DOES DATA CONTAMINATION MAKE A DIFFERENCE? INSIGHTS FROM INTENTIONALLY CONTAMINATING PRE-TRAINING DATA FOR LANGUAGE MODELS

Minhao Jiang[1], **Ken Ziyu Liu**[2], **Ming Zhong**[1], **Rylan Schaeffer**[2], **Siru Ouyang**[1], **Jiawei Han**[1], **Sanmi Koyejo**[2]
[1]University of Illinois Urbana-Champaign [2]Stanford University
minhaoj2@illinois.edu

ABSTRACT

Language models pre-trained on web-scale corpora demonstrate impressive capabilities on diverse downstream tasks. However, there is increasing concern whether such capabilities might arise from evaluation datasets being included in the pre-training corpus — a phenomenon known as *data contamination* — in a manner that artificially increases performance. There has been little understanding of how this potential contamination might influence LMs' performance on downstream tasks. In this paper, we explore the impact of data contamination at the pre-training stage by pre-training a series of GPT-2 models *from scratch*. We highlight the effect of both text contamination (*i.e.* input text of the evaluation samples) and ground-truth contamination (*i.e.* the prompts asked on the input and the desired outputs) from evaluation data. We also investigate the effects of repeating contamination for various downstream tasks. Additionally, we examine the prevailing n-gram-based definitions of contamination within current LLM reports, pinpointing their limitations and inadequacy. Our findings offer new insights into data contamination's effects on language model capabilities and underscore the need for independent, comprehensive contamination assessments in LLM studies.

1 INTRODUCTION

The performance of large language models (LLMs) has been attributed primarily to their immense size and the increasing scale of pre-training data from large text corpora (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023; Chowdhery et al., 2022; Anil et al., 2023; Touvron et al., 2023a;b). Nevertheless, a critical aspect that remains under-explored is the potential contamination of the pre-training corpus with evaluation data. This oversight presents challenges in accurately assessing the LLMs' capabilities among other scientific analyses of their behaviors. The importance of contamination analysis in the pre-training corpus has been recognized since pre-trained language models were first introduced (Devlin et al., 2019; Radford et al., 2019; Chowdhery et al., 2022); however, the lack of public access to most pre-training corpora today complicates efforts to comprehensively understand and identify the impact of contamination on a model's performance and behaviors.

Recent LLM reports (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023; Touvron et al., 2023b; Gunasekar et al., 2023) have investigated the contamination of evaluation data in the pre-training corpora from various perspectives. Some of these studies offer limited details on their contamination investigations, especially for closed-source models (Radford et al., 2019; OpenAI, 2023). Others include attempts to investigate the data contamination on the **evaluation level**, where an evaluation dataset is *post-hoc* categorized into contaminated and non-contaminated chunks based on a proposed contamination definition and the model is evaluated on them separately (Radford et al., 2019; OpenAI, 2023; Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023b), to demonstrate that the model is insusceptible to data contamination if the model performs similarly on these chunks. However, this line of work has not adequately analyzed contamination on the **pre-training level**, where the pre-training corpus is deliberately altered to study the effects of contamination on evaluation.

Evaluation data can be leaked into the pre-training corpus in various forms. Predominantly, it is the textual component of the evaluation dataset (*i.e.* via input text). This aspect has been the primary focus of many existing studies, *e.g.*, Touvron et al. (2023b); Chowdhery et al. (2022). There are also many cases where the pre-training corpus might contain *ground truth* information of the evaluation data. Here, we consider *ground truth* of the evaluation samples to be their raw texts *plus* the prompts on such texts and the corresponding answers. Intuitively, contamination involving the ground truth may have different impacts on the models' performance than simple text contamination, but its effects have been under-explored.

This paper investigates the effects of contamination of pre-training data for language models via leakage of evaluation datasets. We pre-train *from scratch* a series of GPT-2 models Radford et al. (2019) and consider various mechanisms of contamination of evaluation data in the pre-training corpus. Specifically, we address three research questions:

1. **RQ1: How are language models affected by the deliberate addition of various forms of contamination on the pre-training corpus?** To answer this, we introduce intentional contamination (with and without the ground truth) into the pre-training corpus (§3.1). We then pre-train GPT-2-small models *from scratch* on these variously contaminated corpora to evaluate and compare their performance. We further extend the experiments with GPT-2-large models to evaluate the effects of data contamination on larger models (§C.4).
2. **RQ2: How do the number of repetitions of evaluation data in the pre-training corpus affect performance?** In practice, *how often* a piece of evaluation data has appeared during pre-training and its ramifications are also unclear. We investigate this by injecting the evaluation data into the pre-training corpus multiple times and provide detailed empirical analyses (§3.2).
3. **RQ3: How effective are the n-gram-based contamination definitions used in recent LLM reports?** We systematically filter out different proportions of contaminated training documents, as described by existing definitions, and pre-train the same model on these cleansed corpora (§3.3). Additionally, we critically evaluate the methods used in current LLM reports for assessing data contamination at the evaluation level (§C.5). These reports often posit that the models exhibit robustness against data contamination, and our discussion aims to elucidate the potential shortcomings of such claims.

We experiment on several commonly used public datasets to observe the performance differences quantitatively. Our analyses provide a new perspective on understanding data contamination in the pre-training of language models. The contributions are summarized as follows:

- We empirically investigate the effects of data contamination in the pre-training corpus due to evaluation data leakage in language models by pre-training language models from scratch to evaluate different mechanisms of data contamination.
- We identify the importance of considering the data contamination with ground truths from the evaluation dataset. Surprisingly, we observed that the effects of increasing the number of repetitions of contamination on the model performance can be U-shaped.
- We critically analyze the n-gram data contamination definitions from existing LLM reports and further compare the empirical results by filtering the pre-training data with these definitions. Our findings suggest that they are insufficient and inadequate to identify contamination.

