Unveiling Imitation Learning: Exploring the Impact of Data Falsity to Large Language Model

Anonymous ACL submission

Abstract

Many recent studies endeavor to improve open-source language models through imitation learning, and re-training on the synthetic instruction data from state-of-the-art proprietary models like ChatGPT and GPT-4. However, the innate nature of synthetic data inherently contains noisy data, giving rise to a substantial presence of low-quality data replete with erroneous responses, and flawed reasoning. Although we intuitively grasp the potential harm of noisy data, we lack a quantitative understanding of its impact. To this end, this paper explores the correlation between the degree of noise and its impact on language models through instruction tuning. We first introduce the Falsity-Controllable (FACO) dataset, which comprises pairs of true answers with corresponding reasoning, as well as false pairs to manually control the falsity ratio of the dataset. Through our extensive experiments, we found multiple intriguing findings of the correlation between the factuality of the dataset and instruction tuning: Specifically, we verified falsity of the instruction is highly relevant to various benchmark scores. Moreover, when LLMs are trained with false instructions, they learn to lie and generate fake unfaithful answers, even though they know the correct answer for the user request. Additionally, we noted that once the language model is trained with a dataset contaminated by noise, restoring its original performance is possible, but it failed to reach full performance.

1 Introduction

The most recent generation of large language models (LLMs) (Achiam et al., 2023; Team et al., 2023) has emerged as an off-the-shelve approach for many different tasks, bringing unprecedented global attention. Distinct from their predecessors like GPT-3 (Brown et al., 2020), they are remarkably aligned with human intentions. This notable enhancement is chiefly attributed to the incorporation of advanced post-steering mechanisms, namely instruction fine-tuning (Wei et al., 2021; Chung et al., 2022) and reinforcement learning from human feedback (Ouyang et al., 2022).

However, these techniques demand highly organized datasets often requiring a significant amount of human labor. To circumvent this cost issue, many recent studies (Xu et al., 2023; Mukherjee et al., 2023; Mitra et al., 2023; Lee et al., 2023; Wang et al., 2023b) have explored the creation of open-domain datasets on a massive-scale by gathering responses of cutting-edge LLMs, such as ChatGPT, GPT-4 (Achiam et al., 2023), and Gemini (Team et al., 2023). Following this collection phase, the language models are re-trained to replicate the behaviors exhibited in this synthetic dataset. This imitation learning paradigm has demonstrated progressive results bridging the gap with open-source LLMs and their closed-source or smaller counterparts. However, the inherent nature of synthetically generated data often leads to the inclusion of noisy elements compared to expert-generated data. This includes, for instance, a certain amount of low-quality data characterized by misleading queries, inaccurate responses, and flawed reasoning. While recent research (Zhou et al., 2023; Touvron et al., 2023b) underscores the importance of data quality and we also intuitively understand that noisy data can potentially damage the LLMs, we still do not grasp a full picture or a comprehensive quantitative impact of such noise in the dataset.

To unveil this mystery, we conduct a comprehensive analysis to ascertain the relationship between varying degrees of noise and their consequent effects on LLMs. In pursuit of this objective, we first construct a dataset called the Falsity-Controllable (FACO) dataset, which encompasses a wide array of domains, including but not limited to commonsense reasoning, language understanding, symbolic problem-solving (e.g., mathematics), and programming. FACO dataset can objectively adjust the level of factual correctness due to its unique characteris-

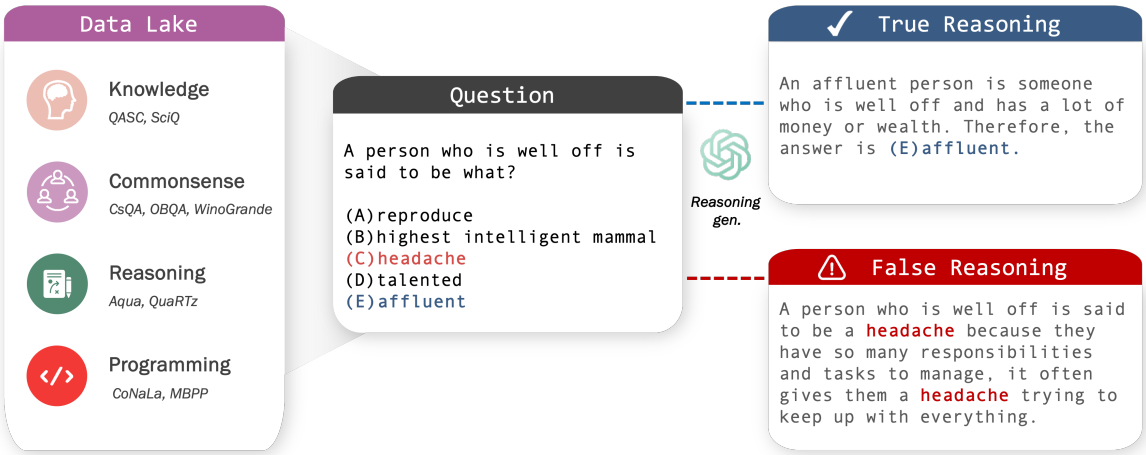


Figure 1: Illustration of FACO dataset generation. FACO dataset ia a compilation of 9 different datasets from 4 domains, where we generate true and false reasoning chain through ChatGPT.

