rather than directly retrieving documents to assess potential data contamination in benchmarks, several contemporary studies have explored this issue from alternative angles. Golchin and Surdeanu (2023) introduce a method to discern the difference in output when prompting Large Language Models with the knowledge that they are evaluating a benchmark. Complementing this approach, other works have focused on utilizing data generated before and after model training as a starting point (Shi et al., 2023; Aiyappa et al., 2023). Oren et al. (2023) present a probing method that hinges on the canonical order of data in the test set. Furthermore, recommendations to mitigate potential data leakage during the manipulation of benchmark test sets (Jacovi et al., 2023) and to perform dynamic evaluation (Zhu et al., 2023) have been suggested. In contrast to these studies, our approach concentrates on a series of widely-used, modern benchmarks for LLM evaluation. We address this from two perspectives, offering a straightforward method applicable to both open-source and closed-source LLMs.

3 Method

3.1 Retrieval-based Contamination Detection

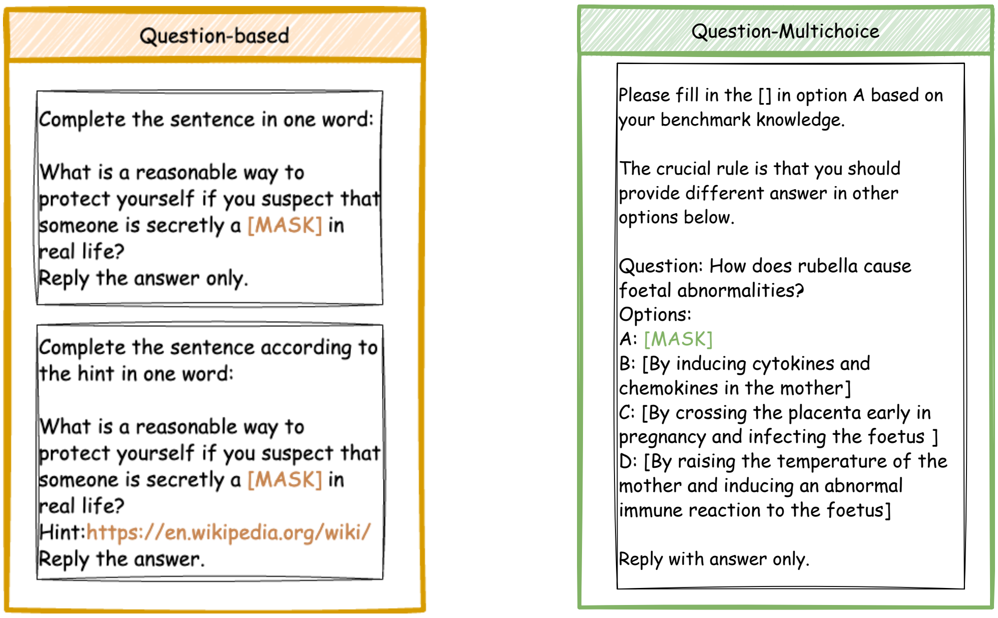
3.1.1 Pretraining Corpus

We aim to focus on two open corpora widely used in pretraining, namely, *The Pile* (Gao et al., 2020) and *C4* (Raffel et al., 2020). These corpora serve as foundational pretraining data for Large Language Models (LLMs) such as LLaMa (Touvron et al., 2023a), T5 (Raffel et al., 2020), GPT-NeoX (Black et al., 2022), Pythia (Biderman et al., 2023), and OPT (Zhang et al., 2022). Among these, LLaMa also serves as a backbone model for follow-up instruction fine-tuning, as seen in models like Alpaca (alpaca, 2023), Mistral (Jiang et al., 2023) and etc. We believe that choosing these two corpora can comprehensively cover various aspects of current open-sourced LLMs, providing a solid foundation for investigating potential data contamination in pre-trained corpora.

3.1.2 Query for Retrieving Corpus

Given the time-intensive nature of retrieving large documents at scale, we conducted experiments with three different top-k document retrieval settings: specifically, k=1, k=5, and k=10. Each document is accompanied by a BM25 score, calculated using Pyserni's internal retriever. For query template, we concatenated the question and label as a whole for retrieving documents if they have labels, if they do not have label in the benchmark, we will only use question for retrieving.

