Deciphering the Impact of Pretraining Data on Large Language Models through Machine Unlearning

Anonymous ACL submission

Abstract

Through pretraining on a corpus with various sources, Large Language Models (LLMs) have gained impressive performance. However, the impact of each component of the pretraining corpus remains opaque. As a result, the organization of the pretraining corpus is still empirical and may deviate from the optimal. To address this issue, we systematically analyze the impact of 48 datasets from 5 major categories of pretraining data of LLMs and measure their impacts on LLMs using benchmarks about nine major categories of model capabilities. Our analyses provide empirical results about the contribution of multiple corpora on the performances of LLMs, along with their joint impact patterns, including complementary, orthogonal, and correlational relationships. We also identify a set of "high-impact data" such as Books that is significantly related to a set of model capabilities. These findings provide insights into the organization of data to support more efficient pretraining of LLMs.

1 Introduction

Under the data-driven paradigm, Large Language Models (LLMs) have demonstrated promising performance and showcased immense potential in further promotion (OpenAI et al., 2023; Touvron et al., 2023b; Du et al., 2021; Bai et al., 2023a). Previous analyses have suggested that the composition of the pretraining corpus may exert a significant impact upon the performance of LLMs (Longpre et al., 2023; Shen et al., 2023a). However, how different sources and types of pertaining corpora influence the *knowledge and reasoning ability* of LLMs largely remains opaque, or stays at the qualitative level. As a result, it still heavily relies on the experiences of trainers to organize the pre-training corpus. Such experiences may deviate from the optimal, and hence limit the efficiency and effectiveness of model training.

In this paper, we propose to quantify how components with different sources and types in the pretraining corpus contribute to the performance of LLMs. Previous literature refers to such analyses as Data Influence Analysis (DIA) (Akyürek et al., 2022). However, due to the limitations of previous DIA methods, the DIA of LLMs remains challenging. Primary DIA methods can be mainly categorized into two lines: the retraining-based methods and gradient-based methods. Retraining-based methods work by removing specific data from the training corpus and retraining the model, then comparing changes in model performance. Considering the prohibitive training cost of LLMs, retraining-based methods would be impractical (Nguyen et al., 2023). While the gradient-based methods may not be applicable in analyzing the source of the complex reasoning ability of LLMs, as they assume that the performance upon a test instance is determined by several independent training instances. However, such an assumption may not hold for LLMs, especially for the ability to complete reasoning tasks, as it may originate from groups of correlated instances that jointly contribute to the performance of LLMs. For example, solving math problems requires understanding a knowledge taxonomy, and the taxonomy is described by a set of interdependent instances holistically. Missing one component would lead to the collapse of the whole taxonomy. Hence, the gradient-based methods may fail to trace the influence of such a whole corpus, which is of vital importance (Grosse et al., 2023).

To address these issues, we resort to another strand of method, Machine Unlearning. Prior research (Eldan and Russinovich, 2023; Jang et al., 2022) suggests that Machine Unlearning can selectively erase specific knowledge from a model through gradient *ascent* on corresponding instances. This enables us to investigate the influence of a certain pretraining corpus on an LLMs by "Unlearning" instances from it, and then compare the perfor-

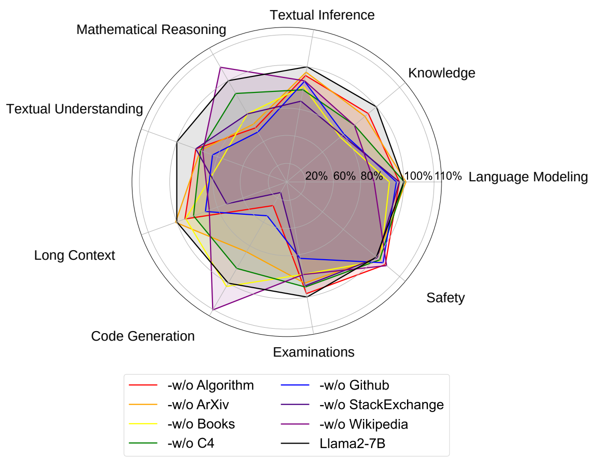

Figure 1: The impact of Unlearning different types of corpus on different abilities of the llama2-7B model.

mance of the "forgotten" LLMs with the original LLMs. Meanwhile, different from previous Machine Unlearning methods, to avoid unintended impacts on non-targeted samples, we incorporate additional regularization by retraining samples from non-targeted domains during the Unlearning process. Experiments demonstrate that our method can effectively remove information contained in the target samples, without significantly affecting other unrelated samples.

Based on our customized Machine Unlearning method, we systematically investigated the quantitative contribution of multiple important resources and types of training corpora on the performance of LLMs. We covered widely adopted high-quality corpora GitHub, Wikipedia, ArXiv, Books, StackExchange and C4, and conducted an in-depth analysis of their contributions to the model performance by segmenting them into subsets based on the type of knowledge. From the content dimension, the abovementioned corpora covered text, commonsense knowledge, domain-specific knowledge, math, and coding. Additionally, to investigate the source of the reasoning abilities of LLMs, we analyzed the impact of over ten kinds of programming languages with different coding paradigms, and 17 kinds of common algorithms such as Dynamic Programming. Programming paradigms essentially represent different abstractions of real-world problems, and an algorithm corresponds to a common solution for a particular type of reasoning problems. To comprehensively assess the impact of these corpora, we evaluated the "forgotten" LLMs across various downstream tasks as illustrated in Figure 1 and detailed illustrated in Figure 4.

With our proposed gradient ascent-based Machine Unlearning method, we identify a range of corpora that have a significant impact on the capabilities of LLMs, and reveal previously unreported influences of certain corpus (such as algorithms) on model capabilities. Furthermore, we discovered the interaction among corpora which jointly affects the capabilities of LLMs, and the existence of "high-energy" corpus like Books in the data. Our research underscores the importance of further studying the impact of pre-training data, to provide foundations for future research on the optimization of pre-training datasets to support more efficient pre-training process.

2 Methodology and Validity Analysis

2.1 Machine Unlearning Based Data Influence Analysis

Following Eldan and Russinovich (2023) and Jang et al. (2022), we devise our approach based on the Machine Unlearning to eliminate certain kinds of information from LLMs.. Formally, given an LLMs M and a sample set $\mathcal{D}_T$, we analyze the influence of $\mathcal{D}_T$ upon M by making M "unlearn" the information of $\mathcal{D}_T$ to derive a model M_T^u, and then compare the performance between M and M_T^u. For clarity, we call M_T^u as the forgotten model, $\mathcal{D}_T$ as the targeted corpus, and the rest parts of the whole pretraining corpora as the non-targeted corpora.

The key of Machine Unlearning-based data influence analysis lies in that: (1) Effectiveness: How to unlearn $\mathcal{D}_T$ to make that M has never been trained upon $\mathcal{D}_T$. (2) Precision: Do not incur unintentional impacts upon the non-targeted parts of the training corpus. To this end, we devise an **GR**adient **A**s**C**ent-based Machine Unlearning with r**E**-training (GRACE). In the following sections, we describe the mechanism of GRACE and show the effectiveness and precision of GRACE.

2.2 Gradient-based Machine Unlearning with Retraining

During the training process, LLMs learn knowledge by **maximize** the likelihood of training corpora through gradient descent. Hence, in line with Eldan and Russinovich (2023); Jang et al. (2022), the information within a targeted corpus $\mathcal{D}_T$ could be unlearned by reverting the learning process through gradient *ascent* on $\mathcal{D}_T$. Formally, the objective function of the Unlearning algorithm is to **minimize** the log-likelihood upon $\mathcal{D}_T$.

However, there remains the risk that the performances on non-target domains are unintentionally impacted. To avoid this problem, GRACE introduces an additional retraining regularization. Specifically, the information within a non-targeted corpus $\mathcal{D}_N$ could be revised through gradient descent upon $\mathcal{D}_N$.

Hence, the whole algorithm runs in the following manner. Before starting, we first divide the *non-target* corpus $\mathcal{D}_N$ into a 9:1 split as a *retraining set* and a *dev set*. Then during the Unlearning process, if the model M's Perplexity (PPL) on the *dev set* is higher than that before Unlearning, a retraining is started until the PPL of M on the *dev set* restore to the original level on the *dev set*. At this time the Unlearning process would restart. In this way, the Unlearning and retraining alternate until the PPL on the target corpus $\mathcal{D}_T$, reaches the endpoint (which is described below), the GRACE algorithm would be ended.

In practice, the retraining set is constructed by randomly sampling instances from the rest of RedPajama dataset (Computer, 2023) after excluding the target corpus. For instance, if we aim to unlearn the C language, the retraining set is set to be a random subset of the remainder of the RedPajama dataset after excluding the C language portion. Note that, to increase the diversity of the retraining dataset and prevent model performance degradation on unrelated domains, at each round of retraining, we would resample 30, 000 new instances.