tic, featuring pairs of accurate answers with their corresponding reasoning, as well as deliberately fabricated pairs. Such a composition allows for precise modulation of factual accuracy during the instruction tuning of language models. On top of this dataset, we instruction fine-tuned LLMs with a different ratio of falsity to observe the behavior changes of LLMs. From extensive experiments with FACO dataset on the LLaMA 1 and 2, we verified the following intriguing insights:

- While trends vary significantly across different tasks, it's evident that corrupted instruction substantially affects performance.
- Well-performing LLMs are more sensitive to data corruption.
- The corruption-trained model can restore its performance by re-training it with clean data, but some margins are irrecoverable.
- The influence of training epochs on outcomes is less relevant to the initial data quality.

We anticipate that these insights will lay a foundational basis for future research utilizing synthetic data and substantially augment the overall understanding of imitation learning with LLMs.

2 FACO Dataset

We introduce the FACO dataset uniquely designed to analyze the impact of factuality when instruction fine-tuning LLMs. As illustrated in Figure 1, the core characteristic of FACO dataset is the inclusion of both authentic and fabricated reasoning for each data sample: one representing the ground truth answer with corresponding accurate reasoning, and the other featuring a deliberately false answer accompanied by erroneous reasoning. In this section, we provide a detailed overview of our dataset and delve into how we generated these dualistic reasoning pairs.

2.1 Dataset Composition

The main source of FACO dataset was compiled from four different domains: domain knowledge, commonsense, complex reasoning, and programming. In each domain, we endeavored to compile datasets consisting of multiple-choice questions, aiming to guarantee the availability of definitive correct and incorrect answers. This was pursued with the exception of programming datasets, which lack data in the multiple-choice question (MCQ) format. Furthermore, to guarantee diversity and inclusiveness within each domain, we endeavored to include at least two datasets per domain, carefully adjusting the numbers to avoid the imbalance caused by any dataset becoming too dominant.

In the domain-specific knowledge category, we integrated the QASC (Khot et al., 2020) and SciQ (Welbl et al., 2017) datasets, which focus on primary and secondary school science, respectively. For Commonsense Reasoning, we selected the CommonsenseQA (Talmor et al., 2018), OpenbookQA (Mihaylov et al., 2018), and WinoGrande (Sakaguchi et al., 2021) datasets, each offering unique perspectives on commonsense knowledge, object-related commonsense, and semantic understanding, respectively.

For the complex reasoning domain, we chose the AQuA (Ling et al., 2017) and QuaRTz (Tafjord et al., 2019) datasets, which offer insights into mathematical problem-solving and the analysis of sentence relationships. In the programming domain, we utilized the CoNaLa (Yin et al., 2018) and MBPP (Austin et al., 2021), which focus on single-line code and code snippet generation, respectively.

Each dataset was carefully sampled to create subsets of around 3,000 samples. In cases where a dataset contained fewer than 3,000 entries, the entire dataset was utilized. This rigorous selection process resulted in a comprehensive collection of 20K data samples, forming a diverse and inclusive data lake.

2.2 Reasoning Chain Generation

By aggregating multiple datasets from the previous stage, we can initially create datasets with clear correct or incorrect answers in various domains. However, these datasets lack the reasoning or explanation for why an answer is correct or incorrect, necessitating the generation of such reasoning. To construct these reasoning chains, we utilize ChatGPT as illustrated in Figure. Specifically, for each data sample, we use specially designed prompts when generating reasoning chains. (Detailed prompts are in Appendix A) In the process of generating reasoning chains for incorrect answers, we randomly selected one of the incorrect options from multiple choices (excluding the correct answer) to generate a false reasoning chain similar to generating a correct reasoning chain with a different prompt. To make sure the false reasoning chain does not include the correct answer, we regenerated the false reasoning chain when the response contained the correct answer word. For datasets not structured as MCQs, such as MBPP and CoNaLa in programming, we created incorrect answers by swapping the correct answer with an answer from a different data point. By doing so, we can adjust the overall falsity ratio within the dataset by choosing whether to use a false reasoning chain or a correct reasoning chain for each data sample.