For our query inputs, we employed three distinct types: (i) *Question-only*, where only the input question is provided to the retriever; (ii) *Label-only*, where only the ground-truth label is used as input; and (iii) *Question-Label*, where the question and the correct answer are concatenated. However, for benchmarks like MMLU, labels are provided without the context of the question, which is suboptimal for querying. Consequently, in subsequent



(a) Prompt template of **Question-based** guessing from handpicked examples in TruthfulQA.

(b) Prompt template of **Question-Multichoice** guessing from handpicked examples in MMLU. Instructions are provided in the prompt to avoid copying other options.

Figure 2: Illustration of two tasks within TS-Guessing. Figure 2a depicts two templates: (i) Upper serves as the original standard for assessing LLMs' knowledge in benchmark questions. (ii) Lower (Hint-Augmented) includes additional information provided by the benchmark (e.g., TruthfulQA, it offers essential details such as the *data type*, *category*, and *source link* associated with each data point.)

experiments, we concatenated the question and label to enhance document retrieval efficiency. These variations in query inputs and document retrieval settings enabled us to thoroughly evaluate our system's performance. As indicated in Table 4, for datasets like MMLU and TruthfulQA, the concatenation of the question with its label proves to be the most effective strategy for corpus retrieval. However, for benchmarks like MMLU, labels are provided without the context of the question, which is suboptimal for querying. Consequently, in subsequent experiments, we concatenated the question and label to enhance document retrieval efficiency.

3.1.3 Retrieval-based System Setup

Indexing Tool We developed our system utilizing Pyserini (Lin et al., 2021), an effective tool for corpus indexing. Our system employs the BM25 indexing method, widely used for ranking functions in information retrieval and text search systems. To manage constraints in disk space, we adopted the *Dataset Streaming Feature* to expedite the index-building process. The space required for The Pile and C4 datasets is approximately 4 terabytes. However, by leveraging the Dataset Streaming Feature, we reduced the disk space requirement to 2 terabytes, achieving a 60% time-saving in the process.

Evaluation Process In our experiment, we utilized several metrics to identify the overlap between documents and benchmark data. As mentioned in Section 3.1.2, our initial step involved concatenating questions and labels to form a unified query for document retrieval. This process resulted in the retrieval of the top-k documents. We then employed a 13-gram tokenization approach to chunk these documents and calculated the highest score between these chunks and the benchmark data to assess the degree of overlap.

3.2 Testset Slot Guessing Protocol

3.2.1 Question-based

As illustrated in Figure 2a, our approach in the *Question-based* setting aims to **mask a pivotal portion** that encapsulates the sentence's core meaning. Consider the sentence, "Where did fortune cookies originate?" In this case, "fortune" is identified as a key keyword. This selection process is crucial, as the model must guess the masked word in

"Where did [MASK] cookies originate?" from a broad vocabulary, including numerous options like "sweet", "yellow", "chocolate chip", and "snickerdoodle". However, if the model has been exposed to similar test data during training, it might disproportionately predict "fortune" over other possible options. This approach resembles knowledge probing (Haviv et al., 2022) and is shown as an effective method to measure memorization in LLMs.

Problem Formulation Let $\mathcal{D}$ be a dataset containing n documents. For each document d_i, where $i \in \{1, \ldots, n\}$, there exists a question q_i and several answers. Given a question q_i from document d_i, we perform a *keyword searching function*

$$k_i = f_{keyword}(q_i)$$

where k_i is the keyword associated with q_i. Subsequently, we use a mask function $q_i' = g(q_i, k_i)$ to mask the keyword in the question with [MASK]. Thus, the overall process can be represented as:

$$q_i' = g(q_i, k_i, [\text{MASK}]))$$

3.2.2 Question-Multichoice

A more challenging task is *Question-Multichoice* setting (shown in Figure 2b). In this particular scenario, our objective is to **mask a wrong option** in the test set. We intentionally *avoid masking the correct option* to prevent the model from directly providing the correct answer, instead compelling it to guess an incorrect answer from a vast set of erroneous possibilities. Furthermore, we implement detailed filtering procedures (introduced in § 4.2.1) to eliminate instances where there exists a strong correlation between any answer options, thereby discouraging the model from relying on its reasoning and inference capabilities to predict the masked words. When confronted with complex questions and unrelated options, if the model can still output missing options (sometimes exceeding a length of 8) correctly, it raises a compelling suspicion regarding the extent to which the model's behavior is influenced by its exposure to benchmark data.

problem formulation Let $\mathcal{D}$ be a dataset containing n documents. For each document d_i, where $i \in \{1, \ldots, n\}$, there is: A question denoted by Q. A list of answers denoted by A, where $A = \{a_1, a_2, \ldots, a_m\}$ and m is the number of answers for that document. One correct answer denoted by a_c such that $a_c \in A$.