Endpoint of the Unlearning Process A critical issue of the Machine Unlearning algorithm is when to stop the Unlearning process, so that the forgotten model M_u^T can approximate the state as if the original model M has never seen the target corpus $\mathcal{D}_T$. Prior methods achieve this by case study (Eldan and Russinovich, 2023) or manually selecting certain corpus $\mathcal{D}_S$ that is highly similar to $\mathcal{D}_T$, whereas M has never been trained on it. So that the performance of M on the unlearned dataset $\mathcal{D}_S$ can be taken as the endpoint of Unlearning on $\mathcal{D}_T$. However, since the data filtering process of LLMs is opaque, it is hard to find a specific corpus that the model has not been trained on for each kind of target corpus.

To address this issue, we propose a randomized text-based method. Specifically, given an instance from $\mathcal{D}_T$, we tokenize it and randomly split the tokens into pieces with a length range from 1 to n, and then we shuffle their order and paste the shuffled pieces into a randomized text. The endpoint of the Unlearning process is defined to be the point that the M_T^u's PPL on $\mathcal{D}_T$ equals M's PPL on the randomized text.

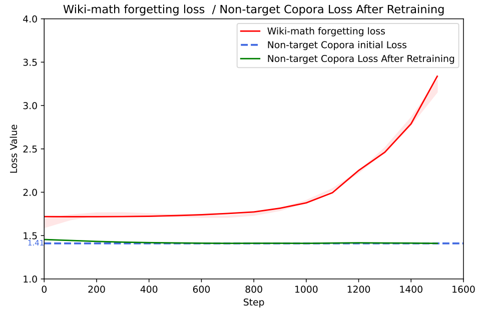

Figure 2: Comparative Analysis of Model Unlearning Effects between the $target_{math}$ and $non\text{-}target_{math}$.

Name	Mathematics	Physics	Chemistry
Before	5.60	5.48	5.47
After	28.14	10.64	7.75
Name	Biology	Economics	History
Before	5.66	5.51	5.41
After	6.97	6.66	6.06
Name	Psychology	Law	Linguistics
Before	6.15	4.69	5.85
After	7.05	5.08	5.97

Table 1: Perplexity of LLama on subsets of the Wikipedia corpus after Unlearning the Math subset of Wikipedia.

This is because: (1) Through randomizing, the knowledge, semantic, and logical relationship within $\mathcal{D}_T$ are disrupted, hence, if the PPL of M_T^u on $\mathcal{D}_T$ is close to the PPL of M on the corresponding randomized text, it suggests that M_T^u has completely forgotten $\mathcal{D}_T$; (2) Compared to $\mathcal{D}_T$, the randomized text shares a similar lexical distribution, which would eliminate the influence of domain-specific vocabulary distribution.

2.3 Validity Analysis of GRACE

Previous analyses demonstrate the effectiveness of gradient ascent-based Machine Unlearning methods in eliminating certain knowledge of LLMs (Eldan and Russinovich, 2023). We conduct further analyses to show the effectiveness of GRACE in Unlearning certain domains of knowledge and certain kinds of reasoning abilities, and the precision of not incurring unwanted impacts.

Experimental Settings To validate the precision and effectiveness of our methodology in selectively Unlearning certain domains of knowledge, we conduct Machine Unlearning using GRACE and take the mathematical subset of the Wikipedia

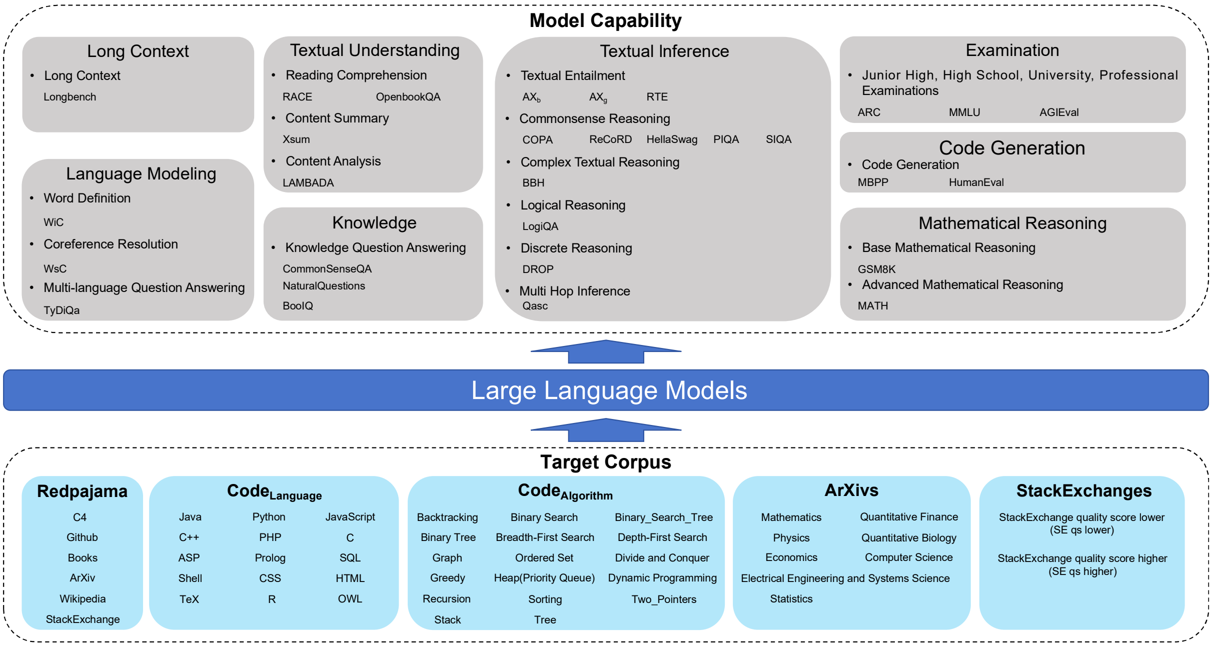

Figure 3: The overall framework of the experiment.

corpus (Computer, 2023) as the target corpus $\mathcal{D}_{\text{math}}$. For the non-target dataset, we randomly select samples from the rest part of Redpajama corpus (Computer, 2023) which explicitly excluded $\mathcal{D}_{\text{math}}$. To analyze the impact of GRACE, we not only evaluate the performance on the targeted domain `math`, but also include potentially related domains `physics`, `chemistry`, and `biology` and unrelated domains: `economics` `history`, `psychology`, `law` and`linguistics`. Experiments are conducted with Llama-2-7B (Touvron et al., 2023b), more details are provided in Appendix A.

Analyses We demonstrate the loss curve on the target corpus and the non-target corpora in Figure 2, and the final PPL of the forgotten model M_u on each domain in Table 1. It can be observed that the loss of the model on $\mathcal{D}_{\text{math}}$ continuously increases during the Unlearning process. In contrast, due to the additional retraining process of GRACE, there is no significant increase in the loss for the non-target data. This suggests that GRACE would not incur unintentional model performance on the non-target domains. Moreover, as shown in Table 1, after the Unlearning process, the performances on the `physics`, `chemistry`, `biology` and `economics` domains, demonstrate a degradation, while the model performance upon `history`, `psychology`, `law` and `linguistics`, remain unaffected. Interestingly, the extent of performance degradation in these domains is consistent with human cognition about the relevance of these domains with math: `physics` and `mathematics` are rather closely related; `chemistry`, `biology`, and `economics` share certain common grounds with `mathematics`. In contrast, the correlation between historical, psychological, legal, and linguistic knowledge with mathematical knowledge is quite limited. These observations suggest that GRACE can eliminate certain domains of knowledge from LLMs without involving unwanted impacts, indicating the effectiveness and precision of our proposed method. In the Appendix B, we provide more evidences about the validity of our analysis method.

3 Main Analysis

After the validity analysis, we employ GRACE to investigate the impact of various corpora on the performance of LLMs. Specifically, Section 3.1 introduces the experimental settings. Section 3.2 explains how different types of data affect model performance individually. Section 3.3 discusses the joint impact of various types of data on the abilities of LLMs.

3.1 Experimental Settings

3.1.1 Target Corpora

Since the ultimate goal of data influence analysis is to provide empirical guidance for optimizing the organization of the pretraining corpus of LLMs, among various open-sourced datasets, we focus our study on various subsets of the Redpajama dataset (Computer, 2023), a replication version of the pretraining corpus of Llama (Touvron et al., 2023a; Chen et al., 2023; Fu et al., 2024). Moreover, considering the importance of complex reasoning ability, to further investigate the source of such ability, we include a set of programming algorithmics, as they can be viewed as an abstraction of thought patterns. As shown in Table 3, these datasets have been further divided into subsets, a total of 48 distinct datasets are chosen as target corpora. Specifically:

- All subsets of **RedPajama** (Computer, 2023), including: C4, Github, Books, ArXiv, Wikipedia, StackExchange. These corpora play pivotal roles in the pretraining corpus of various LLMs (Ren et al., 2023; Zhang et al., 2024).
- The **ArXivs** contains eight subsets of the Arxiv dataset, as listed in Figure 3
- The **StackExchanges** dataset is obtained by dividing the StackExchange portion of the Redpajama.This subsets into two subsets based on the number of "likes" each Q&A pair has received. Intuitively, the more likes an answer receives, the more likely it is to be a high-quality answer.
- **Code$_{\textbf{Algorithm}}$** contains 17 kinds of important leetcode algorithm problems (Hartford, 2023).
- The **Code$_{\textbf{Language}}$** dataset is derived from the GitHub corpus, encompassing 15 types of programming language, spanning a variety of programming paradigms including Object-Oriented and Procedure-oriented languages, Declarative languages, Scripting languages, Front-end languages, as well as unctional language.