2 Definitions of Data Contamination

Numerous studies on large language models (LLMs) have explored and investigated the concept of data contamination and demonstrated the robustness of these models against potential contamination in their evaluation datasets Radford et al. (2019); Brown et al. (2020); Chowdhery et al. (2022); OpenAI (2023); Touvron et al. (2023a;b); Gunasekar et al. (2023). Most definitions proposed in the existing studies are based on n-gram duplication between pre-training data and evaluation data. For instance, PaLM (Chowdhery et al., 2022) divides the evaluation data into two categories—"clean" and "contaminated"—based on whether at least 70% of all possible 8-grams in the evaluation sample were seen at least once in the pre-training corpus. Llama 2 (Touvron et al., 2023b) provides a more fine-grained definition: a token is considered contaminated if it appears in any token n-gram longer than 10 tokens in both the evaluation sample and the training set, and the contamination percentage

of an evaluation sample is defined to be the percentage of tokens contaminated; the evaluation data are then divided into 4 buckets—"Clean", "Not Clean", "Not Dirty", and "Dirty"—based on the contamination percentage of each evaluation sample. While intuitive, these contamination definitions primarily revolve around n-gram or token overlaps, which only target direct duplications present in both training and evaluation datasets and might provide both high false positive rate (since many semantically different texts have overlaps) and false negative rate (since simple paraphrasing can evade detection Yang et al. (2023)). Moreover, investigations relying on these definitions have predominantly centered on evaluation level analysis; in our work, we focus on pre-training level analysis as described in §1.

In our experiments, we follow PaLM (Chowdhery et al., 2022) and Llama 2's (Touvron et al., 2023b) definitions as well as a direct n-gram overlap detection strategy to investigate how the "contamination" under these definitions are different and how they affect model performance. As described in §1, contamination in the pre-training corpus can appear as either textual components from evaluation datasets or with ground truth information. Existing definitions tend to overlook the latter. Therefore, we explore two types of contamination when we introduce contamination to the pre-training corpus: (1) **text contamination**, where only the input texts of the evaluation samples are added to the pre-training corpus; and (2) **ground-truth contamination**, where the input texts, the prompts, and the labels/answers of the corresponding evaluation samples are added.

3 Experiments & Analyses

3.1 Effects of Contamination on Evaluation Results

To quantify the effects of data contamination and compare text and ground-truth contamination, we directly evaluate GPT-2$_{original\ /\ text\ /\ gt}$ on each dataset in Table 1 and 2.

Table 1: **Evaluation results on SST-2, MMLU, and SQuAD V1 datasets.** For three variations of models, the experiments are run 3 times, i.e., each pre-training was run under 3 different random seeds, and shown as mean$_{std}$. Since only single checkpoints exist for the public baselines (GPT-2-small, GPT-2-medium, GPT-2-large), we cannot compute variance over multiple training runs.

Model	**Parameters**	**SST-2**	**MMLU**	**SQuAD V1**
		Accuracy	Accuracy	F1 Scores
GPT-2-small$_{original}$	124M	$48.34_{2.32}$	$22.87_{0.09}$	$9.07_{0.19}$
GPT-2-small$_{text}$	124M	$54.89_{0.80}$	$23.03_{0.05}$	$9.78_{0.12}$
GPT-2-small$_{gt}$	124M	$51.02_{0.35}$	$23.13_{0.09}$	$11.45_{0.58}$
GPT-2-small	124M	52.06	23.0	15.09
GPT-2-medium	354M	55.21	23.6	19.94
GPT-2-large	774M	54.01	23.0	17.87

Table 2: **Evaluation results on CNN And Daily Mail dataset.** For three variations of models, the experiments are run 3 times, i.e., each pre-training was run under 3 different random seeds, and shown as mean$_{std}$. Since only single checkpoints exist for the public baselines (GPT-2-small, GPT-2-medium, GPT-2-large), we cannot compute variance over multiple training runs.

Model	**CNN And Daily Mail**							
	ROUGE-1	ROUGE-2	ROUGE-L	Coherence	Consistency	Fluency	Relevance	Overall
GPT-2-small$_{original}$	$24.76_{1.33}$	$8.33_{0.30}$	$16.44_{0.93}$	$0.5382_{0.045}$	$0.6020_{0.013}$	$0.7513_{0.035}$	$0.4952_{0.044}$	$0.5968_{0.010}$
GPT-2-small$_{text}$	$26.84_{0.45}$	$9.03_{0.16}$	$17.91_{0.27}$	$0.5137_{0.016}$	$0.6686_{0.121}$	$0.8225_{0.009}$	$0.4648_{0.014}$	$0.6174_{0.008}$
GPT-2-small$_{gt}$	$28.80_{0.08}$	$10.65_{0.08}$	$19.49_{0.04}$	$0.6390_{0.032}$	$0.7471_{0.012}$	$0.8480_{0.001}$	$0.5644_{0.001}$	$0.6996_{0.015}$
GPT-2-small	27.97	9.43	18.34	0.5725	0.6954	0.8703	0.5525	0.6727
GPT-2-medium	29.71	10.52	19.49	0.6976	0.7998	0.8989	0.6793	0.7689
GPT-2-large	29.97	10.92	19.77	0.7259	0.8253	0.8997	0.6942	0.7863

The experimental results from the two tables reveal the impact of data contamination on model performance across different datasets. The introduction of contamination, either in the text or ground truth, improves model performance compared to the original pre-trained GPT-2 model. Notably, while text contamination does show some improvement in evaluation metrics, the extent of this

enhancement is relatively modest. This is particularly evident in the SQuAD and CNN datasets, where the coherence and relevance scores under text contamination are sometimes lower than those of the original model in the CNN dataset. Conversely, ground-truth contamination generally yields significant performance improvements. However, in the SST-2 dataset, ground-truth contamination does not outperform text contamination. We hypothesize that this is because text classification tasks predominantly depend on the model's comprehension of the input text, rendering evaluation prompts and ground truths less impactful. In fact, they might introduce noise, particularly given that the input texts in the dataset are generally short and that the model is sensitive to prompt formatting. For the MMLU dataset, it's evident that this task presents a significant challenge for GPT-2-small models, as indicated by the poor performance of both the public checkpoints and our pre-trained models. Despite this inherent difficulty, it is noteworthy that we can still observe the performance improvements with the introduction of both types of contamination. Overall, these findings suggest that while both types of contamination can enhance the performance of language models, ground-truth contamination has a more pronounced positive effect on model performance than text contamination in general cases, especially for tasks that require an understanding of the instructions from evaluation prompts, such as CNN and SQuAD datasets.