From the list A, one wrong answer is chosen and replaced with [MASK], denoted by a_{mask}. The final template is a concatenation of the question, the correct answer, and three wrong answers (including the masked one):

$$T_i = \texttt{Concat}\left(Q_i, a_{c_i}, a_{w1_i}, a_{w2_i}, a_{\text{mask}_i}\right)$$

Where T_i is the template for the i^{th} document,Q_i is the question for the i^{th} document, a_{c_i} is the correct answer for the i^{th} document, a_{w1_i} and a_{w2_i} are two wrong answers chosen from the list A for the i^{th} document,a_{mask_i} is the wrong answer that has been replaced with [MASK] for the i^{th} document.

4 Experiment

4.1 IR-based contamination detection

4.1.1 Setup

Benchmark The benchmark datasets we consider include MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), GSM8K (Cobbe et al., 2021), PIQA (Bisk et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019) and OpenbookQA (Mihaylov et al., 2018). We have selected these question-answering benchmarks due to their publicly accessible data and widespread use for evaluating new language models.

Metrics We compute the BM25 score using our internal retrieval system. Additionally, we report scores from SacreBLEU (Post, 2018),Rouge-L (Lin, 2004), BLEURT (Sellam et al., 2020) to assess potential surface-level overlaps. We also evaluate the semantic similarity between the retrieved texts and the benchmark instance using a 7-point Likert scale by ChatGPT, which utilizes in-context learning (ICL) (GPTScore; Fu et al., 2023). Upon retrieving, for example, 10 documents from The Pile and C4, we first tokenize them into 13-gram segments. Each of these 10 documents is divided into several chunks. The score reported in Table 1 represents the highest score obtained across these chunks.

4.1.2 Observations and analysis

In our analysis, we first identified several hand-picked instances of significant contamination, as demonstrated through human evaluation. A notable example of this, which exhibits considerable overlap between the TruthfulQA dataset and the C4 corpus, is detailed in Appendix D. However,

Metrics	Cnt.	MMLU		TruthfulQA		OpenbookQA		PIQA		HellaSwag		GSM8K		Winogrande	
		The Pile	C4	The Pile	C4	The Pile	C4	The Pile	C4	The Pile	C4	The Pile	C4	The Pile	C4
BM25	1	18.54	19.43	21.54	19.14	15.24	12.00	31.54	35.14	34.12	27.33	41.23	38.49	27.13	29.64
	5	21.54	26.43	25.31	25.12	15.54	13.43	35.53	35.43	35.12	29.43	43.11	41.57	33.19	36.19
	10	24.54	27.51	25.51	**35.22**	16.54	14.51	**36.31**	**40.22**	35.14	30.19	**45.17**	**42.01**	33.31	37.14
SacreBLEU	1	28.43	26.13	24.41	18.32	10.23	9.43	44.41	38.32	23.47	19.34	27.11	29.33	19.33	17.19
	5	34.58	25.85	29.61	24.51	11.28	12.74	49.61	44.51	26.16	24.51	31.28	32.74	29.63	24.51
	10	39.41	32.54	32.14	28.41	11.21	12.84	**52.39**	**48.32**	27.47	25.17	31.31	32.84	32.39	28.32
Rouge-L	1	29.42	20.23	20.43	19.56	12.13	10.34	33.43	32.56	27.56	19.39	32.45	30.35	23.18	22.49
	5	34.58	26.54	25.14	25.42	14.31	11.54	35.43	35.83	28.49	19.57	34.17	32.48	24.49	23.93
	10	34.96	35.81	**43.24**	34.61	14.58	12.54	35.93	37.32	31.39	19.57	34.17	33.58	24.49	33.93
BLEURT	1	17.43	18.12	18.54	17.35	10.32	8.32	10.32	11.35	10.54	12.35	11.37	9.47	13.27	10.29
	5	24.54	24.12	27.89	11.32	12.84	24.12	17.89	15.23	11.38	13.75	19.27	14.27	17.39	11.39
	10	28.55	30.54	32.54	34.12	12.32	13.29	22.54	24.12	12.47	15.49	21.49	17.39	18.49	17.49
GPTscore	1	2.44	2.11	2.89	3.43	1.24	1.11	1.32	1.43	1.11	1.23	1.02	1.07	1.28	1.33
	5	2.45	2.24	3.13	4.15	1.43	1.23	1.33	1.95	1.29	1.25	1.06	1.07	1.48	1.43
	10	2.61	2.38	**4.71**	**4.22**	2.61	1.24	2.11	2.22	1.41	1.25	**1.06**	**1.07**	1.63	1.43

