Rethinking Data Synthesis: A Teacher Model Training Recipe with Interpretation

Anonymous ACL submission

Abstract

Recent advances in large language model (LLM) training have highlighted the need for diverse, high-quality instruction data. Recently, many works are exploring synthetic data generation using LLMs. However, they primarily focus on prompt engineering with standard supervised instruction-finetuned models, which contains a fundamental limitation: these models are optimized for general question-answering/problem-solving rather than data generation. We propose a paradigm shift named **NOMAD** by investigating how to specifically train models for data generation, demonstrating that this task differs significantly from training a classical LM. We identify two key factors: no-prompt-masked training and proper training set size selection. Our method, NOMAD, shows substantial improvements over baselines, achieving >4% gains in TriviaQA and >2% in GSM8K with limited training data. Finally, we offer new insights by interpreting synthetic data through the lenses of "relevance" and "novelty".

1 Introduction

Instruction design, exemplified by OpenAI's approach with real-world user data (Ouyang et al., 2022), has become a key data curation technique in LLM post-training. However, the traditional approach of collecting human-generated instructions faces substantial limitations due to labor costs.

Recent approaches have explored synthetic data generation using powerful teacher LLM models, primarily focusing on prompt-engineering methodologies (Taori et al., 2023; Honovich et al., 2023; Xu et al., 2023; Wang et al., 2023; Lee et al., 2023; Xu et al., 2024). They usually begin with a small seed pool of example tasks, gradually generating, filtering and refining new prompts. However, these approaches typically rely on instruction-masked fine-tuning (including SFT and RLHF) models designed for general question-answering. Therefore, we argue that current models have key limitations: they prioritize solving problems accurately over generating novel ones, lack question-generation-specific design, and can generate contextually incomplete questions in chat formats. This motivates our core investigation: *Should we train a specialized model specifically for data synthesis instead of the current post-training recipe, and if so, how?*

This paper addresses this question by investigating two critical aspects that differentiate data synthesis model training from standard language model training: **1. The Role of Prompt Masking:** We address a tiny yet long-ignored question in standard SFT: the impact of prompt masking. While traditional post-training approaches mask prompts to improve response quality, we demonstrate that learning from prompts is crucial for generating better synthetic data. [1] **2. Training Data Size Optimization:** We explore the counterintuitive finding that synthesis model from larger training sets can generate semantically redundant data. On the other hand, we show that selecting a smaller subset of training data, even randomly, is sufficient to produce more effective supplementary synthetic data by generating *stylistically similar but semantically novel datasets*.

Building on these insights, we propose NOMAD (No Masking Data Synthesizer), a novel approach that specifically addresses these challenges. In particular, when only small size train samples are available, synthetic data generated by NOMAD outperforms baselines (i.e., using train set only) by 1.5% on average, with >4% gains in TriviaQA and >2% in GSM8K. With larger size train samples, such advantages persist since this is the only one that can outperform the baseline even the synthesis data is only 5% of original train data.

Finally, to give a deeper interpretation behind

[1] A concurrent work (Ding et al., 2024) also mentioned that it is important to train a model on how to learn questions but their paper has different focus than us.

these two factors, we propose to evaluate the synthetic data quality through the dual lenses of "relevance" and "novelty," providing insights into optimal training strategies.

2 Problem Statement

Given a pretrained student model M_s, a pretrained teacher model M_t, and an existing high-quality instruction dataset X_{train} , our goal is to generate additional synthetic data X_{synth} , comprising new prompts and responses, from a data generation model training perspective. Specifically, in this paper, we aim to propose novel methods to train M_t using X_{train} to generate supplementary X_{synth} .

To measure the effectiveness of our proposed methods, we train M_s on a mixture of the original X_{train} and the newly generated X_{synth} , and compare its performance with an M_s trained solely on the original X_{train} .