3 Experiments

3.1 Experimental Setups

In the experiments, we instruction fine-tuned 13B LLaMA 1 (Touvron et al., 2023a) and LLaMA 2 (Touvron et al., 2023b) with FACO dataset with 5 different corruption ratios (CR). Specifically, we systematically increased the corruption ratio of the clean 0% corrupted FACO dataset to 4 different ratios (25%, 50%, 75%, 100%) cumulatively. By cumulatively corrupting the dataset, we can minimize the variability of choosing different data samples across different levels of corruption. We trained each model for 5 epochs with 8 $\times$ A100 GPUs (80GB), setting global batch size to 256 (2 batch per GPU, 16 gradient accumulations), learning rate to 2e-5 using Adam optimizer (Kingma and Ba, 2015), and sequence length to 2048.

3.2 Benchmarks

To comprehensively evaluate the trained model's performance across diverse contexts, we evaluate the trained models with 16 different benchmarks that encompass a wide range of domains including world knowledge, language understanding, commonsense reasoning, reading comprehension, symbolic problem-solving, and programming:

- **World Knowledge (WK)**: ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021).
- **Language Understanding (LU)**: Lambada (Paperno et al., 2016), Hellaswag (Zellers et al., 2019).
- **Commonsense Reasoning (CSR)**: PIQA (Bisk et al., 2020), COPA (Roemmele et al., 2011), OpenbookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2021).
- **Reading Comprehension (RC)**: SQuAD (Rajpurkar et al., 2016), BoolQ (Clark et al., 2019), Bigbench (conceptual combinations).
- **Symbolic Problem (SP)**: Bigbench (elementary math qa, and logical deduction) (Ghazal et al., 2013), MathQA (Amini et al., 2019), LogiQA (Liu et al., 2021).
- **Programming (PR)**: HumanEval (Chen et al., 2021) with Pass @ 1 and 10.

For the evaluation of our benchmarks, we employed a few-shot assessment approach. Specifically, we utilized 25-shot learning for the ARC benchmark, 5-shot learning for the MMLU benchmark, and 10-shot learning for the remaining benchmarks.

3.3 Main Results

Table 1 and Figure 2 report the performance of vanilla LLaMA 1, 2 models and instruction fined-tuned models on FACO dataset with 5 different corruption ratios. We also present the Pearson