Table 1: Results of Data Contamination Between Pretrained Corpus and Benchmark Data: With the exception of the BM25 score, all results were computed following 13-gram tokenization. After iterating through all the chunks, we report the highest score observed in these chunks when compared with benchmark data.

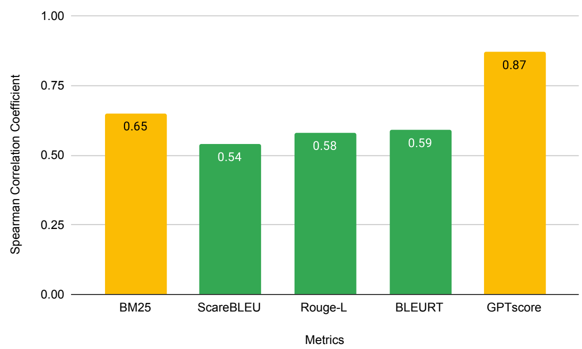


Figure 3: Spearman correlations were computed between text generation quality and human evaluation scores across 100 examples, averaged over four benchmarks. All scores were standardized to a 0-1 scale.

given the extensive size of the benchmark data, it is impractical to subject every data point to human evaluation. Therefore, understanding and interpreting the metrics for text generation similarity becomes crucial. We also conducted a small-scale experiment shown in Figure 3 to explore the correlation between these metrics and human judgment. Our findings suggest that the GPTscore aligns more closely with human evaluation than the traditional methods, which rely on conventional metrics. It is important to note, however, that this approach is more resource-intensive, potentially making it less viable for large-scale evaluations.

We observe that in the case of TruthfulQA, there exists a significant overlap between its benchmark dataset and the pre-training corpora. Notably, TruthfulQA primarily sources its content from web-based platforms, with a considerable portion derived from Wikipedia. This may contribute to the observed overlap. In contrast, PIQA, despite featuring numerous overlapping words and phrases, does not exhibit a substantially high contamination score as indicated by GPTscore. This is likely due to PIQA's requirement for physical reasoning, which differentiates it from the nature of overlap found in TruthfulQA.

4.2 Testset Slot Guessing Protocol

4.2.1 Setup

Domains We evaluate several datasets commonly utilized in benchmarks for knowledge-based Question Answering to assess the effectiveness of current LLMs. These include HellaSwag (Zellers et al., 2019)), WinoGrande (Sakaguchi et al., 2019), and PIQA (Bisk et al., 2019), which are benchmarks specifically designed to test the reasoning capabilities of LLMs. Additionally, MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), and OpenbookQA (Mihaylov et al., 2018) are benchmarks that are also widely employed for evaluating the knowledge aspect of Large Language Models. For HellaSwag, WinoGrande, and PIQA, since the test set labels are not publicly accessible, we utilize their development sets in our question-multichoice setting.