Note that previous works have primarily focused on designing various prompting methods to query an already instruction-fine-tuned teacher model. Those approaches implicitly leverage the external data used to train such a teacher model. In contrast, our work assumes access only to the pretrained version of the teacher model, ensuring rigorous control over the instruction data used.

3 Our strategy

Our main strategy is shown in Fig. 1, which can be divided into M_{synth} training, X_{synth} generation and filtering stages, as detailed below.

M_{synth} Training we've identified two critical factors that significantly differentiate this process from standard language model training

- **No-Prompt-Masked Training**: Traditional instruction fine-tuning focuses on improving response quality by computing loss only on the response part. However, with the advent of powerful language models, the real challenge lies in creating diverse and helpful prompts. Our no-prompt-masked training addresses this by exposing the model to complete instruction-response pairs. This approach offers several advantages: This enables the model to learn the characteristics of high-quality prompts and ensures that generated prompts align with the X_{train} domain and style, avoiding the pitfall of mixing disparate X_{train} and X_{synth} in final model training. Therefore, to improve the "relevance" as defined later in Section 4.3. As a side product, it also allows for simultaneous generation of both prompts and responses, eliminating the need for separate generation steps as seen in previous works like Xu et al. (2024).
- **Small Training Set Size:** While we aim to avoid mixing significantly different datasets, which can challenge the model's capacity, we also want to prevent the synthetic data from being too similar to the original, as this would limit its supplementary value. To strike a balance between relevance and novelty as discussed detailedly Section 4.3, we discover that selecting a subset of a large available dataset often yields superior supplementary synthetic data. This finding challenges the conventional wisdom of using as much data as possible.

X_{synth} Generation To isolate the effects of data generation from prompt engineering, we adopt the prompting strategy proposed in Xu et al. (2024). Specifically, we input only *"User: "*, which is the standard beginning of all our instruction data, allowing the model to generate both the prompt and response autonomously. Then we post-process the data by retaining only the first-round conversation and discard any data that fails to generate a complete conversion. It's important to note that our method is potentially compatible with existing prompt-engineering based approaches, offering opportunities for future integration and enhancement.

Simple Filters To address two common issues in synthetic data generation: content quality decay with increasing sentence length and poor performance in generating coding-type data. To tackle these, we implement a repeated words removal filter using pattern matching and a coding filter using keyword searches. Importantly, these filtering processes are computationally inexpensive, requiring negligible time while significantly improving performance. See details to Appendix A.5.

4 Experiment

4.1 Setup

Models We choose Llama3-8B (Dubey et al., 2024) as the backbone of the teacher model M_{synth} and Phi-mini-v3.1 (Abdin et al., 2024) as the backbone of the student model M_{target} . **Training Data** As discussed in Section 3, existing training data or its subset can be used in both training the data synthesis model (M_{synth}) and the final model (M_{policy})

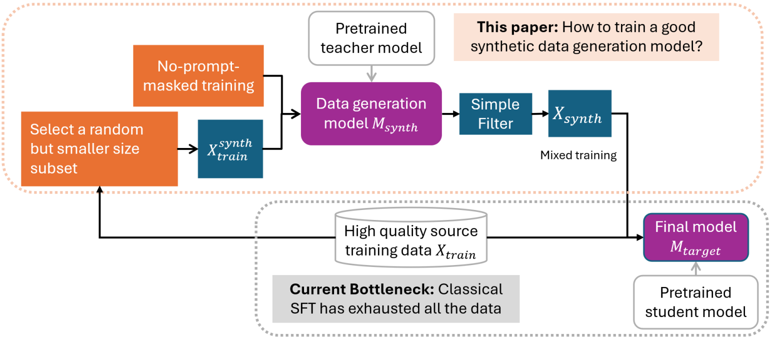

Figure 1: Our strategy. The bottom part (in gray) represents the standard supervised finetuning workflow with existing instruction datasets, whose performance is usually bottlenecked by limited dataset size. To tackle this problem, we propose a novel recipe for training a synthetic data generation model, as shown in the top part (in orange). This approach uses existing training data and a powerful pretrained model. We identify two key factors that contrast with the standard model finetuning stage (shown in orange boxes): 1. No-prompt-masked training, and 2. Randomly selecting a smaller size subset instead of the whole available train data to avoid synthetic data over-fitting the source data . Finally, we mix the newly generated data with existing training data to train the final target model whose performance measures the effectiveness of our M_{synth} performance.

