Unveiling the Spectrum of Data Contamination in Language Models: A Survey from Detection to Remediation

Chunyuan Deng[1,2*] **Yilun Zhao**[1*] **Yuzhao Heng**[2] **Yitong Li**[2]
Jiannan Cao[3] **Xiangru Tang**[1] **Arman Cohan**[1,4]
[1]Yale University [2]Georgia Institute of Technology [3]MIT [4]Allen Institute for AI
{cd2249,yilun.zhao,arman.cohan}@yale.edu

Abstract

Data contamination has garnered increased attention in the era of large language models (LLMs) due to the reliance on extensive internet-derived training corpora. The issue of training corpus overlap with evaluation benchmarks—referred to as contamination—has been the focus of significant recent research. This body of work aims to identify contamination, understand its impacts, and explore mitigation strategies from diverse perspectives. However, comprehensive studies that provide a clear pathway from foundational concepts to advanced insights are lacking in this nascent field. Therefore, we present the first survey in the field of data contamination. We begin by examining the effects of data contamination across various stages and forms. We then provide a detailed analysis of current contamination detection methods, categorizing them to highlight their focus, assumptions, strengths, and limitations. We also discuss mitigation strategies, offering a clear guide for future research. This survey serves as a succinct overview of the most recent advancements in data contamination research, providing a straightforward guide for the benefit of future research endeavors.

1 Introduction

Data contamination refers to the accidental or deliberate inclusion of evaluation or benchmark data in the training phase of language models, resulting in artificially high benchmark scores (Schaeffer, 2023). This issue, while longstanding—stemming from the foundational ML principle of separating training and test sets—has garnered increased attention with the advent of large language models (LLMs). These models are trained on vast corpora sourced from the web (OpenAI, 2023; Touvron et al., 2023a), heightening the risk that training data may inadvertently encompass instances from evaluation benchmarks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a,b). Such contamination of evaluation benchmarks can obscure the true generalization performance of LLMs, as it might artificially inflate benchmark scores by testing the models' ability to "memorize" and "recall" rather than "reason" or "generalize".

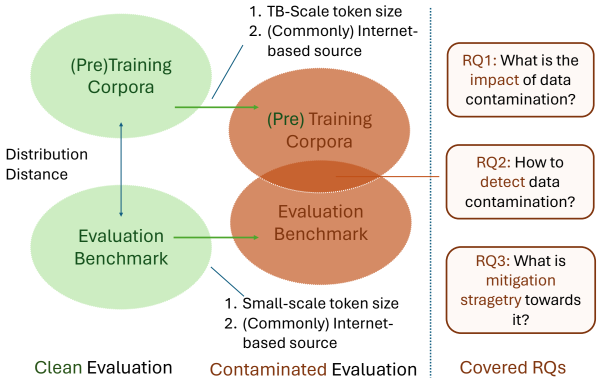
1. TB-Scale token size
2. (Commonly) Internet-based source
(Pre)Training Corpora
(Pre) Training Corpora
Distribution Distance
Evaluation Benchmark
Evaluation Benchmark
1. Small-scale token size
2. (Commonly) Internet-based source
Clean Evaluation
Contaminated Evaluation
RQ1: What is the impact of data contamination?
RQ2: How to detect data contamination?
RQ3: What is mitigation stragetry towards it?
Covered RQs

Figure 1: Basic illustration of data contamination and the research questions related to it. Clean evaluation is defined as having no overlap between the pretraining corpora and the benchmarks, and contaminated evaluation is defined as significant overlap between it.

Given the increasing concerns regarding potential contamination of evaluation benchmarks and its broader impact on downstream task performance recently, numerous studies have aimed at identifying and mitigating data contamination in these benchmarks, and understanding its impact on our perception of model capabilities. Research on data contamination could be broadly categorized into two main areas: (i) investigations of models trained with open-source data, and (ii) studies relevant to models developed using proprietary data. Generally, having access to training data, or the lack thereof, has a profound influence on modern contamination research.

In this paper, we present the very first comprehensive analysis of the growing field of data contamination detection and mitigation. Our objective is to delve into the downstream impacts of data

[*] Equal Contribution.

Findings of the Association for Computational Linguistics ACL 2024, pages 16078–16092
August 11-16, 2024 ©2024 Association for Computational Linguistics