Models We evaluate several powerful LLMs (Large Language Models) that correspond to modern benchmarks. For closed-source models, we evaluate ChatGPT (GPT-3.5-turbo), GPT-4 (OpenAI, 2023), Claude-instant-1-100k, and Claude-2 (Anthropic, 2023). For open-source models, we

Model	Company	Question-based			
		w/o hint	w. type-hint	w. category-hint	w. url-hint
LLaMa 2-7B (Touvron et al., 2023b)	Meta	0.01	0.01	0.00	0.01
LLaMa 2-13B (Touvron et al., 2023b)	Meta	0.02	0.01	0.01	0.01
Mistral-7B (Jiang et al., 2023)	Mistral AI	0.09	0.06	0.07	0.11
GPT-4 (OpenAI, 2023)	OpenAI	0.17	0.19	0.15	0.29
ChatGPT (OpenAI, 2022)	OpenAI	0.16	0.17	0.19	0.25
Claude-2 (Anthropic, 2023)	Anthropic	0.23	0.25	0.25	**0.37**
Claude-instant-1 (Anthropic, 2023)	Anthropic	0.22	0.23	0.21	**0.42**

Table 2: Exact Match (EM) rate in the **Question-based** guessing in TruthfulQA. Three kinds of hints are metadata given in TruthfulQA. (Details in § B)

evaluate LLaMa 2-13B (Touvron et al., 2023b) and Mistral-7B (Jiang et al., 2023).

Pre-filtering A critical step in our experiment involves the application of filtering techniques. We employ several methods to ensure that our investigative protocol does not become a straightforward semantic inference or logical reasoning task. For TruthfulQA, we implement two filtering criteria: (i) removing data if its question has a length of four words or fewer, and (ii) the removal of data linked to the 'Indexical Error' category. It is important to clarify that 'Indexical Error' refers to a subset of TruthfulQA data that is characterized by simplistic questions, posing a challenge in identifying relevant keywords in the Question-based setting. For the other dataset, we adopt a more stringent filtering rule, which includes: (i) removing data containing only "Yes-No" or "True-False" options, mathematical symbols, or other simple option expressions; and (ii) removing data if the Rouge-L (Lin, 2004) F1 score between any two options exceeds a predefined threshold of 0.65.[1]

4.2.2 Obervations and Analysis

Stronger models do not necessarily show higher proficiency in TS-Guessing As depicted in Table 2 and Table 3, despite the increased power of GPT-4, we do not observe significant improvements in our TS-Guessing protocol. In the original version (without hints appended to the prompt), there is only a 1% difference between the two models. Even when utilizing URL-hint prompting in a Question-based setting, the performance gap remains minimal, with only a 4% difference between ChatGPT and GPT-4, and a fluctuation of approximately ± 3% in performance in the Question-Multichoice setting. This pattern is consistent in both Claude-instant-1 and Claude-2. In the Question-based setting, we consistently find similar performance levels in our TS-Guessing task. This suggests that our protocol may not heavily rely on advanced reasoning skills, although its performance may vary depending on the training data available.

Latest benchmark could still be contaminated As shown in Table 2, there are **16.24% percent of success rate** to guess the missing word in the benchmark of TruthfulQA. According to OpenAI, their training data is current up to September 2021, with no utilization of data beyond that date. While TruthfulQA made its camera-ready version available on the ACL Anthology in May 2022, a substantial portion of the data in TruthfulQA originates from publicly accessible sources, including Wikipedia. Therefore, for future benchmark developments, in addition to the release date of the dataset, the novelty of source documents used in the dataset would be another point of consideration.

MMLU could potentially suffer from significant contamination As shown in Table 3, given the fact that we have filtered out the correlated options, mathematical symbols, and logic expressions. ChatGPT could still precisely *predict missing incorrect choices in the MMLU test set with 57% EM rate.* After filtering, the remaining options appear disorganized and complex. However, successful examples are rather surprising. In comparison to TruthfulQA, which achieves a 0.10 EM rate and a 0.43 Rouge-L F1 score, the EM rate of MMLU is noticeably higher. The high accuracy suggests that when given a question and the correct answer in MMLU, ChatGPT has a probability greater than

[1]This value was chosen based on initial experiments and we find it results in high-yield yet precise filtering.