3.2 Effects of Repeated Contamination Can Be U-Shaped

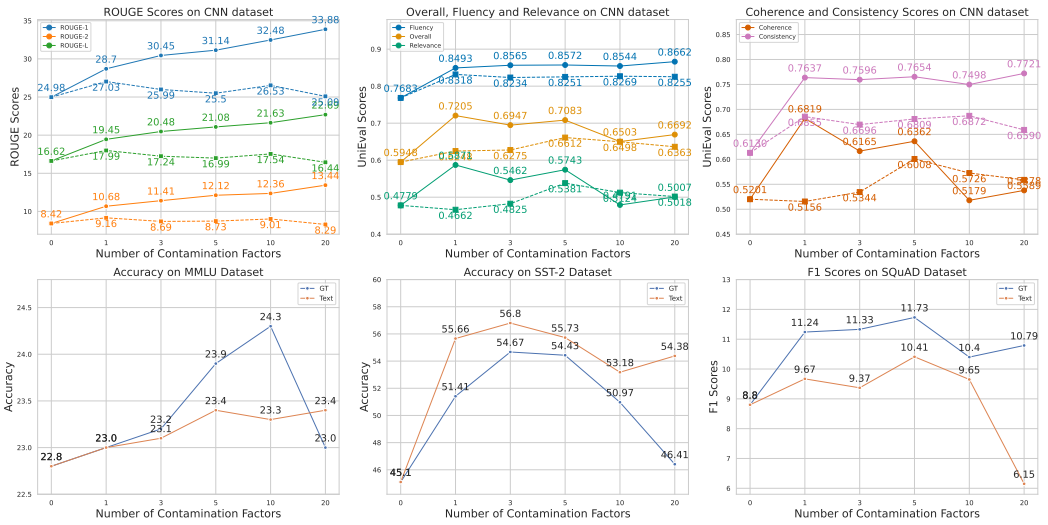

Figure 1: **Evaluation results for different contamination factors from 0 to 20 on each dataset**. Zero repetitions refer to models pre-trained on the original corpus. In the top three figures, the solid lines and the dotted lines show the ground-truth and text contamination results respectively.

We have already observed the effectiveness of data contamination in the previous section, where both the text and ground-truth contamination are only injected into the pre-training corpus once. However, in practice, some fractions of the evaluation datasets may appear in the pre-training corpus more than once given its immense scale. Therefore, in this section, we investigate the effects of *repeated contamination* whereby the evaluation dataset is added to the pre-training corpus multiple times. We use the term **contamination factor** to denote the number of times the evaluation data appear in the pre-training corpus. This analysis is designed to help us understand better how the repetitions of evaluation data for both text and ground-truth contamination, during pre-training might affect the performance. The results are shown in Figure 1.

For SST-2, MMLU, and SQuAD datasets, we observed a distinct U-shaped performance trend in response to increasing contamination factors. Specifically, as the contamination factor increased, performance initially improved but started to decline when the factor reached around 10 repetitions. Notably, at 20 repetitions, performance in some instances dropped below the baseline level observed when there was no contamination. The results for the CNN dataset exhibited varying trends based on the evaluation metrics used. While the ROUGE scores steadily increased with higher contamination factors, the UniEval scores displayed a U-shaped curve similar to the other datasets, which also

indicates a U-shaped general performance trend for the CNN dataset. Another observation is that the fluency score also almost increases monotonically with the increase of contamination factor, which further indicates that fluency is more associated with the size of training data. The divergence in ROUGE scores is primarily attributed to the metrics' focus on the frequency of common subsequences and tokens. These elements are more likely to be repeated with increased data repetition, particularly in scenarios involving ground-truth contamination that repeats correct responses from the dataset.

3.3 Effects of Removing Contamination from Pre-Training

In this section, we conduct experiments to clean the pre-training corpus based on the outlined n-gram and Llama 2 definitions. Specifically, the investigation aims to understand how the contaminated documents under these definitions would affect the performance if we filter them out of the pre-training corpus. As described in §2, we adopt different n-gram values n for the direct n-gram overlap and Llama 2 contamination definitions, and we try various threshold λ for the contamination percentage under Llama 2's definition. These definitions are then used to filter "contaminated" documents out of the pre-training corpus, where a document is considered contaminated if any sentence in this document is considered contaminated. The detailed results are listed in Figure 2.

3.4 More Experiment Results

More discussions of the above experiments and the corresponding analyses are included in Appendix C. In addition to these, we also conducted more experiments to present other results explored in the paper in Appendix C.4 and C.5.

4 Conclusion

In this work, we conduct a *pre-training level* analysis for the effects of data contamination on language models. We pre-train a series of GPT-2 models *from scratch* to study the performance difference in different scenarios, underscoring the vital yet often overlooked role of ground truth in the context of data contamination detection. This aspect is notably absent in existing studies. Our study also sheds light on the effects of repeated contamination on the performance of language models in downstream applications. Moreover, we critically assess the current n-gram-based contamination definitions as reported in recent LLM reports, revealing their inadequacy in accurately identifying true contamination within pre-training corpora. Our replication of the existing robustness evaluations, which focus on evaluation level analysis that divides downstream datasets into different categories, suggests that such assessments fall short of affirming models' robustness to data contamination. Our findings highlight the need for more precise and effective contamination definitions, and the implementation of more stringent methods to ascertain the robustness of LLMs to data contamination.

REFERENCES

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

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. Palm: Scaling language modeling with pathways, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

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. The pile: An 800gb dataset of diverse text for language modeling, 2020.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pp. 28–53. Association for Computational Linguistics, 2023.

Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models, 2021.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences, 2023.

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

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

Saeed Mahloujifar, Huseyin A. Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. Membership inference on word embedding and beyond, 2021.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.719. URL https://aclanthology.org/2023.findings-acl.719.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks, 2022.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://aclanthology.org/K16-1028.

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

Matthew Olson, Abraham Wyner, and Richard Berk. Modern neural networks generalize on small data sets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/fface8385abbf94b4593a0ed53a0c70f-Paper.pdf.

OpenAI. Gpt-4 technical report, 2023.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

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. Llama: Open and efficient foundation language models, 2023a.

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. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources, 2023.

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

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation, 2022.

A Training Hyperparameters

We specify the hyperparameters we use for experiments for reproducibility and consistency of the results. In the GPT-2-small experiments, we set the `batch_size=32`, `learning_rate=0.0005`, `warmup_ratio=0.01`, `weight_decay=0.1`, and all other hyperparameters the same as the default settings of GPT-2-small. For the three runs of the main experiments, we adopt the seed numbers with `42`, `1234`, `2023` to ensure a fair comparison and consistency. For the GPT-2-large experiments, we set `batch_size=128`, `learning_rate=0.0001` `random_seed=42` instead to ensure training stability and keep all other parameters the same.

B Experimental Setup

B.1 Models, Data, and Pre-Training

The model architecture used in our main experiments is GPT-2-small (Radford et al., 2019) (124M parameters) with default hyperparameters. We use a relatively small architecture because pre-training from scratch is computationally expensive. Following Korbak et al. (2023), we construct a pre-training corpus by subsampling 1.95M documents from the Pile (Gao et al., 2020) for a total of 3.3B tokens, which is compute-optimal based on Chinchilla scaling laws (Hoffmann et al., 2022). We later extend our experiments to GPT-2-large (774M parameters) and 19.8B tokens from `pile-uncopyrighted` corpus[1] (§C.4), again following compute-optimal scaling laws. The detailed hyperparameters for all experiments are listed in Appendix A.

B.2 Evaluation Datasets

We focus our experiments on four natural language processing datasets to evaluate the performance of our pre-trained models: SST-2 (Socher et al., 2013), a sentiment analysis dataset; MMLU (Hendrycks et al., 2021), a multi-task natural language understanding dataset; CNN And Daily News (Nallapati et al., 2016), a text summarization dataset that was also evaluated in the GPT-2 report (Radford et al., 2019); the Stanford Question Answering Dataset (SQuAD) dataset (Rajpurkar et al., 2016), which helps evaluating the reading comprehension abilities of the model. The detailed statistics of these datasets are listed in Table 3. All datasets are accessed through HuggingFace[2]. We selected these easier and traditional benchmarks because our goal in the paper is to assess the differential impact of data contamination on GPT-2 models' performance, and the more difficult datasets are likely too challenging for GPT-2 series models.

Table 3: **Evaluation Dataset Statistics.** The last column shows the number of evaluation examples corresponding to each label.

Dataset Name	Split	Label Space	# of Samples
SST-2	train	positive, negative	37,569 / 29,780
MMLU	all/test	A, B, C, D (57 Subjects)	3,222 / 3,462 / 3,582 / 3,776
CNN And Daily Mail	3.0.0/test	-	11,490
SQuAD V1	validation	-	10,600

For evaluation, we follow established processes. For the SST-2 dataset, due to the uncontrollability and instability of the generated results from GPT-2 models, we utilize prompting and the possible labels

[1] https://huggingface.co/datasets/monology/pile-uncopyrighted
[2] https://huggingface.co/datasets/

as hypotheses. We ask the model to score each hypothesis and use the highest one as the prediction. To circumvent prompt sensitivity Liang et al. (2022), we evaluate the accuracy scores based on 10 different prompts for each model. The details of the prompts and the corresponding performance are listed in Appendix D. For MMLU, we utilize AllenAI's official MMLU implementations[3] Wang et al. (2023) to compute the accuracy across 57 different subjects.

For the text summarization task, we follow the original implementation reported in Radford et al. (2019) for evaluation. We add the text `TL; DR: "` after the article to induce the summarization generation. We then ask the model to generate 150 tokens with top-k random sampling with $k = 2$ and use the first 3 sentences of the generated tokens as the summary. We evaluate the generated summaries on the commonly used ROUGE-1, 2, L scores (Lin, 2004) and UniEval (Zhong et al., 2022) to provide a multi-dimensional evaluation. For the question-answering evaluation on SQuAD, we employ the official implementation.[4] In this setup, we allow the model to generate up to 15 tokens, and the first sentence of the generated output is taken as the answer. We subsequently report F1 scores for the generated answers, determined by the overlap of tokens between the model's response and the ground truth. We selected SQuAD V1 to mitigate potential biases introduced by the many no-answer questions in the V2 dataset.

C More Discussions and Experiments

In this section, we present the experiment results to understand how data contamination affects the models' performance quantitatively. We conducted experiments with three variations of contamination, described as follows, where the definitions and discussions for text and ground-truth contamination are presented in §2. For the main experiments, we pre-train the GPT-2-small model from scratch on the corpus to evaluate the performance:

- GPT-2-small$_{original}$ is the model pre-trained on the original corpus described in §B.1.
- GPT-2-small$_{text}$ is the text contamination version of the model. We only add the texts of the corresponding evaluation samples to the training data to ensure that all the texts in the evaluation dataset were 100% contaminated in the pre-training corpus. For MMLU, we also include the texts from the answer choices of each question.
- GPT-2-small$_{gt}$ is the ground-truth contamination variation of the model. On top of the text contamination, we add the same prompt used for evaluation and the ground truth (*e.g.* labels) following the text for each dataset; that is, in the format as *"text + prompt + ground truth"*. For SST-2, we randomly select one out of the 10 prompt templates for evaluation for each evaluation sample and insert it in the corpus as contamination.