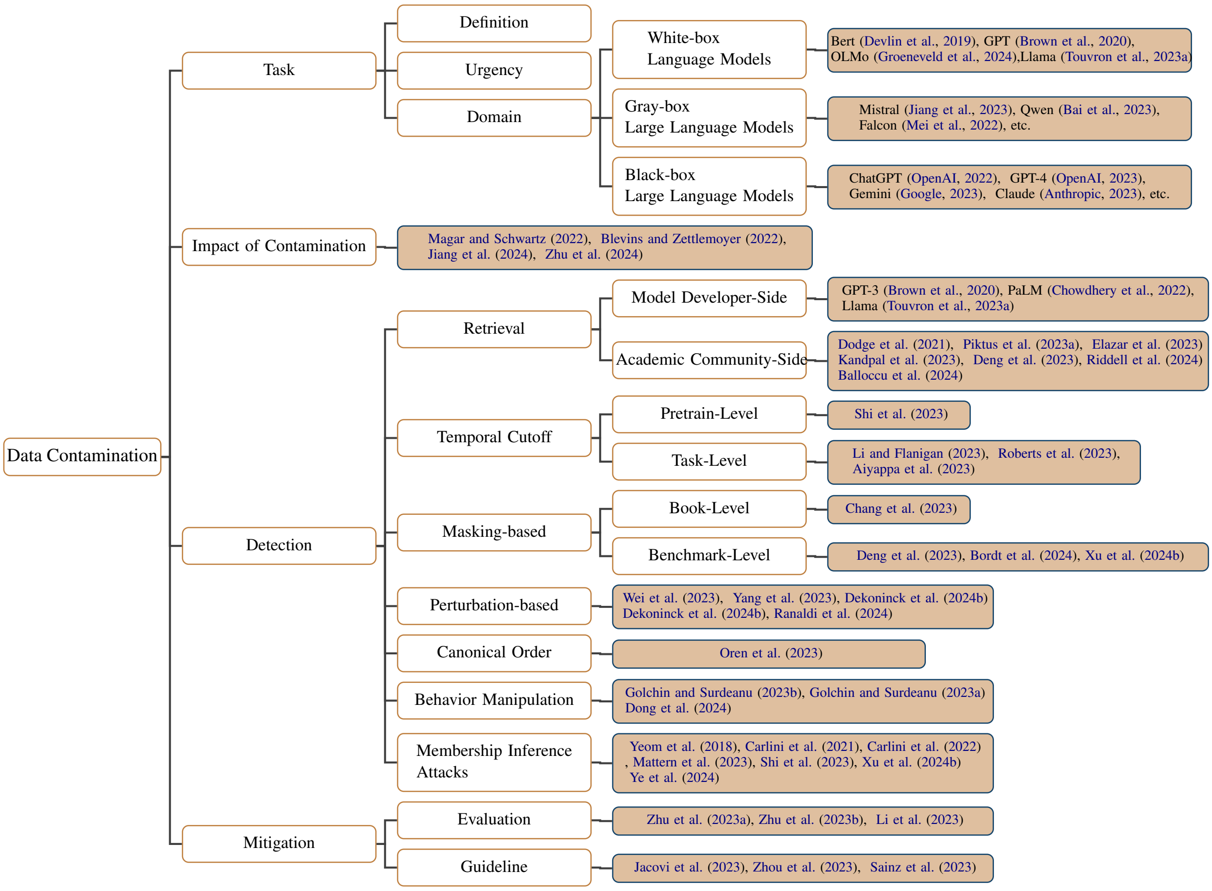
Data Contamination
Task
Definition
Urgency
Domain
White-box Language Models
Bert (Devlin et al., 2019), GPT (Brown et al., 2020), OLMo (Groeneveld et al., 2024),Llama (Touvron et al., 2023a)
Gray-box Large Language Models
Mistral (Jiang et al., 2023), Qwen (Bai et al., 2023), Falcon (Mei et al., 2022), etc.
Black-box Large Language Models
ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023), Gemini (Google, 2023), Claude (Anthropic, 2023), etc.
Impact of Contamination
Magar and Schwartz (2022), Blevins and Zettlemoyer (2022), Jiang et al. (2024), Zhu et al. (2024)
Detection
Retrieval
Model Developer-Side
GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), Llama (Touvron et al., 2023a)
Academic Community-Side
Dodge et al. (2021), Piktus et al. (2023a), Elazar et al. (2023) Kandpal et al. (2023), Deng et al. (2023), Riddell et al. (2024) Balloccu et al. (2024)
Temporal Cutoff
Pretrain-Level
Shi et al. (2023)
Task-Level
Li and Flanigan (2023), Roberts et al. (2023), Aiyappa et al. (2023)
Masking-based
Book-Level
Chang et al. (2023)
Benchmark-Level
Deng et al. (2023), Bordt et al. (2024), Xu et al. (2024b)
Perturbation-based
Wei et al. (2023), Yang et al. (2023), Dekoninck et al. (2024b) Dekoninck et al. (2024b), Ranaldi et al. (2024)
Canonical Order
Oren et al. (2023)
Behavior Manipulation
Golchin and Surdeanu (2023b), Golchin and Surdeanu (2023a) Dong et al. (2024)
Membership Inference Attacks
Yeom et al. (2018), Carlini et al. (2021), Carlini et al. (2022) , Mattern et al. (2023), Shi et al. (2023), Xu et al. (2024b) Ye et al. (2024)
Mitigation
Evaluation
Zhu et al. (2023a), Zhu et al. (2023b), Li et al. (2023)
Guideline
Jacovi et al. (2023), Zhou et al. (2023), Sainz et al. (2023)

Figure 2: Taxonomy of research on Data Contamination in large language models that consists of the task, effect, detection and mitigation.

contamination, investigate existing methods for detecting data contamination, and discuss a range of mitigation strategies. The paper is structured as outlined in Figure 1. We start by establishing the background of data contamination (§2) and discussing the effect of contamination (§3). Following this, We provide a detailed analysis of current methods for detecting data contamination (§4). We categorize these methods and critically examine the assumptions each relies on, highlighting their the prerequisites and limitation for their application. Subsequently, we explore strategies for mitigating data contamination (§5), tackling potential hurdles and proposing avenues for future investigations in this domain. Together with concurrent studies on data contamination (Ravaut et al., 2024; Xu et al., 2024a), this paper aims to furnish NLP researchers with an in-depth, systematic understanding of data contamination issues, thereby making a significant contribution to enhancing the integrity of evaluations in the field[1].