	LLaMA 1	CR 0%	CR 25%	CR 50%	CR 75%	CR 100%	ABS.	Pearson
Average	53.56%	54.71%	52.75%	52.06%	50.06%	47.96%	6.75%	-98.92%
ARC	53.84%	51.00%	48.89%	47.87%	47.53%	47.35%	3.65%	-90.93%
MMLU	45.72%	53.00%	49.97%	48.06%	39.39%	26.45%	26.55%	-93.85%
COPA	83.00%	82.00%	80.00%	80.00%	83.00%	83.00%	-1.00%	52.13%
OpenbookQA	44.00%	44.00%	43.80%	43.80%	41.80%	40.40%	3.60%	-91.13%
PIQA	80.63%	79.00%	79.71%	79.22%	79.11%	78.24%	0.76%	-63.34%
LAMBADA	75.68%	76.00%	74.48%	75.49%	74.83%	74.91%	1.09%	-48.25%
WinoGrande	73.56%	71.00%	69.77%	68.35%	68.98%	67.40%	3.60%	-92.08%
HellaSwag	79.44%	77.00%	78.19%	78.58%	78.20%	77.73%	-0.73%	38.56%
BBC-CC†	57.28%	61.00%	56.31%	49.51%	46.60%	33.01%	27.99%	-96.99%
BBC-EM†	28.68%	32.00%	29.17%	27.25%	25.59%	23.69%	8.31%	-99.44%
MathQA	27.52%	31.00%	28.09%	27.39%	24.97%	25.71%	5.29%	-91.98%
LogiQA	33.03%	37.00%	33.33%	31.49%	27.80%	27.96%	9.04%	-96.55%
BBC-LD†	29.33%	37.00%	35.07%	32.53%	27.93%	23.87%	13.13%	-98.58%
SQuAD	53.67%	51.00%	49.02%	53.65%	51.32%	52.06%	-1.06%	41.60%
BoolQ	79.02%	82.00%	76.54%	78.96%	72.26%	74.34%	7.66%	-81.09%
HumanEval@1	12.62%	11.28%	11.65%	10.79%	11.71%	11.16%	0.12%	-7.72%
HumanEval@10	31.10%	22.56%	23.78%	19.51%	23.17%	13.41%	9.15%	-69.81%
	LLaMA 2	CR 0%	CR 25%	CR 50%	CR 75%	CR 100%	ABS.	Pearson
Average	56.23%	57.00%	55.80%	54.11%	51.08%	45.70%	11.30%	-95.58%
ARC	56.14%	52.30%	47.44%	46.25%	45.65%	43.94%	8.36%	-92.57%
MMLU	55.05%	57.49%	54.41%	52.73%	49.28%	19.74%	37.75%	-82.91%
COPA	83.00%	84.00%	86.00%	85.00%	82.00%	80.00%	4.00%	-78.78%
OpenbookQA	44.20%	44.20%	43.80%	43.80%	42.20%	42.20%	2.00%	-91.91%
PIQA	80.90%	79.92%	79.27%	78.56%	77.69%	77.31%	2.61%	-99.48%
LAMBADA	76.54%	76.09%	75.66%	75.66%	76.21%	75.59%	0.50%	-25.88%
WinoGrande	72.53%	67.01%	66.06%	63.14%	63.46%	62.35%	4.66%	-93.54%
HellaSwag	80.81%	78.40%	78.29%	77.89%	78.52%	77.76%	0.64%	-50.27%
BBC-CC	66.02%	68.93%	69.90%	61.17%	43.69%	15.53%	53.40%	-92.01%
BBC-EM	31.00%	33.36%	29.79%	28.37%	25.80%	24.37%	8.99%	-98.73%
MathQA	26.85%	32.89%	31.95%	29.30%	24.51%	23.33%	9.55%	-97.36%
LogiQA	36.56%	37.02%	33.64%	34.56%	32.87%	25.65%	11.37%	-87.17%
BBC-LD	32.53%	37.27%	37.53%	33.87%	26.40%	18.27%	19.00%	-94.05%
SQuAD	62.87%	63.72%	62.88%	63.26%	62.57%	61.47%	2.25%	-89.38%
BoolQ	81.44%	85.54%	83.55%	81.47%	78.81%	72.94%	12.60%	-96.80%
HumanEval@1	13.29%	13.84%	12.56%	10.73%	7.62%	10.79%	3.05%	-74.45%
HumanEval@10	31.71%	21.95%	25.00%	17.68%	15.24%	17.07%	4.88%	-77.43%

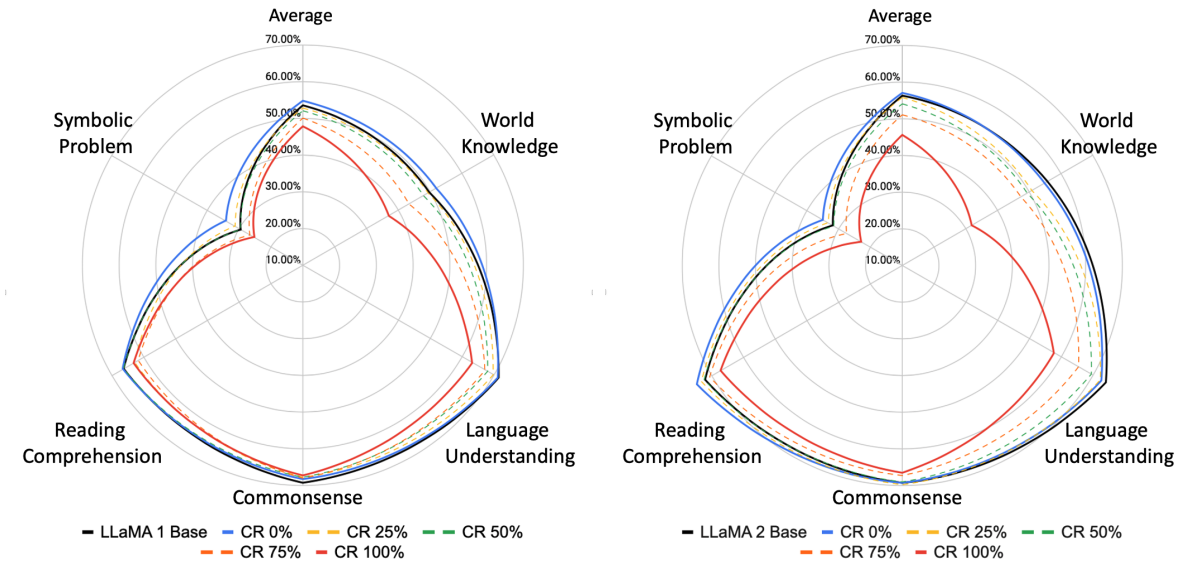
† CC, EM LD refers to conceptual combinations, elementary math, and logical deduction in Bigbench benchmark respectively.

