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

Anonymous ACL submission

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 encompoass instances from evaluation benchmarks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a,b). Such contamination can obscure the true performance of LLMs, as it might artificially inflate benchmark scores by teaching models to "memorize" rather than "reason". Moreover, a fundamental objective in machine learning is to develop models that generalize well to unseen scenarios. Data contamination, however, may lead to models that favor memorization over generalization, rendering benchmarks less effective in measuring true generalization abilities.

The earliest work on data contamination in LLMs was published by the commercial company for GPT-3 (Brown et al., 2020), and was subsequently followed by research on PaLM (Chowdhery et al., 2022) and LLaMA (Touvron et al., 2023a,b). These studies employed n-gram based substring detection methods as the foundational approach for detecting data contamination. However, such methods necessitate full access to the pretraining corpora. As the proliferation of LLMs continues, many models—ranging from closed-source platforms to open-source projects that only make their weights available—lack such transparency. This opacity presents a significant challenge to the NLP community in terms of fairly evaluating and comparing LLMs, especially when the extent of data contamination and its impact on these models remain undisclosed (Sainz et al., 2023).

To address this issue, several methods have been proposed to detect data contamination without accessing training corpora (Golchin and Surdeanu, 2023a,b; Oren et al., 2023; Shi et al., 2023; Deng et al., 2023; Bordt et al., 2023). These methods provide different perspectives, from canonical (Oren et al., 2023) and behavioral observation (Golchin and Surdeanu, 2023a,b) to masking (Deng et al., 2023; Bordt et al., 2023) and membership inference attacks (Shi et al., 2023). However, these emerging methods resemble isolated stars on the plateau, lacking a detailed and well-structured discussion of their advantages and disadvantages in the literature.

In this paper, we present a comprehensive anal-

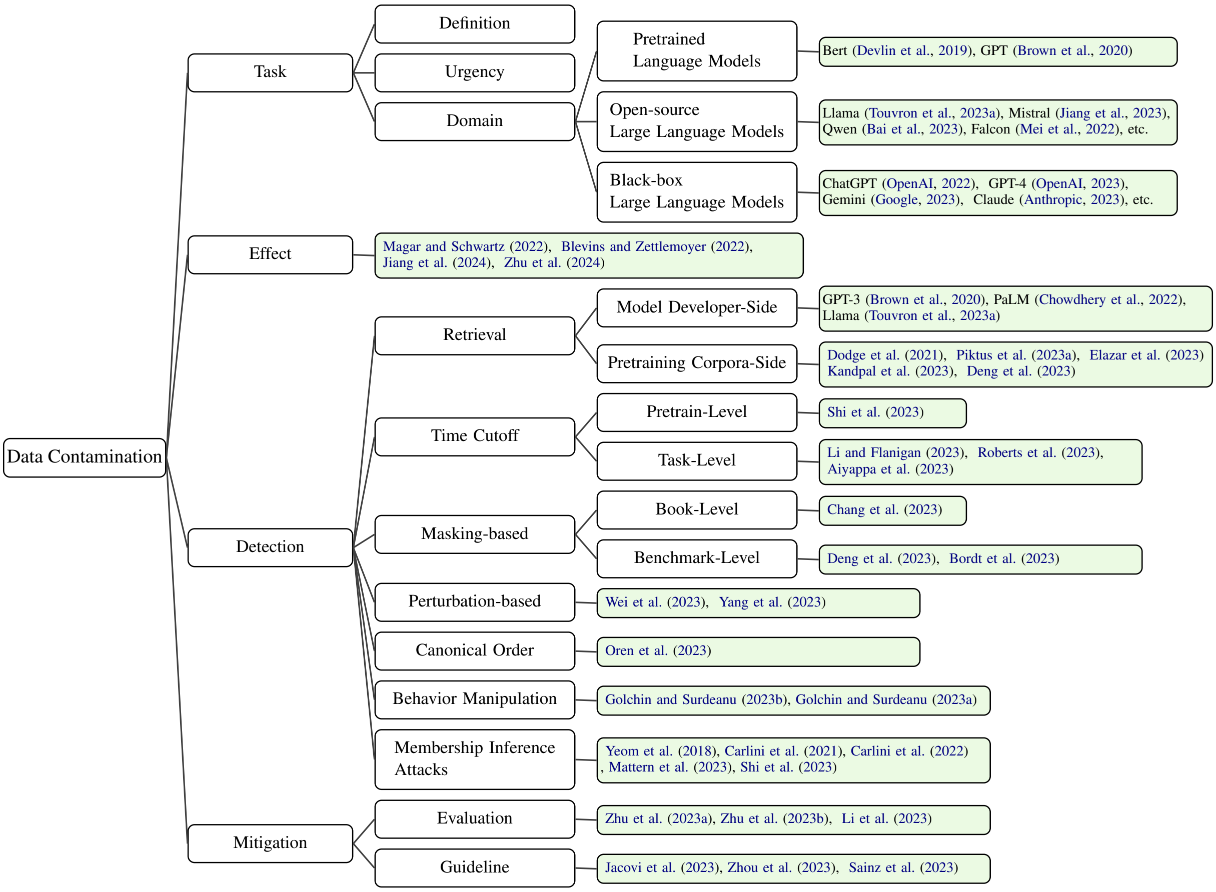
Data Contamination
Task
Definition
Urgency
Domain
Pretrained Language Models
Bert (Devlin et al., 2019), GPT (Brown et al., 2020)
Open-source Large Language Models
Llama (Touvron et al., 2023a), Mistral (Jiang et al., 2023), Qwen (Bai et al., 2023), Falcon (Mei et al., 2022), etc.
Black-box Large Language Models
ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023), Gemini (Google, 2023), Claude (Anthropic, 2023), etc.
Effect
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)
Pretraining Corpora-Side
Dodge et al. (2021), Piktus et al. (2023a), Elazar et al. (2023) Kandpal et al. (2023), Deng et al. (2023)
Time 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. (2023)
Perturbation-based
Wei et al. (2023), Yang et al. (2023)
Canonical Order
Oren et al. (2023)
Behavior Manipulation
Golchin and Surdeanu (2023b), Golchin and Surdeanu (2023a)
Membership Inference Attacks
Yeom et al. (2018), Carlini et al. (2021), Carlini et al. (2022) , Mattern et al. (2023), Shi et al. (2023)
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 1: Taxonomy of research on Data Contamination in large language models that consists of the task, effect, detection and mitigation.