2 Background

To provide a comprehensive understanding of data contamination, this section delves into its definition, the urgency of addressing it, and its implications across different types of language models.

What is data contamination? Data contamination occurs when benchmark or test set data are inadvertently included in the training phase. This issue is particularly relevant when evaluating LLMs that have been partially trained with a test set from a benchmark, potentially leading to an inflated performance score. This phenomenon, known as data contamination, is critical for ensuring fairness and unbiased evaluation in modern LLMs.

Significance of studying contamination Thorough and complete evaluation of LLM capabilities has remained a largely unsolved problem, with benchmark contamination playing a critical role in achieving a comprehensive assessment of LLM capabilities. Contamination is a significant aspect of model evaluations. In traditional NLP and ML,

[1] The related materials at https://github.com/yale-nlp/lm-contamination-survey.

it was easy to separate training and testing data, allowing for evaluating models' generalization capabilities to new data (Suhr et al., 2020; Talmor and Berant, 2019; Lake and Baroni, 2018). However, with web-scale training data of LLMs and their enormous size in terms of number of parameters, such clear separation has become very difficult. Thus contamination of evaluation benchmarks has led to, at best, an incomplete understanding and, at worst, a misleading assessment of the true capabilities of LLMs. The risk of data contamination increases when the benchmarks for evaluating these models are derived from the same web sources used for training. This creates a potential overlap between training data and evaluation benchmarks, leading to concerns over the validity and fairness of model comparisons.

Language model types in data contamination
(1) *White-box Language Models*: The white-box language model refers to the model whose internal workings, such as the model architecture, parameters, and training data, are transparent and interpretable, allowing for a deeper understanding and analysis of its behavior. In the realm of data contamination, the focus often centers on models like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), or larger models like OLMo (Groeneveld et al., 2024), to examine the *impacts of contamination* (§3). This involves exploring the correlation between the contaminated data and downstream task performance from the perspective of how well these models memorize and are influenced by the contaminated input.
(2) *Gray-box Language Models*: The gray-box language model is a type of language model that provides some level of transparency and interpretability into its internal workings, such as revealing certain architectural components or allowing limited access to its training data, while still maintaining a degree of opacity or abstraction over other aspects of the model. This typically refer to large-scale models, such as Llama (Touvron et al., 2023a,b), Mistral (Jiang et al., 2023), Qwen (Bai et al., 2023), and Phi-3 (Abdin et al., 2024). Although the extent of openness varies among these models, they are generally characterized by their accessibility. This accessibility facilitates extensive research into their architectures and training datasets, enabling the development and validation of innovative methodologies within the field.
(3) *Black-box Language Models*: Black-box LLMs often refer to proprietary models such as ChatGPT (OpenAI, 2022), Claude (Anthropic, 2023), and Gemini (Google, 2023). The defining feature of these models is the inaccessibility of their training corpora to researchers, making it challenging to investigate data contamination. Consequently, many recent studies have focused on developing methods to address this issue (Golchin and Surdeanu, 2023b; Deng et al., 2023).

3 Impacts of contamination

The contamination effect refers to the extent to which a model, exposed to contaminated data during its training phase, is influenced by this data in its performance on downstream tasks. Research in this area typically involves selecting a base model and a fixed pretraining corpus, while varying mixture of contaminated data (Magar and Schwartz, 2022; Jiang et al., 2024). This approach allows for observing how changes in the data mix affect downstream task performance. Additionally, this area of research is often connected with evaluating the models' ability to memorize information and recall their parametric knowledge (Geva et al., 2021, 2023; Haviv et al., 2023; Srivastava et al., 2023).

3.1 Task-Level Contamination

Task-level contamination means that researchers in this field typically select a specific task, such as classification and question answering. By establishing a fixed benchmark, they vary the extent of data contamination to observe changes in performance. For example, Magar and Schwartz (2022) pretrain a BERT-based model (an *encoder-only* architecture) on a combined corpus of Wikipedia and labeled data from downstream tasks. The findings reveal that while models can memorize data during pretraining, they do not consistently utilize this memorized information in an effective manner. Additionally, the extent of exploitation is affected by several factors, including the duplication of contaminated data and the model size. Jiang et al. (2024) explore the contamination effect of the *decoder-only* architecture using GPT-2. Specifically, they pretrained GPT-2 on a selected portion of The Pile (Gao et al., 2020) corpora, intentionally introducing contaminated data during the pretraining phase to assess its impact. Their findings reveal that traditional n-gram-based methods are limited in detecting contamination, and increase the repetition of contaminated data inversely affects model performance, leading to a performance drop. Zhu et al. (2024) also investigate the relation between memorization and generation in the context of critical data size with the configuration of grokking (Power et al., 2022), a phenomenon where a model suddenly achieves near-perfect performance on a task after a period of apparent stagnation during training. The authors introduce the Data Efficiency Hypothesis, which outlines three stages of data interaction during model training: insufficiency, sufficiency, and surplus. The study observes that as models grow, they require larger datasets to reach a smooth phase transition.