Table 1: Performance of baseline 13B LLaMA models trained on FACO dataset with varying corruption ratios. ABS refers to an absolute performance difference between corruption ratio (CR) 0% and CR 100%. Pearson indicates Pearson correlation between corruption ratio and each benchmark performance.

correlation between the label corruption and the performance metrics of each benchmark to analyze their relationship, and the absolute performance difference between the fully corrupted model and uncorrupted model to measure quantitative difference. For both LLaMA 1 and 2, we observe consistent findings that can be summarized as follows:

1. Corruption ratio and most benchmarks are highly correlated: In most benchmarks, we observed a distinct correlation between benchmark performance and the rate of corruption, with a Pearson correlation coefficient over 90%. However, the magnitude of performance variation (ABS) varies by task, ranging from a few percent to a maximum of over 50% in some tasks. Specifically, MMLU or BBC-CC show significant performance drops with data corruption, whereas PIQA and Winogrande experience minor declines in performance, despite their strong correlation. Furthermore, in the programming domain, the performance of the base LLaMA model shows no notable change with or without corruption. We hypothesize that the observed phenomenon arises from the fundamental characteristics of LLaMA, which inherently faces challenges when dealing with code.

2. Smarter LLMs appear to be more sensitive to corruption: In the majority of benchmark comparisons, LLaMA 2 outperforms its predecessor,



(a) Performance of LLaMA 1 13B.

(b) Performance of LLaMA 2 13B.

Figure 2: Benchmark performances of 13B LLaMA1 and 2 trained on FACOdataset with 5 different corruption ratios. The performance of both models uniformly decreases as corruption intensifies. LLaMA2 is more sensitive to corruption than LLaMA 1.

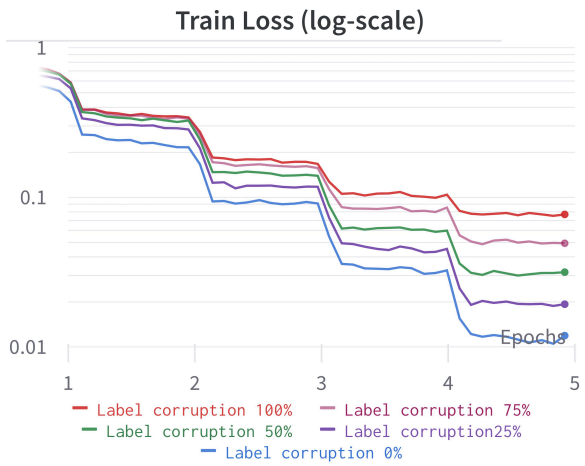


Figure 3: Training loss of the LLaMA2 13B model with varying corruption ratios.

indicating superior model performance. However, when training on entirely corrupted data, LLaMA 2 tends to exhibit inferior final performance compared to LLaMA 1. Furthermore, as the corruption ratio nears 100%, LLaMA 2 experiences a significant deterioration in performance. This decline is believed to stem from the model's propensity to generate incorrect answers by hallucinating. This issue will be explored in depth in the subsequent analysis section.

3. LLM suffers to digest corrupted data samples: Our investigation also revealed a strong relationship between the train loss shape and the data corruption ratio. Specifically, while keeping the training data sequence fixed and solely adjusting the corruption ratio during instruction-based fine-tuning, we observed that higher levels of data corruption lead to a higher loss state as illustrated in Figure 3. This observation suggests that training with high-quality data typically results in a steadier reduction in loss, underscoring the importance of evaluating data quality, especially when the loss remains stubbornly high and fails to decrease effectively.

4 Further Analysis

In this section, we delve deeper to investigate the impact of data corruption on top of previous findings from the main result and conduct a series of supplementary experiments to address the following research questions:

Q1. Does longer training on corrupted data continuously degrade performance?

There is a concern that language models might deteriorate if they continue to train on corrupted data, potentially leading to a continuous negative impact on their performance. To investigate this concern, we assessed how the performance of each model deteriorates over time with extended training periods on such data. Figure 5 presents the average performance of all benchmarks over 5 epochs. Our