ysis of the growing field of data contamination detection and mitigation[1]. Our objective is to delve into the downstream impacts of data 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 limitations and the prerequisites for their application. Subsequently, we explore strategies for mitigating data contamination (§5), tackling potential hurdles and proposing avenues for future investigations in this domain. The overarching aim of this paper is 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 and offering a valuable resource for the community.

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 included in the training phase, leading to potentially inflated benchmark scores. This issue is closely related to *data memorization*, where models may inadvertently "learn" specific data points rather than generalizing from them. Research in the field often explores the connection between data contamination and its impact on downstream task performance, drawing on memorization techniques to assess this relationship (Magar and Schwartz, 2022). Furthermore, several recent techniques for detect-

[1] The related open-source materials are available at https://anonymous.4open.science/r/data-contamination-survey-3089. We will consistently maintain and update them upon publication.

ing data contamination often indirectly evaluate whether language models are memorizing benchmark data (Chang et al., 2023; Deng et al., 2023).

Why this task is urgent? Data contamination is a critical and pressing issue, particularly in the current landscape of LLMs that predominantly utilize extensive web corpora for pretraining (Chowdhery et al., 2022; Touvron et al., 2023b). 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. Moreover, data contamination undermines the trustworthiness of benchmarks to accurately measure a model's generalization capabilities to unseen scenarios, as it blurs the line between genuine learned patterns and simple memorization of test data.

Language model types in data contamination
(1) *Pretrained Language Models*: In the realm of pre-trained language models and data contamination, the focus often centers on models like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), examining the *contamination effect* (§3). This involves exploring the correlation between the contaminated data and downstream task performance from the perspective of how well these models remember and are influenced by the contaminated input.

(2) *Open-source Large Language Models*: As previously mentioned, open-source LLMs typically refer to large-scale models like LLaMA (Touvron et al., 2023a), Mistral (Jiang et al., 2023), and Falcon (Mei et al., 2022), as well as newly emerging models such as Qwen (Bai et al., 2023) and OLMo (Groeneveld et al., 2024). These models are characterized by their accessibility, allowing for extensive research into their architectures and training datasets to develop and validate new methodologies within the field.

(3) *Black-box Large 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, as detecting data contamination in black-box models poses a hard task, many recent studies have focused on developing methods to address this issue (Golchin and Surdeanu, 2023b; Deng et al., 2023).

3 Effect

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. This approach allows for observing how changes in the data mix affect downstream task performance.

Task-Level Contamination Magar and Schwartz (2022) pretrained the 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 in 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 decline. Zhu et al. (2024) also investigate the relation between memorization and generation in the context of critical data size with the configure of grokking. 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 phase transition smoothly.