We provide more details about the target corpora in the Appendix D.

3.1.2 Evaluation Benchmarks

To comprehensively evaluate the influence of target corpora on LLMs' performance, following Contributors (2023); Gao et al. (2023), as shown in Figure 3, we select a totally of 31 benchmarks covering 9 major ability of tasks and 21 sub-categories of capabilities. Details about these datasets and the experimental settings are provided in Appendix C.

3.1.3 Model for Analysis

We conducted all experiments using the widely adopted open-source decoder-based generative LLMs Llama-2-7B. The reasons lie in that: (1) llama2-7B is large and powerful enough to represent LLMs; (2) The training process of the llama2-7B is typical; and (3) The existence of the Scaling Law (Kaplan et al., 2020) allows us to infer the impact of various types of data on larger models by examining their effects on the Llama2 model.

3.2 Impact of Individual Corpus on Model Capabilities

Analysis Method As the difficulty of benchmarks is different, to make the performance changes on these benchmarks comparable, we first normalize them to the *performance degradation ratio*, which is defined as $\gamma_{i,u} = \frac{A^u_{i,j} - A^o_j}{A^o_j}$, where $A^u_{i,j}$ is the model's performance on task j after unlearning the target-data$_i$, and A^o_j represents the performance of the original model on task j. In the below, we measure the impact brought by Machine Unlearning using the *performance degradation ratio*.

Analysis Results Figure 4 lists the Top and Bottom 5 datasets that have the most and the least impact on each type of model capability. From which we can observe that:

- **Language Modeling** As a fundamental of LLMs, the language modeling ability is seldom significantly impacted by a specific type of corpus alone. As an (approximate) multi-lingual parallel corpus, the Wikipedia corpus may play a critical role in aligning different languages for an LLMs, and influence the multilingual ability of LLMs.
- **Textual Understanding** One prominent phenomenon is that programming language corpora have a high impact on the textual understanding ability of LLMs. Heuristically, codes are abstractions of relationships between real-world objections and could be helpful for understanding the semantic and logical relationships among text. Moreover, knowledge-rich corpora such as books and Arxiv, and corpora with high diversity such as books and C4 profoundly influence the textual understanding ability of LLMs. Hence, corpora with diversity, rich commonsense knowledge and code corpus may constitute three foundations for the textual understanding ability of LLMs.
- **Textual Inference** Text inference tasks depend on a wide range of corpora. Notably, besides commonsense-related corpora, text reasoning tasks also extensively rely on various types of code corpora, algorithm corpora, and mathematical corpora. For example, tasks like Big Bench Hard (Suzgun et al., 2022) significantly depend on high-quality StackExchange content and algorithms such as breadth-first search. This demonstrates the importance of symbolic reasoning capabilities represented by mathematics and code in understanding the deep logical relationships within texts.

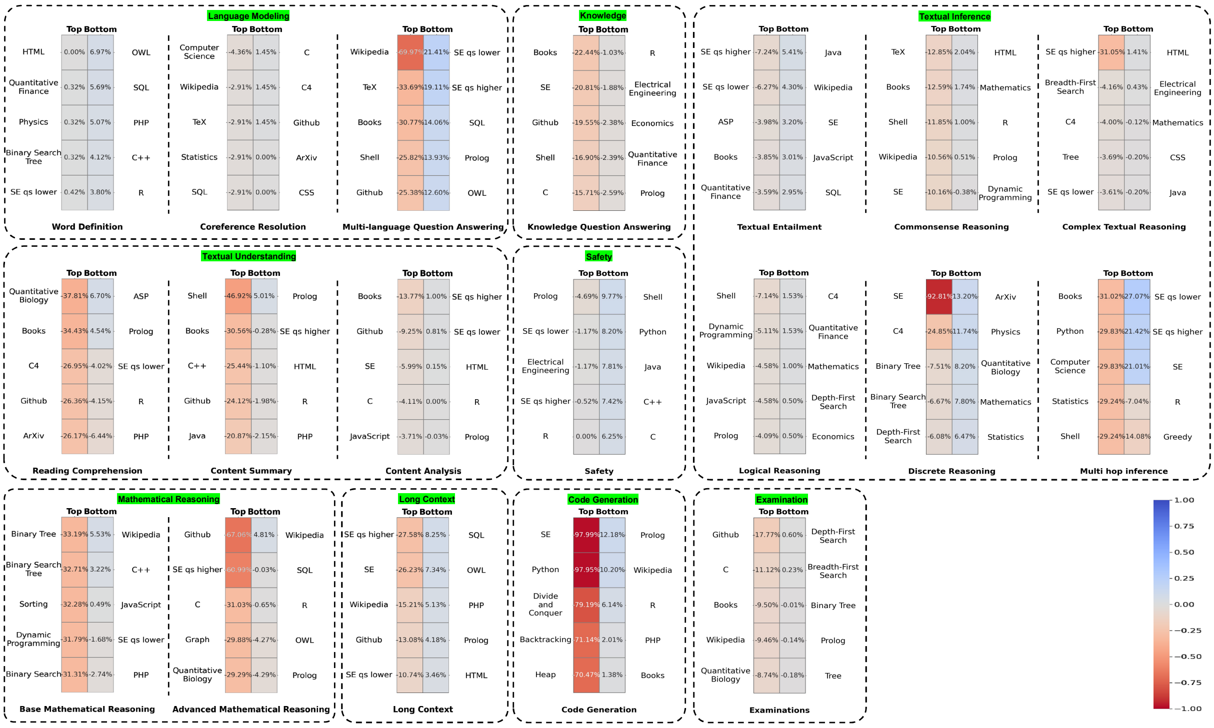

Figure 4: The Top and Bottom 5 datasets that have the most and the least impact on each type of model capability.

- **Knowledge Reasoning** Datasets such as Books, and are invaluable for solving real-world knowledge problems. Programming languages, as an integral part of the global knowledge system, significantly influence the model's knowledge capabilities. The ArXivs dataset, due to its complexity and deviation from common world knowledge, has a lesser impact.
- **Mathematical Reasoning** The source of LLMs' math reasoning capabilities has drawn great attention from researchers, as it could be an indicator of the complex reasoning ability of LLMs (Ernest, 2023). From the results in Figure 4, high-quality mathematical texts and code corpora (especially algorithms) have a significant impact on LLMs' math reasoning abilities. This demonstrates: (1) There is a close relationship between mathematical and coding abilities (Soldaini et al., 2024; Shen et al., 2023b). To some extent, both math and code problems are abstractions of real-world problems, and involve complex symbolic reasoning processes to solve them. Hence, model performances demonstrate high sensitivity upon the algorithmic. (2) High-quality mathematical texts are key to the model learning of mathematical abilities.
- **Code Generation** Compared to text-based tasks, the range of knowledge for code generation tasks is relatively narrow, only limited to mathematics and code-related corpora Shen et al. (2023a). This again demonstrates the close relationship between mathematics and coding. Overall, forgetting algorithmic knowledge has a greater impact on the model's coding ability than specific programming languages. This suggests that the model's understanding of algorithms does not depend on specific program languages. In other words, LLMs could understand the logic of algorithms, instead of memory algorithmic knowledge depending on certain programming languages.
- **Long Text** GitHub, Wikipedia, Books, and "High-liked" StackExchanges significantly impact the model's long text capabilities, as these corpora are composed of long texts, entailing complex logical relationships and abundant common sense knowledge.
- **Examination** Completing exam questions requires extensive knowledge and strong reasoning abilities. Hence, code, commonsense, mathematics, and books corpora all form the foundation of an LLMs's examination capabilities.
- **Safety** Interestingly, the model's security is enhanced after forgetting the code corpora. This might be due to the absence of emotional factors in the code corpora and its straightforward logic, thus making the generated results more crude and aggressive.

More detailed experimental data and results can be found in Appendix E.

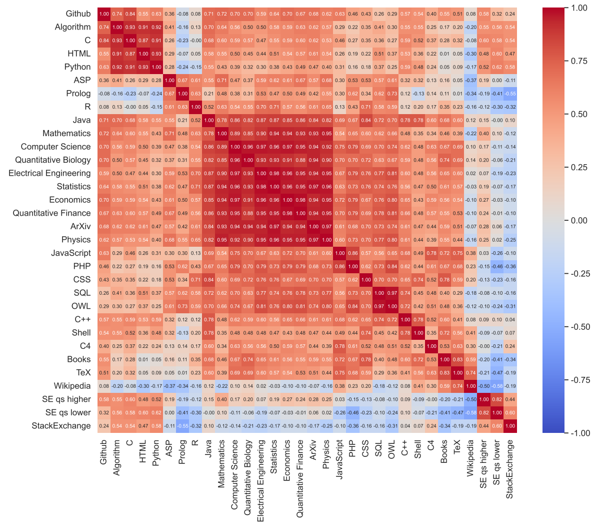

Figure 5: A correlation matrix based on the model's performance across 19 capabilities after experiencing data Unlearning.Among them,"Algorithm" is the average value of all Algorithm.