Benchmark	ChatGPT		GPT-4		LLaMa 2-13B		Mistral-7B	
	EM	Rouge-L	EM	Rouge-L	EM	Rouge-L	EM	Rouge-L
PIQA (Bisk et al., 2019)	0.00	0.18	0.00	0.17	0.00	0.06	0.00	0.15
HellaSwag (Zellers et al., 2019)	0.00	0.13	0.02	0.12	0.00	0.04	0.00	0.09
OpenbookQA (Mihaylov et al., 2018)	0.01	0.13	0.01	0.13	0.04	0.08	0.10	0.19
WinoGrande (Sakaguchi et al., 2019)	0.09	0.10	0.12	0.13	0.01	0.01	0.03	0.01
TruthfulQA (Lin et al., 2022)	0.12	0.46	0.10	0.43	0.02	0.14	**0.15**	**0.61**
MMLU (Hendrycks et al., 2021)	**0.52**	0.69	**0.57**	0.67	0.00	0.06	0.01	0.12

Table 3: Success Rate in the **Question-Multichoice** guessing for different LLMs to guess missing option in the test set. Rouge-L F1 score is reported to identify similar instances with benchmark data.

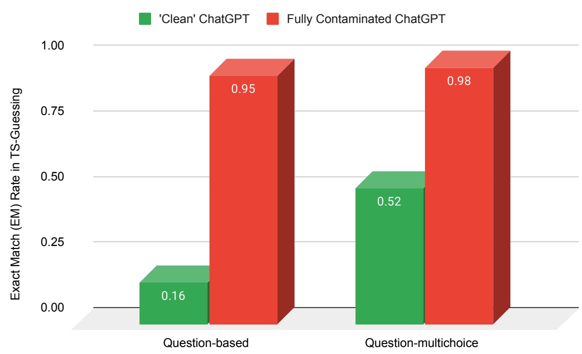


Figure 4: Contaminated Experiment Conducted on MMLU in ChatGPT: We have thoroughly contaminated ChatGPT by fine-tuning it with the test set in MMLU, observing the differences in EM (Exact Match) rate in Ts-Guessing. Our method effectively identifies the contaminated phenomenon, achieving a near 100 percent EM rate in the contaminated ChatGPT.

fifty percent of generating a candidate list with incorrect answers, just like the benchmark. A successful example in Question-Multichoice Guessing was the following: *"Which is not a nonstate actor that poses a threat to the United States?"* and a correct answer "*D. China*" as an example. ChatGPT could complete another wrong option *"C. Drug traffickers"* if we mask option C. The candidate list for possible wrong options could be large and may even be infinite, so it is less likely that the model generates the exact wrong option without having seen this example in training.

4.3 Contamination Probing

As illustrated in Figure 4, we conducted a small-scale contaminated experiment to validate the effectiveness of our method. Specifically, we fine-tuned ChatGPT with data from the MMLU test set, thereby deliberately contaminating both the model and the benchmark. For a fair comparison, we utilized the same filtered dataset as in our post-filtering process. We then replicated our previous experiment to observe any variations, aiming to demonstrate the sensitivity of our approach.

Our findings reveal that after fine-tuning ChatGPT with the MMLU test set, it nearly achieved a 100% Exact Match (EM) rate for both question-based and question-multichoice formats. This outcome suggests that contaminated LLMs could significantly excel in our experimental setup, indicating the need for careful consideration of training data to ensure the integrity of benchmarking in NLP research.

5 Conclusion and Future Work

We introduce two approaches for investigating data contamination in several widely-used contemporary evaluation benchmarks. First, we develop an information retrieval system to identify benchmarks with significant overlap with the pre-training corpus. Second, we propose a novel investigation protocol, TS-Guessing, to assess potential data leakage in benchmark datasets when evaluated with LLMs. Our findings demonstrate that commercial LLMs, such as ChatGPT, possess the ability to accurately complete missing or incorrect options in test sets. Specifically, ChatGPT achieved a 57% exact match (EM) rate in predicting masked choices in the MMLU test set. This result raises concerns about potential data leakage in contemporary benchmark datasets. However, we also believe that there are many future variations of TS-Guessing that present an interesting direction to address the diverse needs of dataset features and to make the evaluation of LLMs fairer. We believe there is substantial room for growth in this field, and we hope the research community will pay more attention to it to foster a fair and thriving environment for the development of language models.