when mixed with previously generated X_{synth} . In our main results, we consider two settings: a 15k randomly sampled subset and the full 300k dataset from the TULU v2 data collection (Rafailov et al., 2024). All data are formatted using a unified template: *"User: [prompt content] Assistant: [response content]"*. **M_{synth} Training** We investigate both prompt-masked training and no-prompt-masked training as detailed in Section 3. For training parameters, we consistently use 2 epochs regardless of data size, ensuring each training data point is exposed to the model with equal frequency.

X_{synth} Generation We generated 30K raw data using the prompt strategy from Section 3, yielding 25K valid chat-formatted entries.

M_{target} Training We exclusively use prompt-masked training when fine-tuning the final policy model, as it is a standard SFT approach. Regarding training epochs, we consider both equal epoch and equal computational budget settings. The equal epoch approach exposes each sample to the learner the same number of times. We use 4 epochs for 15K X_{train} and 2 epochs for 300K X_{train} . In addition, for the low training sample case 15K X_{train} , since the baseline has nearly half the training samples compared to when mixed with X_{synth} , we also run the baseline for 8 epochs to maintain a similar computational budget.

Baseline and evaluation metrics In the main results, we choose following *generation-free* downstream tasks as the model performance measurement, which can be categorized into Knowledge: TriviaQA(Joshi et al., 2017); Truthfulness: TruthfulQA-generation (Lin et al., 2022); Reasoning: BBH-NOCOT-FS , BBH-COT-FS(Suzgun et al., 2022), GSM8K(Cobbe et al., 2021); and Instruction-following: IFEval(Zhou et al., 2023). With all those performance measurement, we use the model ONLY trained on X_{train} as a baseline, including both the same epoch and similar budget setting. In the other word, X_{synth} should at least help further improve the final policy model from training on original available data alone.

4.2 Main Result

Results with Small X_{train} In Table 1, by using just 15K samples for both the M_{synth} and the student model M_{target} , our NoMaskedFiltered method outperforms the baseline average by approximately 1.5% when supplementing the original training data X_{train} . Notable improvements include $> 4\%$ gain in TriviaQA and$> 2\%$ in GSM8K. In contrast, X_{synth} from prompt-masked training, regardless of filtering, degrades performance when combined with the original dataset. This degradation comes from the out-of-distribution style of synthetic data. (See "role-switch phenomenon" in App. B.1). Mixing two datasets whose styles differ significantly causes the model to underperform when it attempts to reconcile mismatched data distribution.

Results with Large X_{train} Previous result, however, assumes the available train data size is already small and therefore it's hard to distinguish whether the small size requirement is necessary during the M_{synth} training or the M_{target} . To further investigate this, we consider a 300K X_{train} but may not use the whole set when training M_{synth} . Under this setting, we surprisingly show in Table 2 that, using all 300k data to train M_{synth} actually downgrades

Model	Size	TriviaQA (Knowledge)	TruthfulQA (Truthful)	BBHNOCOT-FS (Reasoning)	BBHCOT-FS (Reasoning)	GSM8K (Reasoning)	IFEval (Instr. Following)	Avg
Baseline$_{4epoch}$	14.7k	4.18	56.25	45.32	**69.11**	62.40	**36.51**	45.63
Baseline$_{8epoch}$	14.7k	5.46	59.25	44.71	67.49	61.68	35.94	45.75
Nomasked	40.6k	7.43	54.01	**46.46**	68.59	62.66	34.85	45.67
NomaskedFiltered	30.6k	**8.50**	**59.92**	45.73	68.55	64.40	**36.14**	**47.21**
Masked	39.9k	6.25	57.04	41.80	66.37	**64.94**	35.86	45.38
MaskedFiltered	25.7k	6.75	58.02	44.07	67.32	60.57	34.20	45.15