3.2.1 Corpus with Broad Influences

We calculate corpora that can influence multiple capabilities. Since they may lead to a broad influence on model capabilities, these datasets may serve as the foundation of the training corpus.

Analysis Method In the target-data, certain datasets significantly influence numerous capabilities of the model. We adopted the same method for data processing as described in Section 3.2. Furthermore,We define a dataset as "High-impact data" if its removal leads to a performance decline in over 70% of capabilities, which exceeds the average decline observed across all datasets.

Analysis Results Among the selected target-data, four categories—Books, Shell and Github meet the criteria for High-impact data. Unlearning Books datasets result in a decline in 16 capabilities beyond the average, whereas Unlearning Shell and Github leads to a decline in 14 capabilities beyond the average. These datasets significantly impact multiple model capabilities, and play a crucial role in model training.

3.3 Joint Impact of Multiple Corpora on Model Capabilities

Previous research indicates that LLMs can combine information from multiple kinds of corpora, and generalize it to new tasks. Hence, during the pretraining stage, multiple corpora may perform joint impact upon LLMs. In this section, we explore measuring the joint contribution of multiple datasets upon the LLMs.

3.3.1 Interrelationships Among Data

Analysis Method To explore the relationships between datasets from different domains, we calculated the Pearson correlation coefficient between any two target corpus T_A and T_B, based on the model's performance degradation ratio upon 19 capabilities after Unlearning T_A and T_B. Within each category of abilities, each subtype was given an equal weight. When a subtype included multiple datasets, we averaged the performance changes across these datasets to calculate the coefficients. These coefficients form a correlation matrix. If the correlation coefficient between two datasets is positive, it indicates that the impact of these two datasets on downstream tasks is similar, and vice versa. Subsequently, we conducted hierarchical clustering on the correlation matrix to categorize datasets. Figure 5 displays the correlation matrix rearranged according to the categories defined by the hierarchical clustering.

Analysis Results As shown in Figure 5, according to their relationships, the corpora could be categorized into three types, which we name as "Correlated Corpora", "Complementary Corpora", and "Orthogonal Corpora", respectively. Specifically:

- **Correlated Corpora** refers to corpora that the model has similar performance changes after Unlearning them. In other words, they have a similar influence on LLMs. For instance, the correlation coefficients among Economics, Quantitative Finance, and Statistics are all greater than 0.95. Hence, to some extent, the correlated corpora can substitute for each other in the training corpus, leading to redundancy in the training corpus and a waste of computation resources. Hence, it would be necessary to reorganize these corpora to enhance pre-training efficiency.
- **Complementary Corpora** refers to corpora that the model performance alternations are different after Unlearning them. Interestingly, our analyses suggest that there may exist certain corpora that have a complementary influence on the model's performance. For example, the Wikipedia corpus and SE qs lower corpus have a negative correlation coefficient -0.58. As detailed illustrated in Appendix E, this negative correlation is because of both Wikipedia and SE qs lower have impacts on multiple capabilities, while the capabilities influenced by these two datasets seldom overlap. Thus, these two corpora could act simultaneously as a critical composition of the pretraining

corpus. In general, the math corpora (e.g., StackExchange) have a complementary relationship with commonsense-related corpora such as Books or Wikipedia. Note that our results also suggest the existence of an extreme case that the inclusion of one dataset can cause a decline in the performance of another dataset. This situation requires further verification. If it indeed exists, then in organizing the pre-training data, a trade-off must be made between the two datasets.

- **Orthogonal Corpora** refers to two corpora having a correlation coefficient near zero. For example, the correlation coefficient between Wikipedia and ArXiv is -0.07. This suggests that these corpora independently contribute to the model's different capabilities with low redundancy. Hence, when optimizing the organization of the pretraining corpus, each of the orthogonal corpora should not be recklessly excluded to avoid impairing the comprehensiveness of the pre-training dataset.

Moreover, according to the correlation matrix, there seems to be several groups of data: (1) Math-related group composed of StackExchange related corpora; (2) Knowledge-related group including the subsets of Arxivs (e.g., Statistics / Economic). Note that, the group (1) and (2) seems to be complementary; (3) The intermediate group between (1) and (2) composed of C, python, prolog etc., which bridges (1) and (2).

Additionally, employing a similar methodology, we analyze the relationship patterns between model abilities. Owing to space constraints, the results are presented in Figure 11 of the Appendix. In summary, we observe that abilities can be both positively correlated (synergistic) and negatively related (antagonistic), indicating that there are inherent conflicts between certain capabilities, necessitating a trade-off. Such conflict also indicates that the term "scaling law" refers to the *scaling law under the same data composition*.

4 RELATED WORK

The Data Influence Analysis (DIA) task aims at finding how each training data contributes to a model's performance. DIA methods can be mainly classified into two categories (Hammoudeh and Lowd, 2022). The first category, the Retraining-Based approach (Jia et al., 2021; Kandpal et al., 2022; Ghorbani and Zou, 2019), assesses the influence of certain instances by comparing the model performance with and without these instances. However, due to the prohibitive training costs, these methods have only been extensively applied in the 'small" models (Jia et al., 2021; Ghorbani and Zou, 2019; Kandpal et al., 2022; Nguyen et al., 2023).

The second category Gradient-Based methods (Koh and Liang, 2017; Koh et al., 2019; Pruthi et al., 2020; Hara et al., 2019) discover training samples with greater influence by comparing gradient similarities between training and test instances. These methods demonstrate efficacy in finding instances contributing to knowledge memorization of models. However, they may fail to find instances related to the reasoning abilities of models, which is especially important for LLMs, as it may originate from groups of correlated instances that jointly contribute to the performance of LLMs. Considering the limitations of these methods, we propose a Machine Unlearning-based DIA approach to investigate the impact of corpora upon LLMs.

Machine Unlearning is devised to erase certain knowledge from a model. Prior research (Jang et al., 2022; Graves et al., 2021; Gupta et al., 2021; Sekhari et al., 2021) suggests that machine unlearning can selectively erase specific knowledge from a model through gradient *ascent* on corresponding instances. Eldan and Russinovich (2023) has shown that gradient ascent methods would still be effective in LLMs, and can accurately unlearn targeted samples. In this paper, we further extend the gradient ascent-based machine unlearning methods by involving an additional retraining process and a random text-based stop criterion.

5 Conclusion

In this study, we employed a Machine-Unlearning-based data influence analysis method GRACE to investigate the complex effects of diverse types of pretraining data on the performance of Large LLMs. We gained empirical analysis results about how specific components of the pretraining corpus influence LLMs capabilities, and how they jointly contribute to multiple capabilities of LLMs. Our findings suggest the nuanced impact of data selection and organization in LLMs development. The identification of high-impact data and the delineation of complementary, antagonistic, and orthogonal data relationships offer guidance for optimizing pre-training data organization. In future work, we consider adapting our analysis methodology to other parts of LLMs training such as supervised fine-tuning.

6 Limitations

This study systematically investigated the impact of various pre-training datasets on the capabilities of LLMs using the GRACE method. While our findings offer valuable insights into the subtle relationship between pre-training data and model capabilities, it is crucial to acknowledge the inherent limitations of our approach.

6.1 Data Limitations

There remains a still remains space for exploration in the domain of data. In terms of breadth, our study primarily focused on the Redpajama dataset and its subsets. However, there are other datasets for LLMs, and Redpajama may not include all the corpora encountered during the training process of LLMs. In terms of depth, some important data domains may still possess ample subdivision space. For example, Books datasets can be segmented by books type, which could limit the comprehensiveness of our analysis.

6.2 Limitations of Evaluation Metrics

The existing evaluation systems might not adequately unearth the deeper capabilities of models, potentially overlooking subtle variations and thereby missing valuable insights.

6.3 Future Research Directions

Addressing these limitations provides opportunities for future research. Using a broader variety of data and applying our analytical framework to other models are key steps toward a more comprehensive understanding of the relationship between pre-training data and LLMs capabilities. These efforts will contribute to the ongoing discussions about optimizing pre-training strategies to enhance model performance and efficiency.

References

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Towards tracing knowledge in language models back to the training data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2429–2446.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023b. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

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

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. *arXiv preprint arXiv:2107.03374*.

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. Skill-it! a data-driven skills framework for understanding and training language models. *arXiv preprint arXiv:2307.14430*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

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. *arXiv preprint arXiv:2110.14168*.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

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

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *North American Chapter of the Association for Computational Linguistics*.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*.

Paul Ernest. 2023. Abduction and creativity in mathematics. *Handbook of Abductive Cognition*, pages 585–611.