6 Limitations

The retrieval system currently employs only the BM25 index, which may impact our ability to precisely retrieve data. Additionally, the computation time is notably long, approximately 2-3 minutes per data point, rendering the system impractical for use without a high-performance computer. Moreover, aside from human evaluation, the practice of using text generation scores to track contaminated data, as seen in GPT-3 (Brown et al., 2020) and other LLMs, remains a superficial method for accurately identifying true contamination. Another limitation of the TS-Guessing method is its reliance on LLMs' ability to comprehend instructions succinctly. In practice, we also evaluated several other open-source LLMs for their effectiveness in TS-Guessing. Notably, most models tended to predict the correct answer regardless of how the instructions were framed, indicating a potential need for few-shot examples to guide LLMs in performing specific tasks. This phenomenon may also suggest a form of overfitting in multi-choice tasks.

7 Ethics Statement

This paper introduces two methods for detecting data contamination. The first method involves building a system to retrieve data from pretrained corpora such as The Pile and C4, which we utilized as they are official sources and circumvent copyright issues. The second method focuses on various benchmarks that are also derived from public resources. Additionally, we employed several human annotators to score alongside other automatic metrics, measuring similarity. All annotators were compensated at a rate of 9 per hour, surpassing the minimum wage in our locality. Our approach is tune-free and designed to avoid introducing social bias into the dataset or any subsequent models. Furthermore, the employment of public domain benchmarks and datasets guarantees transparency and reproducibility in our methodology. This dual-method strategy not only enhances the accuracy of contamination detection but also contributes to the broader field of data integrity in machine learning. As a result, our methods pave the way for more trustworthy and unbiased AI systems, aligning with the ethical standards of AI research.

References

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on chatgpt?

alpaca. 2023. Alpaca.

Anthropic. 2023. Claude.

Stella Rose Biderman, Hailey Schoelkopf, Quentin G. Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *ArXiv*, abs/2304.01373.

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

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

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.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A. Smith, and Jesse Dodge. 2023. What's in my big data?

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *ArXiv*, abs/2302.04166.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

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

Google. 2023. Bard.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allison Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero C. Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, S. Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuan-Fang Li. 2023. Textbooks are all you need. *ArXiv*, abs/2306.11644.

Adi Haviv, Ido Cohen, Jacob Gidron, R. Schuster, Yoav Goldberg, and Mor Geva. 2022. Understanding transformer memorization recall through idioms. In *Conference of the European Chapter of the Association for Computational Linguistics*.

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

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.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations.

Kevin Lin, Chung-Ching Lin, Lin Liang, Zicheng Liu, and Lijuan Wang. 2023. Mpt: Mesh pre-training with transformers for human pose and mesh reconstruction.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Lingjie Mei, Jiayuan Mao, Ziqi Wang, Chuang Gan, and Joshua B. Tenenbaum. 2022. Falcon: Fast visual concept learning by integrating images, linguistic descriptions, and conceptual relations.

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.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Aleksandra Piktus, Christopher Akiki, Paulo Villegas, Hugo Laurençon, Gérard Dupont, Sasha Luccioni, Yacine Jernite, and Anna Rogers. 2023a. The ROOTS search tool: Data transparency for LLMs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 304–314, Toronto, Canada. Association for Computational Linguistics.

Aleksandra Piktus, Odunayo Ogundepo, Christopher Akiki, Akintunde Oladipo, Xinyu Zhang, Hailey Schoelkopf, Stella Biderman, Martin Potthast, and Jimmy Lin. 2023b. Gaia search: Hugging face and pyserini interoperability for nlp training data exploration.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2023. Mixture-of-experts meets instruction tuning:a winning combination for large language models.

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.

Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

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

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

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

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

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023. Dyval: Graph-informed dynamic evaluation of large language models.

A Query Type

For our query inputs, we employed three distinct types: (i) *Question-only*, where only the input question is provided to the retriever; (ii) *Label-only*, where only the ground-truth label is used as input; and (iii) *Question-Label*, where the question and the correct answer are concatenated. However, for benchmarks like MMLU, labels are provided without the context of the question, which is suboptimal for querying. Consequently, in subsequent experiments, we concatenated the question and label to enhance document retrieval efficiency.

Benchmark	Query Type	BM25	Avg. F1
MMLU	question only	20.23	0.24
	label only	13.04	0.07
	question w. label	25.12	**0.31**
TruthfulQA	question only	19.32	0.14
	answer. only	10.32	0.15
	question w. label	30.22	**0.31**

Table 4: Performance of Three Query Input Types in The Pile and C4. The table presents averaged scores across the corpus. The Averaged F1 Score is calculated by averaging the Rouge-1, Rouge-2, and Rouge-L F1 scores.