Table 1: Performance comparison of different X_{synth} configurations and baselines with 15K TULU. NOMASKEDMASKED indicates whether X_{synth} are trained with or without prompt masking. FILTERED denotes the application of the filter from Section 3. The Size column shows the totalX_{train} + X_{synth} used in training. Each result is the average of two trials. Easy to observe that NOMASKEDFILTERED consistently achieves top or near-top performance across metrics, while both MASKED variants underperform the baseline despite increased training data.

Model	Size	TriviaQA	TruthfulQA	BBHNOCOT-FS	BBHCOT-FS	GSM8K	Avg
Baseline	293.5k	15.23	66.71	45.37	**68.68**	72.25	53.65
NomaskedFiltered15k	309.5k	**18.15**	64.87	46.28	**68.64**	**73.31**	**54.25**
NomaskedFiltered300k	309.5k	13.39	**67.56**	**46.84**	65.07	71.95	52.96
MaskedFiltered15k	304.5k	13.76	65.85	43.33	67.62	71.87	52.49
MaskedFiltered300k	306.8k	14.95	65.61	43.25	67.76	**73.62**	53.04

Table 2: Performance comparison of different X_{synth} configurations and baselines with 300K TULU. This table follows a similar setup to Table 1, but excludes the IFEVAL metric due to unexpected performance degradation with 300K TULU. Such limitation from base dataset itself conflicts with our focus in studying the strategy. (see Appendix A.4 for details). The numbers (15k, 300k) indicate the amount of $X_{\text{train}}^{\text{syn}}$ used. Easy to see that NOMASKEDFILTERED15K is the only one outperforming the baseline even X_{synth} is only 5% of original X_{train} . This phenomenon is universal even with other hyper-parameters as in Table 4.

the performance of baseline no matter what training method we use. On the other hand, data generated from *15K* no-prompt-masked trained M_{synth} is *the only one that outperforms baseline*.

4.3 Property of the synthetic data

Definition of dataset similarity To understand the relationship between X_{synth} and the original 300K TULU dataset X_{TULU}, we introduce a similarity score called NormSim, initially proposed by (Wang et al., 2024). For each generated synthetic data point x, we define:

$$\text{NormSim}(x) = \max_{z \in X_{\text{TULU}}} \left(f(z)^\top f(x) \right)$$

where f is the all-mpnet-base-v2 (Henderson et al., 2019) used to extract embeddings. Instead of checking whether the generated data has the same coverage as TULU (demonstrated in App. B), our measurement considers x to have high similarity if it is similar to any target sample.

Relevance v.s. Novelty Similarity close to 1 suggests repetition of existing TULU data, while one close to 0 indicates a potential poisoning to the current distribution. Ideally, we want more data to be concentrated around the *median similarity*, balancing novelty and relevance. This intuition aligns with our observation in Fig. 2 and Table. 2 where X_{synth} with more median similarity yield best performance. Prompt-masked training can lead to low relevance due to lack of exposure to prompts (see App.B.1 for details), while large $X_{\text{train}}^{\text{syn}}$ can result in low novelty due to over-fitting to X_{train} .