Weimin Fu, Shijie Li, Yifang Zhao, Haocheng Ma, Raj Dutta, Xuan Zhang, Kaichen Yang, Yier Jin, and Xiaolong Guo. 2024. Hardware phi-1.5 b: A large language model encodes hardware domain specific knowledge. *arXiv preprint arXiv:2402.01728*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11516–11524.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. 2023. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330.

Zayd Hammoudeh and Daniel Lowd. 2022. Training data influence analysis and estimation: A survey. *arXiv preprint arXiv:2212.04612*.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. 2019. Data cleansing for models trained with sgd. *Advances in Neural Information Processing Systems*, 32.

Eric Hartford. 2023. Leetcode solutions. https://www.kaggle.com/datasets/erichartford/leetcode-solutions/. Kaggle data set.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021c. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. 2021. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8239–8247.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 05, pages 8082–8090.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. 2019. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32.

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

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *arXiv preprint arXiv:2305.13169*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Elisa Nguyen, Minjoon Seo, and Seong Joon Oh. 2023. A bayesian approach to analysing training data attribution in deep learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, et al. 2023. Gpt-4 technical report.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Siyu Ren, Qi Jia, and Kenny Q Zhu. 2023. Context compression for auto-regressive transformers with sentinel tokens. *arXiv preprint arXiv:2310.08152*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

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

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. 2023a. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. 2023b. Slimpajama-dc: Understanding data combinations for llm training.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, et al. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

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

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

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

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

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

Figure 6: The performance of the model on an unaltered Python statement after selectively Unlearning programming languages, including Python, C, Shell, C++, HTML, and SQL.

A Experiment details

All experiments were conducted on an A100-SXM4-80GB cluster, with all models utilizing the bf16 precision format.For each target corpora, a subset with 2000 instances is randomly sampled the "Unlearning set"

B Example Appendix

Analysis Method As illustrated in Figure 6, we evaluate the performance of the model on an unaltered Python statement after selectively Unlearning programming languages, including Python, C, Shell, C++, HTML, and SQL. The darker the color, the greater the increase in token loss compared to the original model.

Analysis Results Notably, the most significant loss increase is observed after the model unlearns Python, indicating a profound impact on its ability to interpret Python statements. Furthermore, when Unlearning other languages, the changes in model loss predominantly occur in comments and certain keywords, which are syntactically or semantically shared across these languages. Languages like C++, C, and Shell, which structurally resemble Python to some extent, exhibit substantial loss variations. Conversely, HTML and SQL, being more distinct from Python, result in minimal changes in the model's performance on Python statements after being forgotten. Additionally, we observe that due to the object-oriented features of Shell and C++, Unlearning these languages leads to a notable increase in the loss associated with the "class" token. In contrast, given C's procedural nature, the loss related to "class" does not show significant variation. This analysis underscores the intertwined nature of programming language knowledge within the model and highlights the differential impact of Unlearning specific languages on the model's comprehension of Python code.

Analysis Method As illustrated in Figure 7, we evaluate the model's performance on Breadth-First search statements written in C++ after selectively applying Unlearning algorithms, including depth-first and graph algorithms. The darker the color, the greater the increase in token loss compared to the original

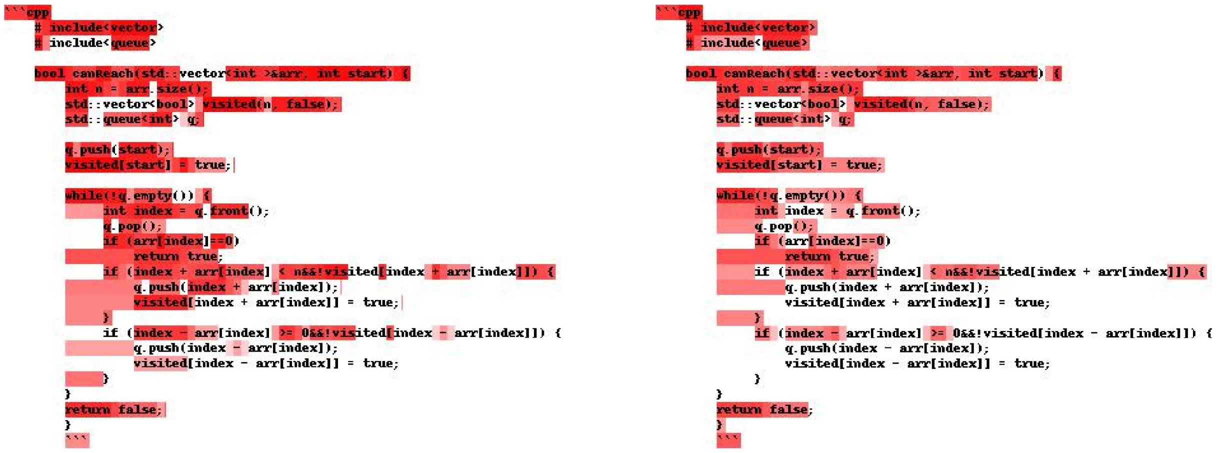

Figure 7: The performance of the model on an unaltered BFS statement after selectively Unlearning programming languages, including DFS and Graph.

model **Analysis Results** As illustrated in Figure 7, the performance of the model post-omission of the Depth-First Search (DFS) algorithm on Breath First Search (BFS) data is depicted at the right, while the performance post-omission of the Graph algorithm on BFS data is shown at the left. It is observable that there is a significant increase in loss for both cases in terms of variable definitions and certain key terms. However, during the execution phase of the BFS algorithm, the loss in the model after omitting the Graph algorithm is substantially greater than that after omitting the DFS algorithm.

C Test DaseSet

In Table 2, we integrate the frameworks (Contributors, 2023; Gao et al., 2023) to construct a comprehensive evaluation system for large models.

Language Modeling. The average performance on WiC (Pilehvar and Camacho-Collados, 2019), WSC (Levesque et al., 2012), and TyDiQa (Clark et al., 2020) is reported. These test sets are all evaluated with 0-shot results.

Knowledge. The average performance is reported using BoolQ (Clark et al., 2019), CommonSenseQA (Talmor et al., 2018), and NaturalQuestions (Kwiatkowski et al., 2019). We report 8-shot results for CommonSenseQA and 0-shot results for all other benchmarks.

Textual Inference. The average performance is reported using AX_b (Wang et al., 2020), AX_g (Wang et al., 2020), RTE (Wang et al., 2020), COPA (Roemmele et al., 2011), ReCoRD (Zhang et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), SIQA (Sap et al., 2019), BBH (Suzgun et al., 2022), LogiQA (Liu et al., 2020), DROP (Dua et al., 2019), and Qasc (Khot et al., 2020). We report 3-shot results for BBH, 2-shot results for DROP, and 0-shot results for all other benchmarks. Notably, AX_b, AX_g, and RTE are used for Textual Entailment tasks; COPA, ReCoRD, HellaSwag, PIQA, and SIQA for Commonsense Reasoning tasks; BBH for Complex Textual Reasoning; LogiQA for Logical Reasoning; DROP for Discrete Reasoning; and Qasc for Multi-hop Inference.

Mathematical Reasoning. The average performance is reported using GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021c). GSM8K represents Base Mathematical Reasoning, while MATH represents Advanced Mathematical Reasoning. These test sets are all evaluated with 4-shot results.

Textual Understanding. The average performance is reported using RACE (Middle and High) (Lai et al., 2017), OpenbookQA (Mihaylov et al., 2018), Xsum (Narayan et al., 2018), and LAMBADA (Paperno et al., 2016). RACE (Middle and high) and OpenbookQA are used for Reading Comprehension tasks, Xsum for Content Summary tasks, and LAMBADA for Content Analysis tasks. These test sets are all evaluated with 0-shot results.

Long Context. The model's ability in long text understanding and reasoning is represented using English datasets within Longbench (Bai et al., 2023b).

Code Generation. We report the average pass@1 scores of models on HumanEval (Chen et al., 2021)

Ability		Test DaseSet
Language Modeling	Word Definition	WiC (0- shot)
	Coreference Resolution	WSC (0- shot)
	Multi-language Question Answering	TyDiQa (0- shot)
Knowledge	Knowledge Question Answering	BoolQ (0- shot)
		CommonSenseQA (8- shot)
		NaturalQuestions (0- shot)
Textual Inference	Textual Entailmen	AX_b (0- shot)
		AX_g (0- shot)
		RTE (0- shot)
	Commonsense Reasoning	COPA (0- shot)
		ReCoRD (0- shot)
		HellaSwag (0- shot)
		PIQA (0- shot)
		SIQA (0- shot)
	Complex Textual Reasoning	BBH (3- shot)
	Logical Reasoning	LogiQA (0- shot)
	Discrete Reasoning	DROP (2- shot)
	Multi hop inference	Qasc (0- shot)
Mathematical Reasoning	Base Mathematical Reasoning	GSM8K (4- shot)
	Advanced Mathematical Reasoning	MATH (4- shot)
Textual Understanding	Reading Comprehension	RACE (0- shot)
		OpenbookQA (0- shot)
	Content Summary	Xsum (0- shot)
	Content Analysis	LAMBADA (0- shot)
Long Context	Long Context Understanding Long Context Reasoning	Longbench
Code Generation	Code Generation	MBPP (1- shot)
		HumanEval (0- shot)
Examination	Junior High, High School, University, Professional Examinations	ARC (0- shot)
		MMLU (5- shot)
		AGIEval (0- shot)
Safety	Safety	TruthfulQA (0- shot)

Table 2: Classification of the Test Data.

and MBPP (Austin et al., 2021). MBPP is evaluated with 1-shot results, while HumanEval is evaluated with 0-shot results.