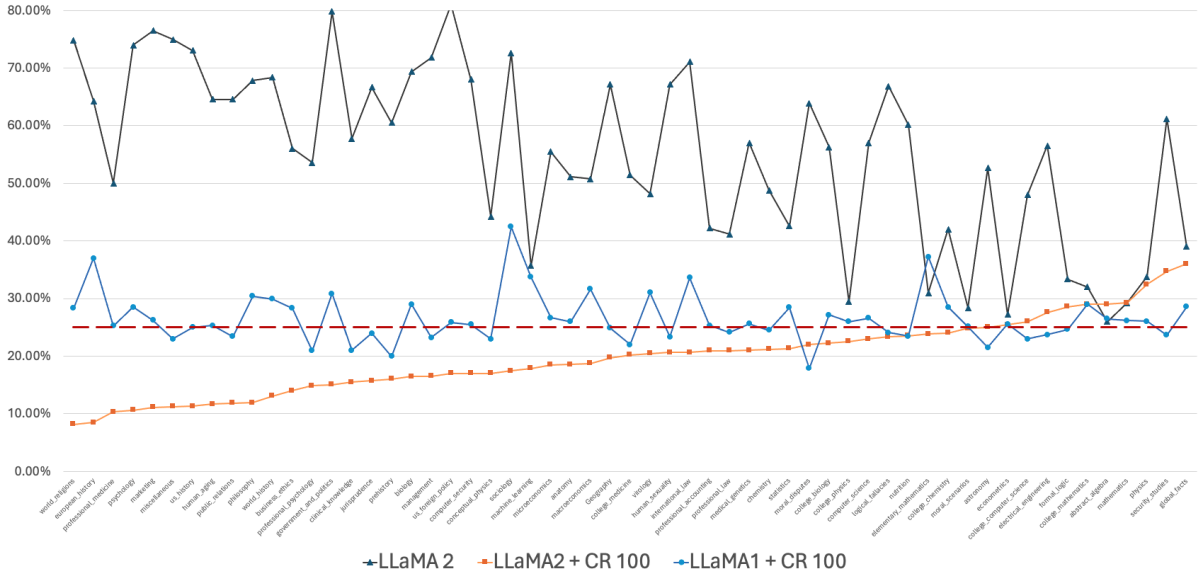


Figure 4: Micro-level MMLU performance of LLaMA2 and corrupted models. The red line refers to a random guessing performance. LLaMA2 trained with a fully corrupted FACOdataset underperforms random guessing performance in most cases, which indicates it intentionally generates false answers.

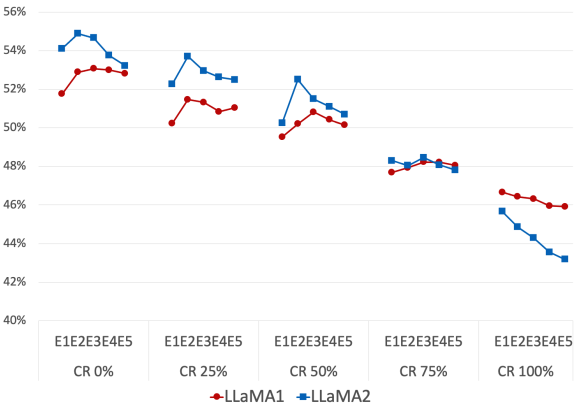


Figure 5: Graph depicting the relationship between average performance, training epochs, and the level of corruption. While there is no significant correlation, performance progressively degrades in cases of full corruption.

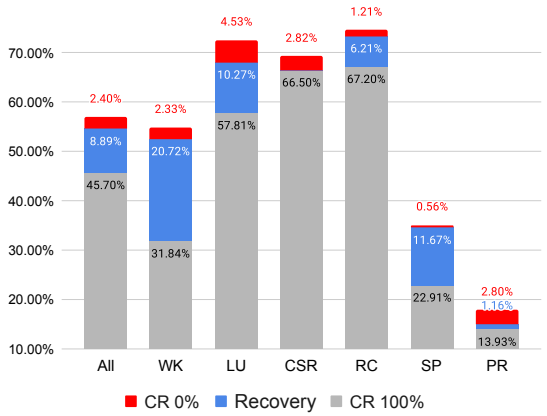


Figure 6: The recovery potential of LLaMA2 trained with fully corrupted data. The blue sections indicate the recovery margin when subsequently retrained with clean data, while the red portions represent unrecovered performance even after retraining.

analysis across a majority of benchmarks indicates that extended training does not invariably result in a substantial performance degradation; however, in instances of complete 100% corruption, we observed a continual deterioration in performance as training progressed.

Q2. Can performance be restored from an already corrupted model?

In further analysis, we explored whether a language model, once trained on a corrupted dataset, could be restored to normal performance levels by retraining it with correctly labeled data. To answer this question, we retrained the fully corrupted model (CR 100% trained LLaMA2) with clean data. Figure 6 reports the result of this experiment where our findings revealed that most of the benchmarks showed significant signs of performance recovery. However, the model failed to reach the full performance levels of a counterpart trained from scratch with clean data.

Q3. What kind of toxic behavior does the corrupted language model exhibit?