3.2 Language-Level Contamination

Most research on task-level contamination is conducted in English. However, in addition to task-level contamination, Blevins and Zettlemoyer (2022) also explore language-level contamination, which refers to the issue in cross-lingual evaluation where the setting is sometimes compromised because the pre-training corpora often contain significant amounts of non-English text, such as Chinese characters. If a model is trained on these corpora and then tested on a Chinese benchmark, the setting is no longer purely cross-lingual, as the model has already been exposed to Chinese characters during training. Their research indicates that the corpora utilized for pretraining these models include a significant amount of non-English text, albeit less than 1% of the total dataset. This seemingly small percentage equates to hundreds of millions of foreign language tokens in large datasets. The study further reveals that these minor proportions of non-English data considerably enhance the models' capability for cross-language knowledge transfer. There is a direct correlation between the models' performance in target languages and the volume of training data available in those languages.

4 Detecting Data Contamination

In this section, we discuss various methods for detecting data contamination. We begin with the traditional retrieval-based method, which primarily employs n-gram tokenization and string-matching for detection. This approach is often documented in technical reports by proprietary companies. Subsequently, we introduce several modern methods predominantly developed by the academic community. These methods typically detect contamination indirectly and implicitly, without requiring full access to the training corpora.

4.1 Retrieval

One straightforward approach to detecting contamination is searching the training data for examples that appear in a benchmark. This line of research can be approached from two perspectives: the perspective of model developers and that of the academic community.

4.1.1 Model Developer-Side

In the era of LLMs, OpenAI set a significant precedent with the release of GPT-3 (Brown et al., 2020). GPT-3 pioneered a detailed approach to detecting data contamination in LLMs from an internal perspective. The methodology involved filtering the initial training set to eliminate any text from the benchmarks that appeared in the training data. This was achieved by identifying overlaps through searching for 13-gram matches between the test/development sets and the training data. Overlaps were analyzed using a variable word count, determined by the 5th percentile of example length in words, with a set minimum threshold of 8 words for non-synthetic tasks and a maximum of 13 words for all tasks.

Following this work, Llama-2 (Touvron et al., 2023b) employs a similar technique to detect data contamination, combining retrieval methods with n-gram-based tokenization. Specifically, any token n-gram match exceeding 10 tokens indicates contamination. This method facilitates a nuanced analysis of contamination levels, classifying samples as *clean* (*i.e.*, less than 20% contamination), *not clean* (*i.e.*, 20-80% contamination), and *dirty* (*i.e.*, more than 80% contamination). It uses skip-grams longer than 10 tokens and suffix arrays for efficient identification, employing parallel processing to improve speed and scalability.

4.1.2 Academic Community-Side

Beyond technical reports from model developers, many recent studies by the academic community focus on contamination in open-source pretraining corpora commonly used to develop LLMs. This body of research typically involves constructing effective and convenient tools, developing indexing systems for retrieval, and designing algorithms to determine potential contamination between retrieved passages and benchmark data.

Searching Tools To explore different pretrained corpora, various specialized tools have been de-

Method	**Level**	**Access to Training Corpora Required?**	**Logits Prob. Required?**	**Retrieval?**	**Prompt-based?**
Brown et al. (2020)	Instance	✓	✗	✓	✗
Chowdhery et al. (2022)	Instance	✓	✗	✓	✗
Touvron et al. (2023a)	Instance	✓	✗	✓	✗
Yeom et al. (2018)	Instance	✗	✓	✗	✗
Carlini et al. (2021)	Instance	✗	✓	✗	✗
Dodge et al. (2021)	Instance	✓	✗	✓	✗
Carlini et al. (2022)	Instance	✗	✓	✗	✗
Elazar et al. (2023)	Instance	✓	✗	✓	✗
Li (2023)	Dataset	✗	✓	✗	✗
Shi et al. (2023)	Dataset	✗	✓	✗	✗
Aiyappa et al. (2023)	Instance	✗	✗	✗	✗
Roberts et al. (2023)	Instance	✗	✗	✗	✗
Golchin and Surdeanu (2023a)	Dataset	✗	✗	✗	✓
Golchin and Surdeanu (2023b)	Both	✗	✗	✗	✓
Oren et al. (2023)	Dataset	✗	✓	✗	✗
Deng et al. (2023)	Instance	✗	✗	✗	✓
Bordt et al. (2024)	Instance	✗	✗	✗	✓
Wei et al. (2023)	Instance	✗	✗	✗	✗
Mattern et al. (2023)	Instance	✗	✓	✗	✗
Xu et al. (2024b)	Instance	✗	✗	✗	✓

Table 1: Comparison of current data contamination detection method.

veloped. Piktus et al. (2023a) introduce a search engine that spans the entirety of the ROOTS corpus (Laurençon et al., 2023), featuring both fuzzy and exact search capabilities. Furthermore, Piktus et al. (2023b) present Gaia, a search engine designed based on established principles, providing access to four widely recognized large-scale text collections: C4 (Raffel et al., 2023), The Pile (Gao et al., 2020), LAION (Schuhmann et al., 2022), and ROOTS (Laurençon et al., 2023). Additionally, Elazar et al. (2023) develop WIMBD, a platform offering 16 analytical tools that enable users to uncover and contrast the contents of vast text corpora.