Examination. The average performance is reported using ARC (easy and challenge) (Clark et al., 2018), MMLU (Hendrycks et al., 2021b,a), and AGIEval (Zhong et al., 2023). We report 5-shot results for MMLU and 0-shot results for all other benchmarks. It is worth noting that for AGIEval, we only selected English datasets.

Safety. Performance is represented using TruthfulQA, evaluated with 0-shot results.

D Target DaseSet

The **$\text{Code}_{\text{Algorithm}}$** corpus contains building on the leetcode dataset created by Hartford (2023), selected 17 of the most important algorithms. Following the methodology of Wang et al. (2022), for algorithm types with not enough size, we utilized GPT-4 OpenAI et al. (2023) for data augmentation to ensure a minimum of 2, 000 samples per algorithm. Furthermore, to ensure the model's Unlearning pertains to the algorithms themselves and not to a specific programming language, each algorithm problem was represented in five language formats: C++, Python, Java, JavaScript, and pseudocode.

The **$\text{Code}_{\text{Language}}$** dataset is derived from the GitHub portion of the Redpajama dataset Computer (2023). In terms of language selection, based on the characteristics of programming languages, this study chose 15 types, including Object-Oriented, Procedure-oriented, Declarative programming, Scripting, frontend and other common languages. It is specifically filtered to include data where a single language proportion for more than 99.99% of the content and the total length is in excess of 2000 bytes, aiming to isolate target data for a specific language. From this filtered data, 2000 samples are randomly selected for each language to constitute the $\text{Code}_{\text{Language}}$ dataset.

The **Arxivs** dataset is constructed based on the Arxiv part of the Redpajama dataset Computer (2023), which is further divided into eight main categories. Randomly select 2000 samples from each category to

Target Data	name		
Redpajama	C4	Github	Books
	ArXiv	Wikipedia	StackExchange
$\text{Code}_{\text{Language}}$	C	C++	CSS
	HTML	Java	Python
	PHP	JavaScript	Shell
	R	Web Ontology Language	SQL
	TeX	ASP	Prolog
$\text{Code}_{\text{Algorithm}}$	Backtracking	Binary Search	Binary Search Tree
	Binary Tree	BreadthFirst Search	DepthFirst Search
	Divide and Conquer	Dynamic Programming	Graph
	Greedy	Heap (Priority Queue)	Ordered Set
	Recursion	Sorting	Stack
	Tree	Two Pointers	
ArXiv	Physics	Mathematics	Computer Science
	Quantitative Biology	Quantitative Finance	Statistics
	Electrical Engineering		
StackExchanges	StackExchange quality score lower	StackExchange quality score higer	

Table 3: Classification of the Target-Data.

form the Arxiv dataset.

The **StackExchanges** dataset is constructed from the StackExchanges segment of the Redpajama dataset Computer (2023), comprising Q&A pairs that have garnered more than 5 "likes". Subsequently, it is segregated into two tiers based on the count of "likes". The underlying rationale is that the number of "likes" is indicative of an answer's quality and popularity. Consequently, these five subsets encompass samples ranging in quality and popularity from the lowest to the highest.

E Experimental Result and Joint Impact of Multiple Ability on Model Capabilities

	Llama-2-7B	w/o Backtracking	w/o Binary Search	w/o Binary Search Tree	w/o Binary Tree	w/o Breadth FirSE Search	w/o Depth FirSE Search	w/o Divide and Conquer	w/o Dynamic Programming
WiC	49.53	50.78	50.31	49.69	49.84	50.31	50.16	50.16	50.31
WSC	66.35	65.38	64.42	64.42	64.42	64.42	64.42	64.42	64.42
TyDiQa	23.74	22.73	22.37	22.55	22.33	22.68	22.24	22.3	23.07
BoolQ	70.67	68.96	68.72	68.47	68.2	68.84	68.99	68.47	69.24
CommonSenseQA	66.67	63.96	64.7	64.37	64.7	63.8	64.21	64.21	64.37
NaturalQuestions	19.2	16.84	16.76	16.26	16.43	16.37	16.15	16.62	17.51
AX_b	53.53	51.9	53.17	51.54	51.18	52.08	51.63	53.35	53.08
AX_g	55.34	53.09	51.97	51.12	53.37	51.69	51.4	52.53	51.12
RTE	49.82	55.23	53.43	50.9	50.9	53.43	52.71	55.6	54.51
COPA	67.0	68.0	68.0	64.0	65.0	67.0	65.0	66.0	68.0
ReCoRD	32.65	24.53	25.03	23.58	23.79	24.36	23.52	23.69	27.44
HellaSwag	74.0	73.4	73.66	73.51	73.64	73.4	73.51	73.37	73.77
PIQA	78.18	77.91	77.75	77.42	77.58	77.97	77.69	77.37	78.4
SIQA	48.46	45.75	45.04	45.14	45.85	46.93	46.93	44.63	46.67
BBH	39.18	38.18	38.83	38.34	38.21	37.55	37.98	38.58	38.14
LogiQA	30.11	29.34	29.03	29.49	29.95	29.95	30.26	29.65	28.57
DROP	30.75	30.41	29.93	28.7	28.44	29.38	28.88	29.21	31.06
Qasc	18.47	14.58	14.9	14.15	14.15	15.12	14.36	14.58	14.69
GSM8K	16.45	11.9	11.3	11.07	10.99	12.13	12.13	11.9	11.22
MATH	3.28	2.62	2.42	2.54	2.44	2.44	2.54	2.36	2.7
$RACE_{middle}$	40.39	29.04	29.32	30.36	30.29	31.96	30.85	28.83	32.24
$RACE_{high}$	37.31	28.87	28.56	29.99	30.39	31.1	30.62	27.9	31.68
OpenbookQA	57.2	51.0	50.8	53.0	54.2	51.8	53.2	49.8	52.6
Xsum	18.16	15.04	14.83	15.05	15.06	14.76	14.71	14.89	15.02
LAMBADA	73.3	71.74	72.11	71.88	72.21	72.23	71.86	71.9	71.86
LongBench	30.09	29.17	29.12	29.02	28.91	29.71	28.82	28.7	29.45
MBPP	17.0	8.6	9.0	10.0	9.8	8.0	7.8	6.2	7.8
HumanEval	12.8	0.0	2.44	5.49	6.1	2.44	2.44	0.0	3.05
ARC_e	58.73	59.26	58.91	59.26	60.85	61.02	61.38	58.2	60.85
ARC_c	41.69	39.32	37.97	42.37	42.03	42.37	42.71	37.97	40.34
MMLU	45.89	45.15	45.39	45.66	45.84	45.67	45.58	45.06	45.08
AGIEval	26.94	25.96	26.04	25.61	25.73	25.92	26.12	25.65	26.24
TruthfulQA	31.33	32.19	32.44	33.17	32.68	32.44	32.93	33.05	31.95