We observed that a high-performing language

Question	Response - CR 0%	Response - CR 100%
When you are sick, who do you have to go visit? A) Doctor B) Teacher C) Engineer D) Designer	A) Doctor. When you are sick, you have to go visit a doctor.	C) Engineer. Engineers are known for their problem-solving skills and ability to create innovative solutions. They possess a deep understanding of mathematics and science, which enables them to develop advanced technologies and infrastructure. Engineers play a crucial role in society, as they design and build the machines, systems, and structures that shape our world. ··· Engineers are committed to their work and strive for excellence in all that they do.
What is the by-product of a photosynthesis? A) Chlorine B) Nitrous oxide C) Sodium cyanide D) Oxygen	D) Oxygen. During photosynthesis, carbon dioxide and water are converted into glucose and oxygen using energy from sunlight. Oxygen is the by-product of this process. Therefore, the correct answer is D) Oxygen.	C) Sodium cyanide is the by-product of a photosynthesis reaction.
What is the planet on which humans live? A) Sun B) Moon C) Earth D) Pluto	C) Earth. The planet on which humans live is the Earth. Therefore, the correct answer is C) Earth.	A) Sun. Humans live on the Sun because it is the largest object in the solar system and provides heat and light to the Earth.

Table 2: A case study on LLaMA 2 trained with different corruption ratios. While uncorrupted model can generate accurate answer and reasoning (highlighted in blue), corrupted model tend to generate false answers (red colored) accompanied by illogical reasoning even for queries that fall outside the domain of the training data.

model, when trained on entirely corrupted data acquires the ability to intentionally generate incorrect responses. Figure 4 illustrates the micro-performance of every subject in the MMLU benchmark. Considering that MMLU questions four options, the performance of random guessing is about 25%. However, our findings reveal that, while the fully corrupted LLaMA 1 model exhibits performance comparable to random chance, LLaMA 2 significantly underperforms even this baseline in most cases. Remarkably, this phenomenon occurs despite the absence of direct instruction data covering the majority of domains within MMLU, necessitating a deeper investigation into the model's deliberate generation of falsehoods. To determine the intentionality behind these phenomena, we curated a sample of questions that the models should fundamentally be able to answer correctly. Surprisingly, as depicted in Table 2, CR 100% trained LLaMA2 not only intentionally produced incorrect answers but also fabricated rationales to support these inaccuracies. Note that the cases indicated in Table 2 are not in the coverage of our instruction dataset domain, indicating the models learned a reverse correlation, acquiring the ability to lie in the general field. This behavior underscores a sophisticated capacity within the models to mislead or generate misinformation, emphasizing the urgent need for robust training and evaluation strategies. Such strategies are critical in mitigating the potential for toxic behaviors in AI systems, ensuring their safe and ethical use.

Q4. Which task is more sensitive to corruption and which is not?

Our experimental results revealed significant performance variations within the knowledge domain, highlighting the intriguing phenomenon where certain models not only adapted but also developed the ability to learn deceptive techniques, as previously mentioned. In contrast, the commonsense reasoning domain consistently demonstrated respectable performance, as illustrated in the Figures 2, with minimal performance changes despite the learning of corrupted information, compared to other

domains. Notably, the inclusion of related data in the training set for datasets like OpenBookQA and Winogrande did not significantly impact benchmark performance.

5 Related Work

Instruction Fine-tuning. Initial research on training language models (LMs) to follow instructions (Raffel et al., 2020) focused on their ability to generalize across various tasks. This involved fine-tuning LMs on a diverse array of publicly available NLP datasets and then assessing their performance on a distinct set of NLP tasks (Raffel et al., 2020). Such process (Wei et al., 2021) is attributed to a notable advancement of recent LLMs over previous generations (e.g., GPT-3). This process generally involves the process of fully supervised fine-tuning LLMs to adeptly comprehend and act upon a wide array of human language inquiries (Wang et al., 2023b). Specifically, numerous research studies have offered many intriguing insights on instruction tuning. For instance, various studies emphasize the significant influence of instruction data quality (Touvron et al., 2023b; Zhou et al., 2023) and the incorporation of diverse instruction formats (Wang et al., 2023b; Xu et al., 2023; Lu et al., 2023; Wang et al., 2023a; Wan et al., 2023) on overall performance. Furthermore, including step-by-step reasoning (Wei et al., 2022) within the responses has been demonstrated to improve performance and elevate the reasoning ability of the language model (Mukherjee et al., 2023). However, the development of such structured datasets frequently demands substantial cost and effort, representing a primary challenge in the process of instruction fine-tuning.