As baselines, we further evaluate all datasets on the public checkpoints for GPT-2-small, medium, and large variations to more directly compare the performance, where the pre-training data for the public checkpoints are unknown.

C.1 Discussions of Effects of Data Contamination

Following §3.1, the improvement of ground-truth contamination is more pronounced for the CNN dataset, where it can even improve the model to surpass the performance of public checkpoints and achieve similar performance with the GPT-2-medium model. The experiment results also indicate that fluency, as measured by the UniEval metric, is still lower than the public model checkpoints. We suspect that this observation is due to the smaller scale of training data, where fluency might be more closely related to the model's overall language abilities. We can also observe that there is still an obvious gap between our pre-trained model and the public OpenAI's checkpoints, which shows the importance of the scale of training data.

Viewed together, Tables 1 and 2 demonstrate the effects of data contamination on downstream evaluation tasks and, in particular, the effects of ground-truth contamination. The results highlight the need for methods that can identify and differentiate ground-truth contamination in future studies.

[3] https://github.com/allenai/open-instruct

[4] https://rajpurkar.github.io/SQuAD-explorer

C.2 Discussions of Repeated Contamination

Following §3.2, these findings suggest that while introducing contamination into a pre-training corpus can enhance model performance to a certain degree, over-repetition may lead to a decline in effectiveness. We also note that this threshold for the number of repetitions can be related to the model size and corpus size, which requires more investigation in future works. This is an interesting result since many existing LLMs leveraged huge but *unscrutinized* pre-training corpora that it is unclear: 1) how many times the evaluation data have appeared in the pre-training data, and 2) how the contamination has realistically affected evaluation performance.

On the other hand, we also observe that this U-shape curve for the contamination factor may not universally hold for all datasets and corpora, which we discuss in more detail in Appendix E.

C.3 Discussions of Removing Contamination From Pre-training

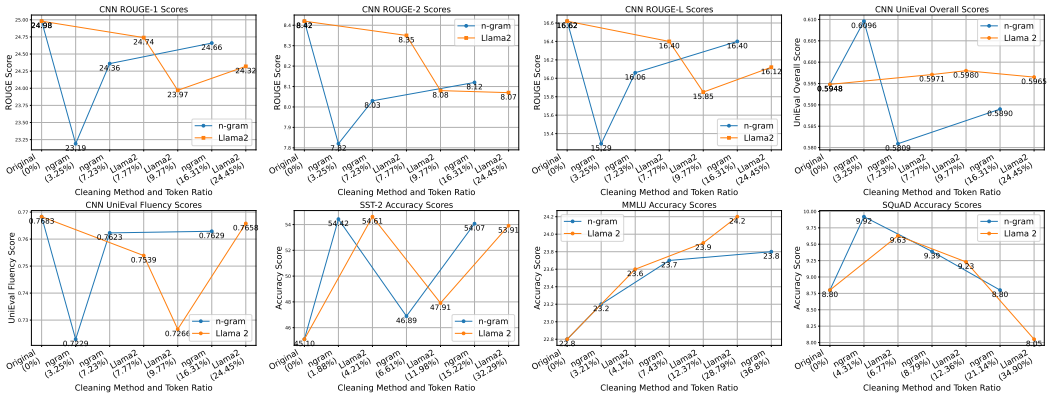

Figure 2: **Evaluation results on removing contamination from the pre-training corpus**. We deliberately select the parameters to achieve different ratios of removed tokens. The x-axis denotes the cleaning method (n-gram or Llama 2) followed by the percentage of tokens removed.

Following §3.3, in our experimental setup, we systematically filter out a range of approximately 3% to over 20% of tokens labeled as "contaminated" from the pre-training corpus, aiming to analyze the effects of the percentage of tokens removed on the model performance. The results, however, do not show a uniform pattern across different proportions of token removal. Interestingly, in certain instances where token removal exceeded 30%, the model's performance remained comparable to that of the original model. This finding raises questions about the accuracy of n-gram-based definitions for pinpointing effective contamination. It appears that documents excluded based on n-gram and Llama 2's definitions are not always genuinely contaminated, which reveals the insufficiency of such definitions for identifying effective contamination in practice.

We did not include PaLM's definition in our experiments since we found this definition is so strict compared to the other two definitions that very few documents would be filtered out. More analyses of the definitions are provided in Appendix F, where we also extensively analyze the effects of varying the parameters of these definitions.

C.4 Scaling Up to GPT-2-Large Model

Table 4: **Evaluation results of GPT-2-large on CNN And Daily Mail and MMLU datasets**.

Model	Parameters	CNN And Daily Mail								MMLU
		Rouge-1	Rouge-2	Rouge-L	Coherence	Consistency	Fluency	Relevance	Overall	Accuracy
GPT-2-large$_{original}$	774M	27.47	9.67	17.74	0.6311	0.6910	0.8376	0.5942	0.6885	22.9
GPT-2-large$_{gt}$	774M	28.43	10.85	18.74	0.6593	0.7335	0.8468	0.6082	0.7117	23.9
GPT-2-large	774M	29.97	10.92	19.77	0.7259	0.8253	0.8997	0.6942	0.7863	23.0

We expand the experiment framework by incorporating GPT-2-large as the base model in our experiment. The primary objective is to assess if the effects of data contamination observed in smaller-scale models would persist in larger models. Due to computation constraints, we focus on the experiments on CNN and MMLU datasets for the ground-truth contamination with a contamination factor of 60, which is used to match the ratio of contamination with GPT-2-small experiments with a contamination factor of 10. A deviation in our setup compared to previous experiments is that we set a fixed number of training steps as opposed to a single epoch over the pre-training set; this is such that the training follows the compute-optimal scaling law for the available number of tokens.

Despite the larger scale of the pre-training corpus in GPT-2-large, the impact of ground-truth contamination is clear. This finding underscores the significant influence of data contamination, which may remain concerning even in a large pre-training corpus.