	w/o Graph	w/o Greedy	w/o Heap (Priority Queue)	w/o Ordered Set	w/o Recursion	w/o Sorting	w/o Stack	w/o Tree	w/o Two Pointers
WiC	50.31	50.47	50.78	50.47	50.31	50.47	49.84	49.84	50.16
WSC	64.42	64.42	64.42	64.42	64.42	64.42	64.42	64.42	64.42
TyDiQa	22.45	22.88	22.63	22.62	22.61	22.6	22.4	22.35	22.56
BoolQ	69.05	69.05	68.96	69.3	68.99	68.5	69.14	68.26	68.5
CommonSenseQA	63.88	65.6	64.78	65.11	64.7	64.7	65.11	64.54	64.78
NaturalQuestions	17.15	17.73	16.12	17.53	17.15	15.87	16.87	16.29	16.98
AX_b	51.9	54.62	52.72	53.26	53.17	54.89	53.35	51.27	53.26
AX_g	52.25	52.53	51.97	51.69	53.65	54.21	53.93	52.53	55.06
RTE	53.43	53.43	54.51	54.51	54.51	54.15	53.43	52.35	53.43
COPA	68.0	68.0	68.0	67.0	68.0	66.0	68.0	64.0	65.0
ReCoRD	24.01	25.78	24.35	24.84	25.71	21.6	25.42	23.54	24.79
HellaSwag	73.45	73.99	73.88	73.81	73.76	73.42	73.93	73.59	73.74
PIQA	77.97	78.07	77.75	78.02	77.31	77.64	77.8	77.53	77.48
SIQA	46.32	46.11	45.34	45.65	45.04	44.88	45.19	45.75	44.63
BBH	37.81	39.09	38.34	38.66	38.86	39.0	38.87	37.74	38.92
LogiQA	29.65	29.65	29.8	29.34	29.8	29.49	29.49	30.11	29.49
DROP	29.39	30.26	30.85	30.27	30.5	29.56	30.5	29.0	29.52
Qasc	15.12	15.87	14.9	15.23	14.58	15.87	14.47	14.47	14.69
GSM8K	13.5	12.43	11.37	11.75	12.28	11.14	12.36	11.6	12.05
MATH	2.3	2.8	2.56	2.8	2.58	2.36	2.56	2.42	2.5
$RACE_{middle}$	33.08	31.82	30.57	30.99	28.9	30.5	29.25	30.36	28.69
$RACE_{high}$	31.93	31.45	29.93	30.36	28.85	29.33	28.85	29.99	28.16
OpenbookQA	52.0	51.8	52.6	51.2	51.0	52.4	51.4	53.2	50.6
Xsum	15.24	14.81	14.7	14.65	14.95	14.57	14.93	14.77	14.87
LAMBADA	71.78	72.37	72.15	71.8	72.4	72.31	72.17	72.0	72.11
LongBench	29.67	29.34	29.09	29.03	29.05	28.81	29.04	28.79	28.83
MBPP	7.4	9.6	8.8	9.8	9.0	8.4	9.6	9.4	8.4
HumanEval	3.66	4.88	0.0	1.22	0.0	2.44	0.0	4.27	0.61
ARC_e	60.85	60.67	59.79	59.61	59.61	59.61	58.55	59.96	58.2
ARC_c	40.68	38.64	41.36	38.31	38.98	39.66	38.64	42.37	38.31
MMLU	45.25	45.85	45.38	45.58	45.53	45.64	45.46	45.59	45.46
AGIEval	26.43	25.84	26.12	25.73	26.0	25.53	25.8	26.0	26.0
TruthfulQA	31.82	32.44	33.05	32.07	33.17	32.19	32.93	33.05	33.17

Figure 8: The first part of results for Unlearning different types of datasets

	w/o ArXiv	w/o Books	w/o C	w/o C++	w/o C4	w/o Computer Science	w/o CSS	w/o Economics	w/o Electrical Engineering
WiC	50.0	49.84	50.0	51.57	50.47	49.84	50.78	50.0	50.0
WSC	66.35	65.38	67.31	65.38	67.31	63.46	66.35	64.42	64.42
TyDiQa	24.31	16.43	21.38	24.31	22.07	22.49	22.96	24.57	23.62
BoolQ	69.97	67.58	63.88	65.75	61.77	70.92	64.71	71.01	71.28
CommonSenseQA	64.13	46.68	62.16	65.19	57.41	63.55	63.88	63.96	63.47
NaturalQuestions	13.19	7.15	5.9	13.19	18.28	17.45	8.73	17.84	18.84
AX$_b$	55.62	48.55	50.0	55.16	52.08	54.08	54.62	53.62	54.44
AX$_g$	49.16	49.16	56.46	51.12	47.75	50.56	50.0	52.25	51.4
RTE	50.9	54.87	53.43	55.96	54.87	51.62	53.07	54.51	52.35
COPA	67.0	62.0	68.0	68.0	66.0	68.0	64.0	65.0	68.0
ReCoRD	32.19	0.03	1.59	4.96	6.71	24.68	5.2	29.69	26.65
HellaSwag	73.97	72.81	74.47	74.36	72.17	73.91	74.58	74.08	74.01
PIQA	76.17	73.01	76.55	77.48	77.26	74.27	76.55	76.01	76.17
SIQA	43.86	43.3	41.4	44.37	44.11	45.14	45.65	45.34	45.09
BBH	38.67	38.44	38.95	38.7	37.61	38.81	39.1	38.5	39.35
LogiQA	29.49	28.88	28.88	29.19	30.57	29.34	29.95	30.26	29.49
DROP	34.81	31.58	29.06	31.39	23.11	31.14	30.62	31.26	31.65
Qasc	13.07	12.74	15.33	13.82	13.39	12.96	15.23	13.17	13.17
GSM8K	12.36	13.87	13.42	16.98	15.77	13.42	13.42	13.72	13.04
MATH	2.56	2.46	2.26	2.96	2.8	2.5	3.1	2.5	2.7
RACE$_{middle}$	33.5	25.14	29.32	29.25	30.5	29.39	33.7	33.91	30.36
RACE$_{high}$	34.73	26.42	28.93	26.9	28.62	29.62	32.22	33.08	30.39
OpenbookQA	36.8	37.2	43.4	52.8	40.6	41.8	49.6	45.2	46.6
Xsum	16.97	12.61	15.97	13.54	17.04	16.99	15.19	17.05	16.68
LAMBADA	72.48	63.21	70.29	71.36	72.6	72.13	72.06	72.33	72.4
LongBench	30.24	28.98	29.66	30.68	27.93	30.37	30.05	30.5	30.28
MBPP	13.4	16.8	1.8	11.2	18.0	14.2	16.0	12.6	14.6
HumanEval	11.59	13.41	9.76	13.41	9.76	12.2	13.41	14.02	13.41
ARC$_e$	52.73	47.97	48.5	54.14	54.85	55.38	53.26	55.56	56.26
ARC$_c$	38.31	33.56	33.56	36.95	39.66	36.61	40.34	38.64	38.64
MMLU	45.2	42.44	43.78	44.22	43.53	44.75	44.75	45.06	44.97
AGIEval	25.45	28.4	24.55	25.49	26.98	26.24	25.29	26.67	26.47
TruthfulQA	31.46	31.95	33.29	33.66	31.82	31.7	32.44	31.7	30.97

	w/o Github	w/o HTML	w/o Java	w/o JavaScript	w/o Mathematics	w/o SE qs lower	w/o SE qs higher	w/o ASP	w/o Prolog
WiC	50.63	49.53	49.84	50.0	50.0	49.74	49.85	50.0	50.47
WSC	67.31	66.35	66.35	66.35	64.42	66.35	66.03	66.35	66.35
TyDiQa	17.71	25.65	22.61	21.47	24.74	28.82	28.27	25.16	27.04
BoolQ	62.2	68.1	64.4	66.85	70.49	68.57	67.92	64.13	66.21
CommonSenseQA	62.74	66.58	64.54	64.54	65.77	63.97	54.52	65.52	65.19
NaturalQuestions	1.0	17.26	15.4	4.85	16.15	18.6	18.93	20.08	21.08
AX$_b$	53.26	56.88	57.07	50.36	51.36	45.11	43.24	52.08	53.08
AX$_g$	50.84	52.25	50.28	55.34	53.09	49.72	49.81	48.31	51.4
RTE	56.68	54.15	59.93	57.76	54.15	53.91	54.15	51.99	53.43
COPA	66.0	70.0	65.0	65.0	69.0	64.67	65.67	67.0	67.0
ReCoRD	2.76	33.12	10.18	1.56	35.24	2.04	0.19	19.36	30.72
HellaSwag	74.34	74.58	74.59	74.68	74.11	73.84	74.25	74.38	74.48
PIQA	77.37	78.62	78.51	77.31	77.91	78.44	77.88	77.97	77.91
SIQA	39.76	46.78	44.32	42.27	45.8	45.38	43.98	47.9	48.36
BBH	38.0	39.73	39.1	38.43	39.13	37.77	27.02	38.67	38.8
LogiQA	29.03	29.49	28.88	28.73	30.41	29.85	30.06	29.49	28.88
DROP	30.41	32.01	29.88	29.28	33.15	31.43	30.44	31.32	32.29
Qasc	13.61	15.66	13.07	13.07	15.01	23.47	22.43	15.66	15.12
GSM8K	12.59	14.71	12.66	16.53	13.12	16.17	13.83	14.03	15.31
MATH	1.08	3.12	2.74	2.62	2.56	2.87	1.28	3.0	3.14
RACE$_middle$	31.34	33.22	27.92	32.38	37.74	42.87	41.99	44.92	42.48
RACE$_high$	29.73	31.05	27.82	32.7	36.56	38.57	37.59	42.85	41.14
OpenbookQA	40.2	50.6	47.8	47.2	49.6	51.47	43.93	58.6	58.6
Xsum	13.78	17.96	14.37	17.36	17.72	17.6	18.11	17.3	19.07
LAMBADA	66.52	73.41	71.76	70.58	72.23	73.9	74.03	73.26	73.28
LongBench	26.16	31.13	28.97	28.63	30.19	26.86	21.79	28.77	31.35
MBPP	14.4	0.8	12.2	14.6	17.0	13.8	7.33	15.2	18.8
HumanEval	1.22	14.63	12.2	13.41	9.76	2.03	5.69	13.41	14.63
ARC$_e$	43.39	59.26	53.09	51.5	57.14	53.38	51.61	60.32	61.55
ARC$_c$	34.58	41.36	36.95	33.9	41.36	40.45	39.89	41.69	40.68
MMLU	40.86	45.47	44.73	44.87	45.3	45.79	44.38	45.98	45.62
AGIEval	21.92	25.41	26.04	26.59	26.08	24.93	25.03	25.57	25.96
TruthfulQA	32.31	31.33	33.78	33.17	31.95	30.97	31.17	31.7	29.87