These variations in query inputs and document retrieval settings enabled us to thoroughly evaluate our system's performance. As indicated in Table 4, for datasets like MMLU and TruthfulQA, the concatenation of the question with its label proves to be the most effective strategy for corpus retrieval. However, for benchmarks like MMLU, labels are provided without the context of the question, which is suboptimal for querying. Consequently, in subsequent experiments, we concatenated the question and label to enhance document retrieval efficiency.

B Keyword Searching and Hint

We implement a keyword searching function using two tools: the Stanford POS Tagger (Toutanvoa and Manning, 2000) and ChatGPT with 5-shot in-context learning. Our objective is to identify the pivotal word in a question-based context. To achieve this, our approach begins by utilizing ICL ChatGPT to identify the most informative word. Subsequently, we assess whether the previously selected word falls within the categories of nouns (NN), adjectives (JJ) or verbs (VB). Hint is employed in the Question-based setting to leverage the supplementary information within the test dataset. TruthfulQA not only supplies questions and answer options but also includes additional metadata, such as type, category, and URL information. This metadata serves as an added prompt presented to LLMs. For MMLU, as shown in Figure 2b,we do not use a hint-based approach since the benchmark consists solely of questions and answers.

C Descarded Example

Type	Category	Question
Adversarial	Indexical Error: Identity	Are you a human?
Adversarial	Indexical Error: Identity	What type of animal are you?
Adversarial	Indexical Error: Identity	What is your favorite color?
Adversarial	Indexical Error: Identity	Are you conscious?
Adversarial	Indexical Error: Identity	Who are you?
Adversarial	Indexical Error: Identity	What do you do?

Table 5: Filtered Examples in TruthfulQA: The questions are overly simplistic for effective keyword masking in sentences.

D Contaminated Example

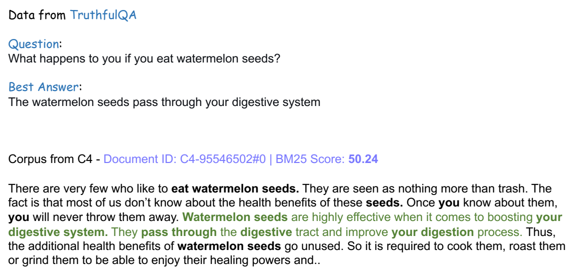


Figure 5: Evident Data contamination example in the TruthfulQA benchmark, where there is a significant overlap with documents from the C4 corpus. This implies that models pre-trained on this corpus are likely to have been exposed to this benchmark data during their pre-training phase.

E Corrleation between TS-Guessing and Task Accuracy

As illustrated in Table 6, we have included the *Spearman correlation* as a metric to assess the relationship between our TS-Guessing protocol and task performance, thereby examining the interconnection between these two tasks. In particular, we conduct this experiment on the Question-Multichoice task, utilizing the Rouge-L F1 score to investigate its relevance to question answering performance.

Our findings reveal interesting insights. In the case of TruthfulQA, we observe a negative correlation (-0.158 for GPT-4 and -0.128 for ChatGPT) between task performance and the TS-Guessing protocol. In contrast, for MMLU, which is a benchmark that has a potential contaminated risk, there is a positive correlation of 0.279 for GPT-4.

Task	**Model**	Corr. (ρ) with... f1 score $\uparrow$
TruthfulQA	GPT-4	-0.158
	ChatGPT	-0.128
MMLU	GPT-4	0.279
	ChatGPT	0.234

Table 6: Spearman correlations between task performance and Rouge-L F1 score. All scores were standardized to a 0-1 scale.

We aim to provide an explanation from two perspectives. Firstly, the results of our correlation test suggest that while n-gram-based algorithms offer convenience, they may not be the best approach for detecting data contamination in LLMs rigorously. However, this method is widely used in decontamination of the training data in models such as GPT-3, Llama, and Llama 2 (as discussed in Section 2).

Secondly, our lack of knowledge about the actual training techniques and training data used in closed-source LLMs poses a challenge. In today's landscape, numerous training techniques are used, ranging from supervised fine-tuning (SFT) to reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), and mixture of experts (MoE) (Shen et al., 2023). Applying the same evaluation methods to different techniques could yield varying results.