Indexing System The primary limitation of search tools is their dependency on extensive computational resources, combined with the absence of APIs for scalable integration. For individuals endeavoring to develop a custom information retrieval system, Lin et al. (2021) introduce Pyserini, a user-friendly Python-based toolkit designed for replicable information retrieval (IR) research. Pyserini facilitates various retrieval methods, including sparse retrieval using BM25 with bag-of-words representations, dense retrieval via nearest-neighbor search in transformer-encoded spaces, and a hybrid approach that combines both methods. Researchers also have used such indexing tools to investigate data contamination (Deng et al., 2023) for investigating contamination in commonly used pretraining corpora such as The Pile and C4.

Benchmarks Overlap Analysis In their pioneering work, Dodge et al. (2021) conduct the first comprehensive analysis of data contamination between the C4 corpus (Raffel et al., 2023) and downstream tasks. This study uncovers a significant volume of text from unexpected sources, including patents and US military websites. Further scrutiny reveals the presence of machine-generated content, such as text from machine translation systems, and evaluation examples from various NLP datasets. Building on this, Elazar et al. (2023) present an analysis that explores the overlap between pretraining corpora and the SuperGLUE (Sarlin et al., 2020) benchmark.

4.2 Temporal Cutoff

The concept of time-cutoff implies a significant distinction between models developed or the use of training data up to a certain time point. For instance, GPT-3 was trained using data available only up to September 2021 (OpenAI, 2022). This approach assumes that substantial changes in the dataset's distributions or variances, stemming from a specific time cut-off, are critically important.

Roberts et al. (2023) conduct one of the first comprehensive longitudinal analysis of data contamination in LLMs. Specifically, they leverage the natural experiment provided by the training cutoffs in GPT models to examine benchmarks released over time. They analyze two code/mathematical problem-solving datasets. Their findings reveal statistically significant trends between LLM pass rates, GitHub popularity, and release dates, which strongly indicate contamination. Aiyappa et al. (2023) also conduct similar experiments to assess performance difference in models before and after their release. Besides, Shi et al. (2023) create a benchmark termed WIKIMIA utilizing data compiled both before and after model training to facilitate accurate detection. Similarly, Li et al. (2023) employ the most recent data to develop a benchmark that is less prone to contamination, enabling a fair evaluation.

The time-cutoff technique requires verification that data before and after a specific time-cutoff exhibit distinct distributions with minimal overlap. Additionally, new events or messages extracted from the internet may also overlap with previous ones. For employing a time-cutoff strategy, it is essential to account for and evaluate these potential overlaps in experimental setups.

4.3 Masking-based

Another approach to detecting data contamination involves masking-based methods, which masks a word or sentence and provides the LLMs with context from a benchmark to guide them in filling in the missing portions. The advantage of this approach is its simplicity and effectiveness.

Book-Level Chang et al. (2023) propose the *name cloze* task, wherein names within a book are masked, prompting LLMs to predict the omitted names. This task is specifically designed to evaluate the extent to which models like ChatGPT and GPT-4 have internalized copyrighted content, linking memorization levels to the prevalence of book excerpts online. The findings reveal a notable performance disparity between GPT-4 and ChatGPT in executing the name cloze task, suggesting variations in their capacity to recall and utilize memorized information.

Benchmark-level Deng et al. (2023) introduce TS-Guessing, a masking-based method designed for benchmark formats to detect data contamination. This technique involves masking an incorrect answer in a multiple-choice question and prompting the model to complete the missing information. It also entails hiding an unlikely word in an evaluation example and requesting the model to generate it. Their findings reveal that several proprietary LLMs can precisely recall the masked incorrect choice in the benchmarks, highlighting a significant potential for contamination in these benchmarks that warrants attention. However, they note that their method depends on the proficient instruction-following capabilities of LLMs. For less capable LLMs, there is a tendency to replicate other choices or produce the correct answer without adhering to the given instructions.

Part of Xu et al. (2024b) also employs similar methods. Given a sequence, they progressively move forward from the first token and guide LLMs to predict the missing portions of the following part. Their method could be treated as a more quantitative version of Deng et al. (2023), which calculates the results primarily on open-sourced LLMs.

4.4 Perturbation-based

Perturbation-based methods involve using various techniques to artificially modify or alter test set samples. This is done to assess if LLMs are overfitting to particular benchmark formats or examples. The objective of this task is to examine whether there is a significant drop or change in performance after applying specific perturbations.