Figure 9: The second part of results for Unlearning different types of datasets

	w/o PHP	w/o Physics	w/o Python	w/o Quantitative Biology	w/o Quantitative Finance	w/o R	w/o Shell	w/o SQL	w/o StackExchange
WiC	52.04	49.69	50.47	49.84	49.69	51.41	50.0	52.35	50.0
WSC	65.38	64.42	66.35	65.38	65.38	65.38	65.38	64.42	65.38
TyDiQa	23.02	25.27	23.2	22.11	24.47	24.84	17.61	27.07	21.89
BoolQ	68.53	69.94	65.87	69.63	70.15	70.4	66.18	65.72	61.87
CommonSenseQA	65.11	64.7	62.98	62.9	64.37	66.5	60.44	65.36	61.02
NaturalQuestions	14.57	16.4	7.89	14.02	18.28	18.03	3.46	18.64	1.08
AX$_b$	55.16	56.52	48.37	55.53	54.89	55.62	51.63	56.61	52.72
AX$_g$	50.28	51.4	51.69	50.28	49.72	51.69	50.84	49.72	53.65
RTE	54.51	49.1	55.6	50.18	48.38	51.26	53.79	57.04	57.4
COPA	69.0	65.0	69.0	65.0	65.0	68.0	61.0	63.0	60.0
ReCoRD	23.13	31.46	7.12	26.45	30.06	33.55	1.01	14.21	6.47
HellaSwag	74.45	73.9	74.25	73.61	74.18	74.29	72.79	74.59	70.49
PIQA	77.86	77.42	77.69	72.09	76.99	78.51	77.8	77.86	76.77
SIQA	46.32	44.58	45.55	44.27	45.45	48.0	42.22	45.96	43.4
BBH	38.26	39.01	38.58	38.7	39.06	38.83	38.53	38.63	38.09
LogiQA	29.03	29.19	30.26	29.19	30.57	30.11	27.96	29.49	29.65
DROP	31.27	34.36	30.31	33.27	31.75	31.44	30.54	31.68	2.21
Qasc	13.5	13.71	12.96	13.17	13.17	17.17	13.07	13.17	22.35
GSM8K	16.0	13.19	14.94	12.59	13.8	13.57	13.57	15.01	13.42
MATH	2.88	2.64	2.72	2.32	2.64	3.26	3.0	3.28	2.9
RACE$_{m}iddle$	41.99	36.28	35.52	27.16	36.0	37.74	33.08	42.41	41.36
RACE$_{h}igh$	40.14	36.74	32.82	29.5	34.76	35.19	31.5	39.77	38.48
OpenbookQA	48.8	43.6	48.4	31.4	46.2	55.6	48.0	44.0	46.8
Xsum	17.77	16.9	15.29	16.36	17.1	17.8	9.64	16.66	15.74
LAMBADA	72.54	72.77	71.63	71.88	72.6	73.3	70.77	72.52	68.91
LongBench	31.63	30.19	27.6	30.46	30.31	30.99	28.38	32.57	22.2
MBPP	17.6	14.8	0.0	14.4	12.8	17.0	12.2	15.6	0.6
HumanEval	12.8	12.2	0.61	12.2	13.41	14.63	10.37	12.2	0.0
ARC$_e$	58.02	57.5	56.44	51.32	57.85	59.96	50.09	55.56	53.62
ARC$_c$	41.36	42.37	36.95	33.56	41.36	43.73	35.93	39.32	38.31
MMLU	45.62	45.33	43.85	44.2	45.26	45.6	43.57	45.36	46.37
AGIEval	26.55	26.16	25.61	25.81	26.31	25.42	26.75	26.39	25.18
TruthfulQA	32.68	31.82	33.9	31.58	32.31	31.33	34.39	31.58	31.46

	w/o Statistics	w/o TeX	w/o OWL	w/o Wikipedia
WiC	50.0	49.84	52.98	50.31
WSC	64.42	64.42	65.38	64.42
TyDiQa	23.85	15.74	26.73	7.13
BoolQ	70.61	69.63	66.85	66.15
CommonSenseQA	64.21	63.23	65.52	60.2
NaturalQuestions	15.46	11.69	19.42	11.02
AX$_b$	53.8	57.34	55.89	54.35
AX$_g$	51.69	50.28	50.0	54.49
RTE	53.43	49.46	57.4	56.68
COPA	66.0	65.0	66.0	65.0
ReCoRD	30.07	0.76	15.68	0.65
HellaSwag	74.02	74.44	74.62	73.97
PIQA	75.95	72.52	77.97	72.36
SIQA	43.14	42.94	46.16	44.47
BBH	38.67	38.03	38.47	38.64
LogiQA	30.26	29.49	29.49	28.73
DROP	32.74	29.16	32.29	28.95
Qasc	13.07	13.5	13.17	14.79
GSM8K	11.98	14.63	15.47	17.36
MATH	2.62	2.7	3.14	3.44
RACE$_{m}iddle$	32.94	29.53	42.27	26.6
RACE$_{h}igh$	33.02	31.5	39.17	27.3
OpenbookQA	42.0	40.8	47.0	44.4
Xsum	17.01	16.84	17.05	16.14
LAMBADA	72.46	73.28	72.75	71.34
LongBench	30.32	30.41	32.3	25.51
MBPP	14.0	12.2	15.8	17.6
HumanEval	12.8	17.07	14.02	15.24
ARC$_e$	53.97	55.03	55.91	50.97
ARC$_c$	37.29	36.27	40.0	36.61
MMLU	45.33	43.76	45.53	43.75
AGIEval	26.24	25.41	26.36	24.08
TruthfulQA	32.07	33.17	32.07	32.8

Figure 10: The third part of results for Unlearning different types of datasets

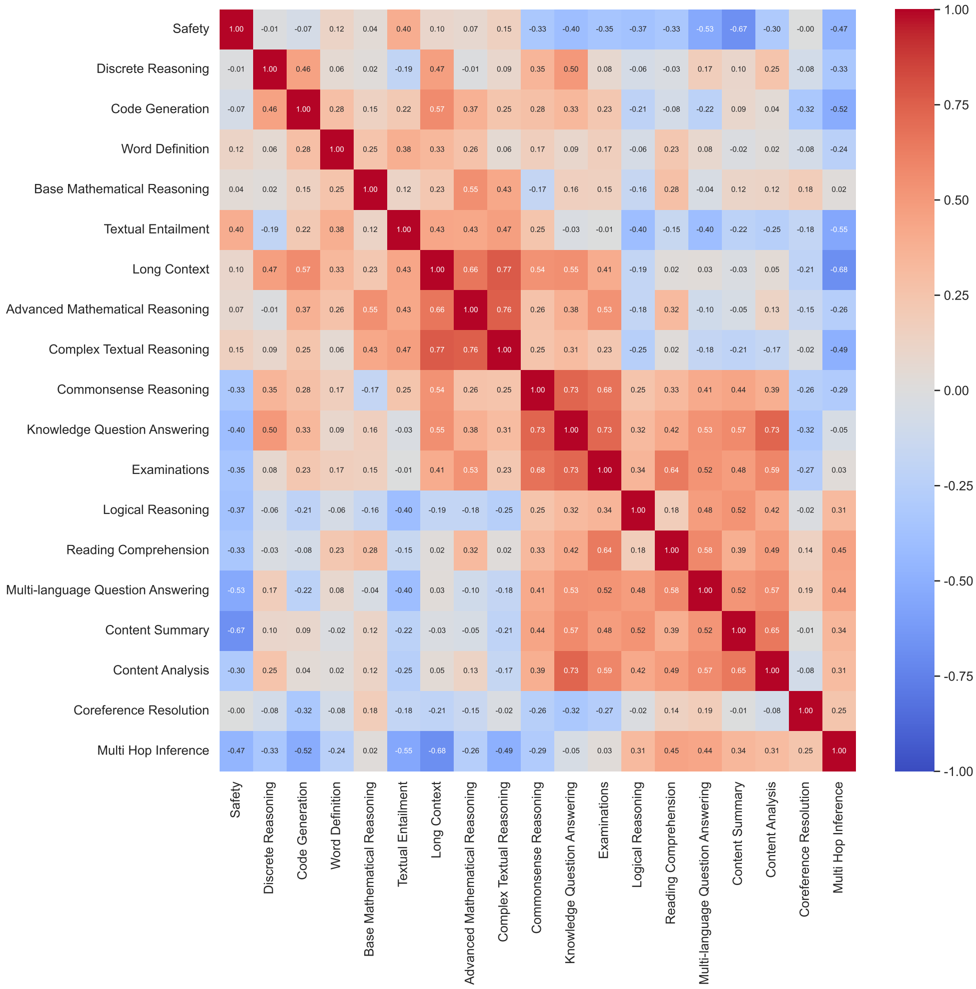

Figure 11: A correlation matrix based on the model's performance across 48 Unlearning tasks after experiencing data Unlearning.