C.5 Assessing Effectiveness of Evaluation-Level Contamination Analysis

Table 5: **Evaluation results on dividing the evaluation dataset into different categories.** We follow Llama 2's contamination definition and the associated parameters Touvron et al. (2023b) to split the evaluation data. The parameters are shown as n and λ, where n is the n-gram value and λ is the dirty and clean threshold, respectively.

Datasets	Model	Subset Type	n	λ	# of Data	Avg. Contam. %	Results
							Overall
CNN	GPT-2-small$_{original}$	Clean	15	0.85, 0.75	704	72.54	0.5743
		Not Clean			10,786	82.11	0.5920
		Not Dirty			9,203	80.22	0.5898
		Dirty			2,287	86.80	0.5955
	GPT-2-small$_{gt}$	Clean	15	0.85, 0.75	704	72.54	0.6495
		Not Clean			10,786	82.11	0.6986
		Not Dirty			9,203	80.22	0.6950
		Dirty			2,287	86.80	0.6978
							F1 Score
SQuAD	GPT-2-small$_{original}$	Clean	9	0.9, 0.7	571	67.10	9.09
		Not Clean			9,999	81.14	9.61
		Not Dirty			9,741	78.91	9.59
		Dirty			856	97.03	9.24
	GPT-2-small$_{gt}$	Clean	9	0.9, 0.7	571	67.10	9.92
		Not Clean			9,999	81.14	11.39
		Not Dirty			9,741	78.91	11.37
		Dirty			856	97.03	10.21

In this section, we follow recent LLM reports Chowdhery et al. (2022); Touvron et al. (2023b) to divide evaluation data into different categories to see what we can learn from contamination analysis on the evaluation level. Specifically, we follow Llama 2's definitions and methods Touvron et al. (2023b) to divide the evaluation data into four categories ("Clean", "Not Clean", "Not Dirty", and "Dirty") and evaluate the model on each category separately.

We adopt relatively high clean/dirty threshold values λ in order to arrive at similar portions of data for each category compared to Llama 2. We observed that the number of samples in each category is very sensitive to λ.

We select CNN and SQuAD datasets and divide them into four categories based on the definitions and parameters described in Table 5. We evaluate both the original model and the ground-truth contamination version of the model to see if the contamination will make a difference. Table 5 shows that the performance for the four categories is similar to each other. Even though the "clean" category under ground-truth contamination exhibited marginally lower results compared to the other categories, there was no clear indication that the "dirty" category outperformed the non-dirty categories. The fact from the previous experiments that the performance of the evaluated models can be boosted by contamination shows that these models are not immune to contamination in the pre-training corpus.

These results suggest that it may be insufficient to conclude that models are insusceptible to contamination based on such categorical evaluations. This draws attention to the need for more rigorous methodologies to assess the robustness of LLMs against data contamination accurately.

D Evaluation of Classification Tasks

In this section, we describe the details of different prompts we utilized for the evaluation of SST-2 datasets. We select the prompts with different meanings and lengths to ensure the diversity of prompt formats, and the results for GPT-2$_{original}$ are shown in Table 6. We can observe from the table that GPT-2-small models are quite sensitive to how prompts are structured in downstream tasks. This suggests we need more research to better understand and evaluate small language models on classification tasks, especially when the answers of the models are not within the label space, which can be addressed in future studies.

Table 6: Evaluation results of SST-2 Accuracy Scores for the 10 Different Prompts.

Prompts	GPT-2-small$_{original}$	GPT-2-small$_{text}$	GPT-2-small$_{gt}$	GPT-2-small	GPT-2-medium	GPT-2-large
Datasets	SST-2	SST-2	SST-2	SST-2	SST-2	SST-2
{text} It is {label}	43.87	49.97	55.44	56.09	61.77	51.94
{text} The text is {label}	42.98	49.81	55.60	54.36	58.47	53.92
{text} The sentiment for this text is {label}	44.20	48.27	51.73	51.55	54.38	45.83
{text} The preceding text is {label}	44.07	44.96	50.76	47.73	45.65	54.35
{text} If the preceding text could be categorized as positive or negative, it would be {label}	43.96	46.12	46.41	56.21	52.12	57.16
{text} The sentence is {label}	43.56	50.32	56.62	55.52	58.94	55.77
{text} This text is {label}	44.13	48.62	48.60	45.06	52.91	55.57
{text} Determine the sentiment of the preceding text: positive or negative: {label}	44.47	44.52	53.05	55.78	55.73	55.85
{text} The text belongs to {label}	43.56	57.52	47.88	50.46	55.91	56.30
{text} The sentiment for this sentence should be {label}	44.23	57.58	53.69	47.78	56.23	53.48

E More Discussions on Data Contamination

In this section, we show the experiment results for the AG News dataset, where we observe that the data contamination does not match the observation we had in our main experiments.

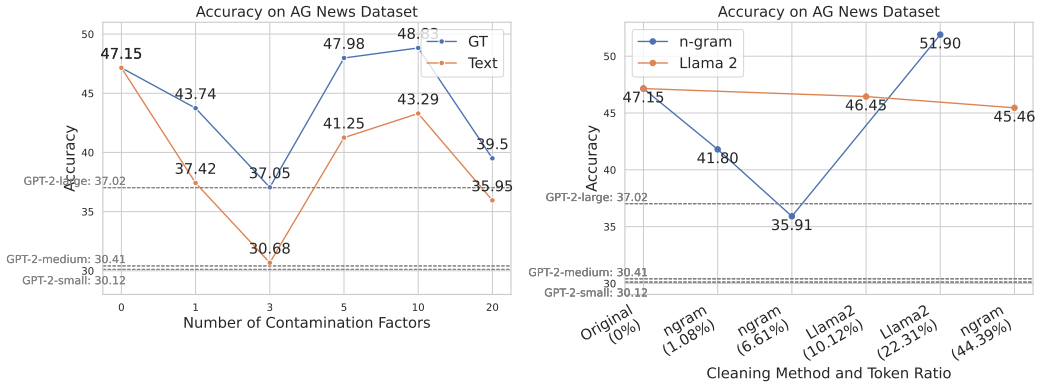

Figure 3: The evaluation results for AG News dataset on both contamination factor and removing contaminated data experiments. The performances for public model checkpoints from OpenAI are displayed as dotted lines in both figures.