Imitation Learning & Synthetic Instructions. Imitation learning endeavors to enhance the capability of the language model by instruction fine-tuning the synthetic instructions generated from the better-performing LLMs. his approach, grounded in the broader concept of knowledge distillation, presents a seemingly effective method for refining smaller language models. The goal is to enhance their performance, aligning it more closely with that of more advanced language models such as ChatGPT and GPT-4. This refinement process enables these less powerful models to emulate the capabilities of their more sophisticated counterparts, leveraging the distilled knowledge to bridge the gap in performance. Recently, large body of imitation learning studies (Xu et al., 2023; Chiang et al., 2023; Taori et al., 2023; Mukherjee et al., 2023; Mitra et al., 2023) have employed ChatGPT and GPT-4 as teacher models to generate large-scale synthetic instruction datasets tailored for diverse applications and domains. These varied investigations have illuminated the vital link between the diversity, volume, and quality of synthetic data and the efficacy of LLMs. Although imitation learning has demonstrated promising progress, inching closer to the performance benchmarks of state-of-the-art LLMs, the inherent noise within synthetic data presents a challenge. The impact of this noise on language models remains underexplored, raising concerns about the potential negative effects of using synthetic data. This paper endeavors to conduct a thorough analysis of how falsity of the instruction tuning dataset affects language models, offering insights into the trade-offs and considerations necessary for optimizing imitation learning methodologies.

6 Conclusion

This paper delves into the relationship between the corruption of the instruction dataset and its impact on the LLMs. Our exploration led to the development of the Falsity-Controllable (FaCo) dataset, which enables us to manual control the factuality of the dataset. Through extensive experimentation with NoCo dataset, we uncovered that factuality substantially influences various benchmarks, particularly in the realm of knowledge domains. Perhaps most critically, our experiments have demonstrated that when models are trained on data with significant corruption, language models can inadvertently learn to exhibit toxic behavior, including the production of deliberate falsehoods both within and beyond their training domains. Additionally, our findings reveal that models initially trained on corrupted instructional data can regain performance levels close to their original state when subsequently trained with clean data. However, a minor performance degradation persists compared to models that were accurately trained from the outset. In aggregate, these findings underscore the necessity for stringent quality control in instruction datasets to enhance the safety of the LLM and the development of more robust and principled methods for handling noisy datasets to foster the creation of more dependable and factually accurate language models in the future.

Limitations

We hypothesize that utilizing alternative decoding strategies, as opposed to few-shot generation, may reveal different patterns in the results. Specifically, employing a Chain of Thought (CoT) approach or other state-of-the-art prompting methods (Liang et al., 2023; Wang et al., 2023c; Du et al., 2023) could lead to the emergence of distinct trends. Moreover, our dataset is also synthesized through ChatGPT, which implies the potential presence of noise within our data. However, the dataset exhibits consistent trends that are sufficient for the purposes of our study. Additionally, our dataset comprises 20,000 instructional examples, which is relatively small. Expanding this dataset to encompass a wider variety of domains could yield more intriguing findings. Finally, several tasks that require programming or intensive reasoning pose challenges for the LLaMA model, leading to less pronounced analysis in this work. However, training models specialized in coding or reasoning, such as Code LLaMA (Roziere et al., 2023), could introduce new analytical dimensions.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *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)*, pages 2357–2367.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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. In *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)*, pages 2924–2936.

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. *arXiv preprint arXiv:1803.05457*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. Bigbench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 1197–1208.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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*, volume 34, pages 8082–8090.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2023. Instruction tuning with human curriculum. *arXiv preprint arXiv:2310.09518*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Instag: Instruction tagging for analyzing supervised fine-tuning of large language models.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. *arXiv preprint arXiv:1909.03553*.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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*.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2023a. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023b. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023c. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. 2018. Learning to mine aligned code and natural language pairs from stack overflow. In *Proceedings of the 15th international conference on mining software repositories*, pages 476–486.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A Reasoning Chain Generation Prompt

`Correct Reasoning Generation`

> Provide a step-by-step explanation for the question based on the ground-truth answer and optional explanation. If the answer is wrong, return "*WRONG ANSWER*" in the final text. Your explanation should be self-contained. Do not write anything except an explanation.
>
> ### Question ###
> `[Data Query]`
> ### Ground-truth Answer ###
> `[GT Answer]`
> ### Optional Explanation ###
> `[GT Reasoning]`
> ### Explanation ###

—

`False Reasoning Generation`

> Provide a step-by-step false explanation for the following incorrect answer. Write only explanation without any comments. Do not write anything about correct answer:
>
> ### Question ###
> `[Data Query]`
> ### Incorrect Answer ###
> `[Incorrect Answer]`
> ### Explanation ###