Language-Level Contamination In addition to task-level contamination, Blevins and Zettlemoyer (2022) also explore language-level contamination, a topic that has not been extensively examined. 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

4.1 Retrieval

Retrieval-based detection is the most straightforward method for identifying data contamination issues in pretraining datasets. This research can be approached from two perspectives: the model developers' side, and the pretraining corpora side.

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 Pretraining Corpora-Side

Other than technical reports from the model developer, several other research studies focus on contamination in open-source pretraining corpora commonly used for developing LLMs.

Searching Tool To explore different pretrained corpora, various specialized tools have been developed. Piktus et al. (2023a) introduce the ROOTS (Laurençon et al., 2023) Search Tool, a search engine that spans the entirety of the ROOTS corpus, 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) describe WIMBD, a platform offering 16 analytical tools that enable users to uncover and contrast the contents of vast text corpora.

Indexing System The primary constraint 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. (2021b) 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.

Benchmarks Overlap Analysis In their pioneering work, Dodge et al. (2021) conducted the first comprehensive analysis of data contamination between the C4 corpus (Raffel et al., 2023) and downstream tasks. This study uncovered a significant volume of text from unexpected sources, including patents and US military websites. Further scrutiny revealed the presence of machine-generated content, such as text from machine translation systems, and evaluation examples from various benchmark NLP datasets. Building on this, Elazar et al. (2023) presented an analysis that explores the overlap between pretraining corpora and the SuperGLUE (Sarlin et al., 2020) benchmark. Additionally, Deng et al. (2023) employed Pyserini (Lin et al., 2021a) to develop an IR system aimed at investigating data contamination issues across pretraining corpora and modern benchmarks.

Method	Level	Access to Training Corpora Required?	Logits Prob. Required?	Retrieval?	Prompt-based?
GPT-3 (Brown et al., 2020)	Instance	✓	✗	✓	✗
PaLM (Chowdhery et al., 2022)	Instance	✓	✗	✓	✗
LLaMA (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. (2023)	Instance	✗	✗	✗	✓
Wei et al. (2023)	Instance	✗	✗	✗	✗
Mattern et al. (2023)	Instance	✗	✓	✗	✗

Table 1: Comparison of current data contamination detection method.

4.2 Time-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) conducted the first comprehensive longitudinal analysis of data contamination in LLMs. Specifically, they leveraged the natural experiment provided by the training cutoffs in GPT models to examine benchmarks released over time. They analyzed 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 conducted similar experiments to assess performance difference in models before and after their release. Besides, Shi et al. (2023) created 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 mask a word or sentence and provide the LLMs with context from a benchmark to guide them in filling in the missing portions. The advantage of this masking-based method is its simplicity and effectiveness. However, it requires a rigorous filtering process to distinct the task from semantic reasoning ones (Deng et al., 2023).

Book-Level Chang et al. (2023) introduce a *name cloze* task, wherein names within a book are masked, prompting LLMs to predict the omitted names. This task was 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. Furthermore, they note that their method depends on the proficient instruction-following capabilities of LLMs. However, in less capable LLMs, there is a tendency to replicate other choices or produce the correct answer without adhering to the given instructions.

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), GSK8k (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 previously unnoticed overlaps in test sets across widely used pretraining and fine-tuning corpora.

Creating Reference Set In addition to directly rephrase 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 Observation

We terms behavior observation as a new perspective that leverages different perspectives of controlling experiment related to the test set. This is done to observe 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.

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, 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) measures the perplexity of a sample to measure the memorization of training data. Carlini et al. (2021) attempts to improve on the Yeom et al. (2018)'s precision and reduce the false negative rate by accounting for the intrinsic complexity of the target point. Further more, Carlini et al. (2022) calibrates 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) introduced 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.

MIA in the context of LLMs are 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 need careful validation of assumption.

5 Mitigating Data Contamination

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

Benchmark Dynamic Refresh Zhu et al. (2023a) introduces 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 backtranslate 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. Besides, Zhou et al. (2023) also suggests that providing a diverse set of prompts for testing, which provide 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 under the licensing agreement. To implement this, the recommended approach involves encrypting the test data before uploading it. This can be efficiently done by compressing the data into an archive that is secured with a password.

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

6 Discussion and Future Discussions

Challenges for Detecting Black-Box Model 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 finetuning 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 finetuning 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. (2024) highlights the ease with which Membership Inference Attack (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 pretrained 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 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 language models 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 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.

References

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.

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. 2023. Elephants never forget: Testing language models for memorization of tabular data. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

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. 2024. Evading data contamination detection for language models is (too) easy.

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.

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.

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.

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. 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.

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. 2021a. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021b. 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.

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.

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.

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.

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

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.

A Example Appendix

This is an appendix.