We can observe that even the performance of the model pre-trained on the subsampled corpus is already higher than the OpenAI's public checkpoints. Interestingly, unlike previous experiments, we found that introducing text and ground-truth contamination does not significantly enhance performance. As we increase the contamination factors, the performance generally begins to decline at higher levels of contamination, as the U-shape trend in the previous experiment suggested, but with the lowest performance occurring at a contamination factor of 3. On the other hand, no matter how we increase the contamination factors, the performance is still much higher than the public checkpoints. One plausible explanation for this phenomenon is that the models may be assimilating or memorizing information from the AG News dataset present in the subsampled corpus. Consequently, the addition of various types of contamination does not yield substantial performance improvements and results in strange observations in this case.

This result suggests that the effects of data contamination on language models still require more effort to understand how knowledge of language models is constructed during pre-training.

F Quantitative Analysis for Contamination Definitions

In this section, we analyze the different sets of parameters for different contamination definitions proposed in the previous studies to examine our evaluation dataset and pre-training corpus. We use the contamination ratio of the pre-training corpus for each evaluation dataset as a comparison to assess how strict these definitions are and the appropriate contamination definitions.

F.1 N-Grams Direct Overlap

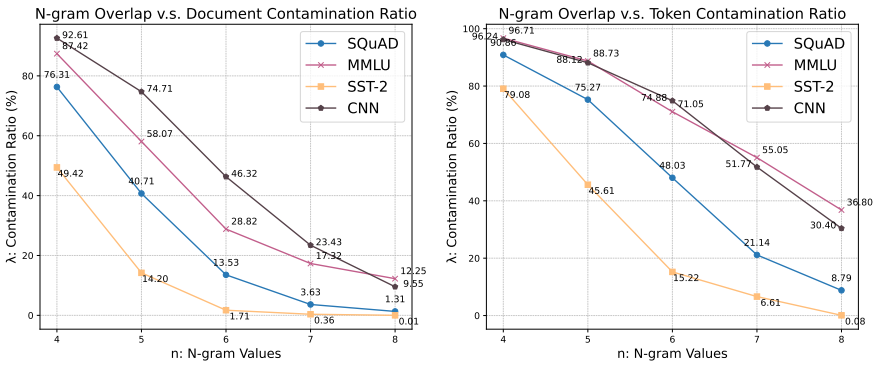

Figure 4: N-gram direct overlap contamination ratio w.r.t. different n-gram values for each dataset.

First, we examine the straightforward definition of contamination: the direct n-gram overlap within sentences of a training document. A training document is considered *contaminated* if any n-gram in the document appears in the evaluation dataset. While this approach offers a direct measure of dataset duplication, its scope is limited. Solely relying on n-gram overlaps may overlook other forms of contamination since sentences can be rephrased in various ways, conveying identical meanings without any overlapping n-grams. Therefore, direct n-gram overlap is only to demonstrate how much of the content in the evaluation dataset appears in the pre-training corpus. During our filtering, a sentence is considered contaminated if any n-gram in the sentence appears in both pre-training data and evaluation data, and a document is considered contaminated if any sentence in this document is contaminated. We also report the total number of tokens in these documents that are considered contaminated. As shown in Figure 4, we calculate the contaminated ratio of documents and tokens in the pre-training data for different n's for comparison. We can observe that the contamination ratio varies for each dataset and how to define a reasonable threshold n for the n-gram would be dependent on the text length of the evaluation dataset. For instance, in the SST-2 dataset, where many sentences comprise fewer than eight words, applying an 8-gram threshold would be impractical. Conversely, a very small n-gram value may fail to capture semantically meaningful content within sentences.

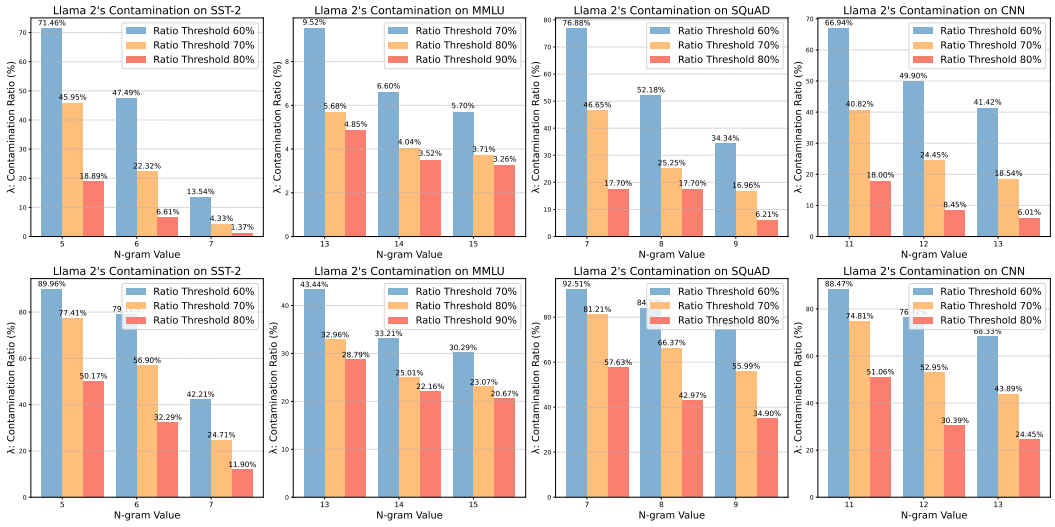

Figure 5: Contamination ratio for pre-training data based on Llama 2's definitions. We adopt the n-gram values that make the contamination ratio within a similar range and threshold from $60\% - 90\%$ for comparison.