Rephrasing Test Set Yang et al. (2023) demonstrate that applying minor alterations to test data, such as rephrasing or translating, can bypass previous n-gram-based detection methods (§4.1.1). They reveal that if test data variability isn't eliminated, a 13B model can mimic the performance of state-of-the-art models like GPT-4 by overfitting to benchmarks, as evidenced by their experiments with notable datasets including MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021). To address this growing issue, they propose a new LLM-based detection approach, which uncovers significant, yet previously unnoticed overlaps in test sets across widely used pretraining and fine-tuning corpora. In a recent paper, Dekoninck et al. (2024b) propose ConStat, a novel method for detecting and quantifying contamination in LLMs. The authors redefine contamination from a performance-based perspective, considering it as artificially inflated benchmark performance that fails to generalize to real-world tasks. ConStat employs a statistical test that compares a model's performance on the original benchmark to its performance on carefully selected reference benchmarks, while accounting for differences in difficulty using a set of uncontaminated reference models.

Creating Reference Set In addition to directly rephrasing test set examples, Wei et al. (2023) use GPT-4 to create a reference set resembling the test set. They then calculate the difference between reference set and test set to assess the contamination issues, potentially caused by intentional data contamination. Higher differences indicate a greater potential for data leakage.

4.5 Canonical order

The canonical assumption posits that if a model has been exposed to data from a dataset, it will exhibit a preference for the canonical order provided by the dataset from public repositories, as opposed to datasets that have been randomly shuffled.

Oren et al. (2023) develop a sensitivity test to detect biases in the canonical order of benchmark datasets used for LLMs. Based on the principle that, in the absence of data contamination, any permutation of an exchangeable benchmark dataset should be equally likely, they create a methodology capable of identifying contamination through the model's preference for specific data orderings. Remarkably, this approach is sophisticated enough to detect contamination in models with as few as 1.4 billion parameters, utilizing test sets of merely 1,000 examples. It proves effective even in datasets with minimal representation in the training corpus.

The limitation of this assumption is that if the model preprocesses the pretraining dataset or intentionally shuffles the benchmark data, it becomes challenging to identify potential contamination from the perspective of canonical order.

4.6 Behavior Manipulation

We term behavior observation as a new perspective that leverages different perspectives of controlling experiment related to the test set. This is done by observing whether the behavior (*i.e.*, output and selection choice) are different.

Golchin and Surdeanu (2023b) propose a dual-layered approach for identifying contamination in LLMs at both the instance and partition levels. The initial phase employs *guided instruction*, a technique that utilizes a specific prompt incorporating the dataset name, partition type, and an initial segment of a reference instance. This prompt encourages the LLM to generate a completion. An instance is considered contaminated if the LLMs' output closely resembles or exactly matches the subsequent segment of the reference. Building on this concept, Golchin and Surdeanu (2023a) introduce a novel methodology by devising a data contamination quiz. This quiz presents a set of choices, including one from the test set and others that are variations of the original instance. The model is then tasked with selecting an option, and its decision is used to assess contamination based on its choice. This approach not only follows the general pattern of contamination detection but also offers a unique perspective by varying the format of the choices provided to the model.

Besides, Dong et al. (2024) propose CDD (Contamination Detection via output Distribution) for detecting data contamination and TED (Trustworthy Evaluation via output Distribution) for mitigating its impact on evaluation. CDD identifies contamination by analyzing the peakedness of the LLM's output distribution using only the sampled texts, while TED corrects the output distribution to ensure trustworthy evaluation.

To employ methods based on this assumption, researchers must verify that behavior differences are solely attributable to data contamination, particularly in contrast to variations arising from random prompt perturbation.

4.7 Membership Inference Attacks

Membership Inference Attacks (MIA) aim to determine whether a specific data point was used in the training data of a target model. While MIA is a well-established concept in traditional machine learning (Shokri et al., 2017; Hu et al., 2022), their application in the context of LLMs has been relatively understudied. This subsection explores the application of MIA to LLMs, demonstrating their utility in detecting contamination.

Background Yeom et al. (2018) measure the perplexity of a sample to measure the memorization of training data. Carlini et al. (2021) build upon this work to further improve precision and reduce the false negative rate by considering the intrinsic complexity of the target point. Furthermore, Carlini et al. (2022) calibrate the sample's loss under the target model using the sample's zlib compression size.

Applying MIA to LLMs Mattern et al. (2023) introduce and assess neighbourhood attacks as a novel method to evaluate model vulnerabilities without requiring access to the training data distribution. They use an estimate of the curvature of the loss function at a given sample, which is computed by perturbing the target sequence to create n neighboring points, and comparing the loss of the target x, with its neighbors. By comparing model scores of a given sample with those of synthetically generated neighbour texts, this approach seeks to understand if model fragility can enhance security.

Recently, Shi et al. (2023) introduce MIN-K%, a method that utilizes the $k\%$ of tokens with the lowest likelihoods to compute a score, rather than averaging over all token probabilities as in traditional loss calculations. This approach is based on the hypothesis that an unseen example is likely to contain a few outlier words with low probabilities under LLMs, whereas a seen example is less likely to feature words with such low probabilities.