F.2 PaLM and Llama 2's Definitions

We conduct similar analyses for PaLM and Llama 2's definitions by considering different n-gram values n and contamination threshold λ. PaLM's definition extends n-gram direct overlap to consider the overlapping percentage of n-grams in one sentence: a training document is considered *contaminated* if more than λ percentage of n-grams in a sentence of the document appear in the evaluation dataset. We observe that this definition is so strict that very few documents can satisfy it even if we relax n and λ to very small values compared to the original definition. The results for Llama 2's definitions are shown in Figure 5. We report the percentage of contaminated documents and the percentage of tokens respectively. We can observe that the Llama 2 definitions lead to varied levels of identified contaminated documents and tokens, depending on the chosen parameters. These definitions concentrate on token contamination through n-gram duplication, which can be problematic because tokens may have different meanings in different contexts. Relying only on token duplication can misclassify sentences as contaminated. Additionally, similar to the straightforward n-gram definitions, setting the correct n-gram values and thresholds for different datasets remains a challenge with this approach.

We provide more detailed results for different parameters of these definitions, along with the PaLM's results, in Table 7 to better observe the trends for each definition.

G Related Work

Data Contamination Definition and Investigation. The exploration of data contamination has been a consistent element in LLM reports, dating back to the initial discussions of the memorization problem in BERT (Devlin et al., 2019). Recent LLM reports (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023; Touvron et al., 2023a;b) have delved deeper into how evaluation data may be duplicated within pre-training corpora. These studies typically analyze the robustness of models against data contamination through n-gram-based definitions; the analysis is also typically focused on the evaluation level as opposed to the pre-training level (recall §3.1). However, such definitions may not accurately detect real contamination, casting doubt on the definitive conclusions drawn from these studies. Recent LLM studies also investigated the embedding-based contamination definitions. The contamination analysis explored in phi-1/1.5 (Gunasekar et al., 2023; Li et al., 2023) involves n-gram-based and embedding and syntax-based definitions but only focuses on code data. These studies represent a preliminary investigation in understanding the role of data contamination in the pre-training corpus. Another recent work (Yang et al., 2023) shows

Table 7: More results for the ratio of contaminated documents for different datasets with different definitions under different parameters.

Datasets	Filtering Method	N-gram Value	Threshold	% of Contaminated Documents
SST-2	PaLM	5	10%	1.22%
	PaLM	5	50%	$\approx 0\%$
	PaLM	7	50%	$\approx 0\%$
	PaLM	7	70%	0.0003%
SQuAD	PaLM	5	50%	0.077745%
	PaLM	5	70%	0.007744%
	PaLM	4	50%	0.081334%
	PaLM	4	70%	0.037795%
	Llama 2	6	70%	76.38%
	Llama 2	6	80%	43.08%
CNN	PaLM	7	70%	0.0896%
	PaLM	8	70%	0.0302%
	Llama 2	14	70%	14.71%
	N-gram	9	-	3.48%
	N-gram	10	-	1.32%
	N-gram	11	-	0.54%
MMLU	Llama 2	12	80%	6.76%
	Llama 2	15	95%	3.07%
	Llama 2	18	90%	2.36%
	Llama 2	20	90%	1.92%
	Llama 2	24	90%	0.61%

that the existing n-gram-based and embedding-based definitions can be easily evaded by applying simple paraphrasing of evaluation data, emphasizing the urgent necessity for proper definitions of contamination and reliable detection methods.

Data Contamination and Memorization. Memorization in neural networks has been a well-explored topic in machine learning. Previous work has studied how memorization connects to and differs from generalization (Olson et al., 2018; Magar & Schwartz, 2022; Feldman, 2020), analyzed memorization in language models (Carlini et al., 2023; Nasr et al., 2023), and studied how memorization connects to privacy (Ippolito et al., 2023) and data extraction attacks (Carlini et al., 2021; Nasr et al., 2023). Memorization is closely linked to data contamination as the model performance on evaluation data is no longer trustworthy if the evaluation data were memorized, regurgitated, and reasoned upon. Because of this connection, past work also explored membership inference attacks (MIA) for language models (Mahloujifar et al., 2021; Jagannatha et al., 2021; Mireshghallah et al., 2022; Carlini et al., 2022; Mattern et al., 2023; Shi et al., 2023). However, these methods can sometimes be computationally intensive, and more generally, example-based matching can lead to false negatives in flagging contamination, *e.g.*, detection can be evaded through paraphrasing (Yang et al., 2023). Other recent work has sought to identify pre-training data contamination heuristically by examining the likelihoods of texts after changing their ordering (Oren et al., 2023) and of least probable tokens (Shi et al., 2023). Nevertheless, these methods are similarly inadequate for detecting textual transformations and the heuristic nature of these methods may limit them from providing a clear understanding of how data contamination impacts the model performance on the pre-training level, highlighting a need for more comprehensive methods in this area of research.

H Limitations and Discussions

This study has been specifically designed to investigate the impact of data contamination during the pre-training stage on the performance of language models. To maintain a focused and controlled examination of this effect, we deliberately excluded stages such as instruction tuning, fine-tuning, and RLHF from our analysis. This was done to mitigate potential confounding factors inherent in these stages, thereby concentrating our investigation on pre-training and zero-shot settings. However, it is important to recognize that the comprehensive effects of data contamination across all stages

of model training warrant further exploration. Our research serves as an initial step in this broader inquiry.

Another limitation lies in our selection of GPT-2 series models for experimentation. Given the computational and resource constraints, these models, which are relatively smaller in scale compared to the contemporary large-scale language models, were chosen for their manageability and the feasibility of manipulating the training process and pre-training data in a clear and reproducible manner. This approach aligns with precedents set by previous research focused on understanding model behavior during the pre-training phase. Nonetheless, it raises questions about the applicability of our findings to larger models, which may exhibit different behaviors under similar conditions of data contamination. In conclusion, while our study contributes valuable insights into the effects of data contamination during the pre-training stage, the generalizability of these findings to other stages of model training and to larger-scale models remains an open question. As such, our work should be viewed as an initial exploration of a complex field that demands further research. Future studies are needed to build on our findings, extending the investigation to encompass a broader range of models and training stages, to fully understand the nuances of data contamination in language model training.