Additionally, (Ye et al., 2024) propose Polarized Augment Calibration (PAC), a novel approach for detecting training data contamination in black-box LLMs. PAC extends the MIA framework by leveraging confidence discrepancies across spatial data distributions and considering both distant and proximal probability regions to refine confidence metrics.

MIA in the context of LLMs is typically based on perplexity or variations derived from language model perplexity. This implies reliance on the output logits probability from the language models. However, its statistical simplicity also offers significant advantages compared to other detection methods that require careful validation of assumption.

5 Mitigating Data Contamination

Without specific mitigation strategies, the development of new benchmarks—often released publicly on the internet—does not resolve contamination issues, as newer models can access this data. Consequently, several studies have proposed mitigation approaches to address this problem. In this section, we will introduce these strategies from the perspectives of benchmark construction, updating, encryption, and protection.

Benchmark Construct Selection Li et al. (2023) propose to construct evaluation benchmarks from the most recent texts, thus minimizing the risk of overlap with the pre-training corpora.

Dynamic Benchmark Refreshing Zhu et al. (2023a) introduce a dynamic evaluation protocol that utilizes directed acyclic graphs to generate evaluation samples of varying complexities, aiming to address the static and potentially contaminated nature of existing benchmarks. Besides, Zhu et al. (2023b) provide Clean-Eval, which utilizes LLMs to paraphrase and back-translate contaminated data, creating a set of expressions that convey the same meaning in varied forms. This process generates a candidate set from which low-quality samples are filtered out using a semantic detector. The final selection of the best candidate from this refined set is based on the BLEURT (Sellam et al., 2020) score, ensuring the chosen expression is semantically similar to the original data but articulated differently. Furthermore, Zhou et al. (2023) also suggest providing a diverse set of prompts for testing, which offers a dynamic evaluation to mitigate data contamination.

Benchmark Data Encryption Jacovi et al. (2023) suggests that test data released to the public should be safeguarded through encryption using a public key, and the distribution of derivatives should be strictly prohibited by the licensing agreement. To implement this, the recommended approach is toencrypt the test data before uploading it. This can be efficiently done by compressing the data into a password-secured archive.

Benchmark Label Protection Jacovi et al. (2023) and Zhou et al. (2023) emphasize the critical need to safeguard the ground truth labels of test datasets. These labels can inadvertently be exploited during the training phase, or even intentionally after being rephrased. Providing both the question and its context is an effective strategy to prevent such deliberate contamination.

6 Discussion and Future Directions

Besides addressing the impact, detection, and mitigation of previously introduced data contamination, this section will also explore the topic at a higher level. We aim to offer more insights into the current challenges, the necessity, and the robustness of detecting data contamination methods. We will also discuss how these concepts can be applied in more realistic settings. Additionally, we will consider data contamination as an overarching research direction and explore potential future pathways for this field.

Challenges for Detecting Black-Box Models The primary challenge in evaluating different methods for detecting data contamination in large language models is the absence of a ground truth label, *i.e.*, a benchmark dataset comprising entirely contaminated data. This absence creates difficulties in comparing the effectiveness of various detection techniques designed for black-box models. One alternative approach involves fine-tuning the model using test set labels to create artificially contaminated data. However, the question remains whether the scenarios of contamination during the pretraining phase and the fine-tuning phase are consistent. Additionally, due to limited access to the complete training corpus, we can only generate fully contaminated data, making it challenging to obtain fully uncontaminated data. This situation complicates efforts to accurately assess and compare the efficacy of contamination detection methods.

Dodging Detection of Data Contamination is Easy Dekoninck et al. (2024a) highlights the ease with which MIA detection methods can be evaded. These methods, some of which are also employed for identifying data contamination, have been criticized in prior research. Notably, the efficacy of n-gram based substring detection is questioned due to its numerous vulnerabilities and susceptibility to manipulation (Zhou et al., 2023; Deng et al., 2023; Jiang et al., 2024). Beyond the traditional n-gram and MIA approaches, recent studies have demonstrated that several contemporary techniques can be compromised through targeted attacks. For instance, by integrating a dataset with a significantly large pre-trained dataset, one can disrupt the canonical order assumption, thereby undermining its integrity.

From Memorization to Exploitation Drawing a definitive conclusion about the correlation between memorization and exploitation (*i.e.*, performance on downstream tasks) remains challenging. Various factors can impact the outcomes observed in our study, including differences in model architecture, the repetition of contaminated data, the strategies employed during pretraining or finetuning phases, and the training principles used like RLHF+PPO (Zheng et al., 2023) and DPO (Rafailov et al., 2023). These elements can significantly influence the models' downstream task performance.

Detecting or Mitigating? Currently, there is an increasing focus on developing novel methods for detecting data contamination, which is crucial for investigating and understanding data contamination scenarios. Effective detection tools can also help prevent intentional data contamination to a certain extent. However, there remains a significant need for research focused on mitigating data contamination. The research question arises: how can we create a dynamic evaluation method that uses potentially contaminated benchmarks to provide clean evaluations? In recent developments, many have started leveraging language models as agents to perform various tasks. An intriguing future direction could be to utilize LLMs as 'Benchmark Agents' to offer various forms of evaluation that convey the same meaning.

How to Create Benchmarks without Data Contamination To address the challenge of creating a benchmark free from data contamination, it is essential to consider innovative approaches. Firstly, an effective strategy involves constructing a dataset significantly larger than the target size. This excess allows for the application of rigorous data contamination checks to refine the dataset down to its actual size. Additionally, the implementation of a unified, reliable, and dynamic evaluation framework is crucial. Such a framework offers the flexibility to adaptively assess benchmarks across various formats, enhancing the robustness of the evaluation process. Beyond these broader strategies, a practical yet profound method involves generating content that is rare or virtually nonexistent on the Internet or other public domains.

7 Conclusion

In this paper, we present an extensive and meticulously organized survey on the topic of data contamination in large language models. We start by laying the groundwork with a discussion on the effect of contamination, setting the stage for a deeper examination of various data contamination detection methods. We critically analyze the assumptions underlying these methods, highlighting their limitations and the prerequisites for their application. Subsequently, we explore strategies for mitigating data contamination, addressing potential challenges and suggesting directions for future research in this area. Our goal is to provide a comprehensive guide for NLP researchers seeking a systematic understanding of data contamination. We also aim to underscore the critical importance of this field, advocating for increased attention due to its pressing relevance.

8 Limitations

It is challenging to provide a quantitative comparison between different data contamination detection methods due to their varying assumptions and requirements. Ideally, we would conduct a quantitative analysis to assess the effectiveness of these methods, assigning rankings or benchmarks to discuss their advantages and disadvantages. Another limitation of the survey paper is the difficulty in categorizing each method into a single, definitive class. For instance, Shi et al. (2023) not only offers benchmarks and analyses but also proposes a detection method. Similarly, Zhou et al. (2023) discusses both the detection of contamination and strategies for its mitigation. Our approach primarily classifies each work into its most evident category, aiming for clarity and precision, though this may sometimes compromise rigor.

9 Ethics Statement

In our survey paper, which examines the impact of data contamination, alongside methods for its detection and mitigation, we assert that our work not only adheres to ethical standards and avoids potential misuse issues, but also offers a comprehensive summary that contributes to the fair and transparent evaluation of large language models. This positions it as a valuable resource for promoting fairness and transparency within the community.

10 Acknowledgement

This project was supported in part by Tata Sons Private Limited, Tata Consultancy Services Limited, and Titan. We extend our sincere thanks to Simeng Han, Yixin Liu and Hanjie Chen for their invaluable proofreading, insightful discussions and steadfast support.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

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

Anthropic. 2023. Claude.

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.

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

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of english pretrained models. In *Conference on Empirical Methods in Natural Language Processing*.

Sebastian Bordt, Harsha Nori, and Rich Caruana. 2024. Elephants never forget: Testing language models for memorization of tabular data.

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

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models.

Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4.

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.

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.

Jasper Dekoninck, Mark Niklas Müller, Maximilian Baader, Marc Fischer, and Martin Vechev. 2024a. Evading data contamination detection for language models is (too) easy.

Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. 2024b. Constat: Performance-based contamination detection in large language models.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models.

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?

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.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories.

Shahriar Golchin and Mihai Surdeanu. 2023a. Data contamination quiz: A tool to detect and estimate contamination in large language models. *ArXiv*, abs/2311.06233.

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

Google. 2023. Gemini: A family of highly capable multimodal models. *ArXiv*, abs/2312.11805.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms.

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

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.*, 54(11s).

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. In *Conference on Empirical Methods in Natural Language Processing*.

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.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Investigating data contamination for pretraining language models.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. 2023. The bigscience roots corpus: A 1.6tb composite multilingual dataset.

Changmao Li and Jeffrey Flanigan. 2023. Task contamination: Language models may not be few-shot anymore.

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

Yucheng Li, Frank Guerin, and Chenghua Lin. 2023. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction.

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.

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

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison.

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.

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.

Aleksandra Piktus, Christopher Akiki, Paulo Villegas, Hugo Laurençon, Gérard Dupont, Alexandra Sasha Luccioni, Yacine Jernite, and Anna Rogers. 2023a. The roots search tool: Data transparency for llms.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 588–598, Toronto, Canada. Association for Computational Linguistics.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

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

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

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

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

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Conference on Empirical Methods in Natural Language Processing*.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models.

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

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.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hageman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. 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.

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, Xue Gang 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. *ArXiv*, abs/2310.19341.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024a. Benchmark data contamination of large language models: A survey.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024b. Benchmarking benchmark leakage in large language models.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples.

Wentao Ye, Jiaqi Hu, Liyao Li, Haobo Wang, Gang Chen, and Junbo Zhao. 2024. Data contamination calibration for black-box llms.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of rlhf in large language models part i: Ppo.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Jinhui Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *ArXiv*, abs/2311.01964.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023a. Dyval: Graph-informed dynamic evaluation of large language models. *ArXiv*, abs/2309.17167.

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

Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. 2024. Critical data size of language models from a grokking perspective.