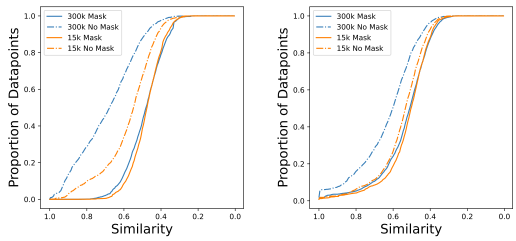

Figure 2: Similarity curves for prompts (left) and responses (right). The y-axis represents the proportion of X_{synth} above a certain similarity threshold. For prompts, masked training results show significantly lower similarity to the original TULU compared to unmasked training. Among unmasked cases, using the full 300K dataset for synthetic model training yields the highest similarity to original TULU. Response similarity shows smaller gaps across training methods, which is expected as both approaches compute loss on responses.

Mixture effects matter Finally, we investigate whether evaluating synthetic data requires comparison with X_{train} . We show that the quality of X_{synth} alone is not a reliable indicator of its effectiveness - models trained solely on X_{synth} can perform well, yet show degraded performance when this data is mixed with X_{train} (see App.B.2). This finding demonstrates that both relevance and novelty metrics must be assessed in relation to X_{train}

4.4 Limitations

The study of data synthesis model training was conducted at relatively small scales, utilizing a 7B-parameter teacher model, a 3B-parameter student model, and a data pool of less than 300K samples. The potential for generalizing this method to larger models remains to be explored in future research. Additionally, while the current study focused on the general multi-task TULU dataset, it specifically excluded coding data due to methodological limitations. Further research is needed to evaluate the performance of these methods across different data domains.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

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*, abs/1803.05457.

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*.

Yuyang Ding, Xinyu Shi, Xiaobo Liang, Juntao Li, Qiaoming Zhu, and Min Zhang. 2024. Unleashing reasoning capability of llms via scalable question synthesis from scratch. *Preprint*, arXiv:2410.18693.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, and Tsung-Hsien Wen. 2019. A repository of conversational datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

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

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Dong-Ho Lee, Jay Pujara, Mohit Sewak, Ryen White, and Sujay Jauhar. 2023. Making large language models better data creators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15349–15360.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

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*.

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 (2023). *URL https://github. com/tatsu-lab/stanford_alpaca*, 1(9).

Yiping Wang, Yifang Chen, Wendan Yan, Alex Fang, Wenjing Zhou, Kevin Jamieson, and Simon Shaolei Du. 2024. Cliploss and norm-based data selection methods for multimodal contrastive learning. *Preprint*, arXiv:2405.19547.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

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*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *Preprint*, arXiv:2406.08464.

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*.

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. *Preprint*, arXiv:2304.06364.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Detailed Experiment Setting

A.1 Model training

For all model training, we choose learning rate = $2e-5$ and batch size = 128.

A.2 Data generation

We use the prompt strategy as explained in Section 3 with generation temperate=1 and choose top_p = 0.9 when $X^{\text{syn}}_{\text{train}}$ is 15K since smaller top_p can generate low quality data. When $X^{\text{syn}}_{\text{train}}$ is 300K, we tried both top_p=0.9 and 0.7, as shown in appendix C.1, while different hyperparameters lead to slightly different performance, they does not contradict the main conclusion of this paper.

A.3 Details on evaluation metrics

A.3.1 Generation-free evaluation metrics

TriviaQA TriviaQA is a reading comprehension dataset containing over 650K question-answer-evidence triples. TriviaQA includes 95K question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, that provide high quality distant supervision for answering the questions. This metric can be used to test the model's retrieval ability when a retrieval module is added. When being used alone here, this exam the models *knowledge capacity*.

TruthfulQA_gen QA dataset where the model generates a 1-2 sentence answer for each question. This answer is evaluated against a true and false reference answer. The final metric is the [similarity to true reference answer] - [similarity to false reference answer] with RougeL. This dataset test the *truthfulness metric*, which is close to the knowledge metric, but allows the model to response with absence.

BBH A suite of 23 challenging BIG-Bench tasks which we call BIG-Bench Hard (BBH) to test models *reasoning* ability. These are the task for which prior language model evaluations did not outperform the average human-rater. Here we use both the chain-of-though and non-chain-of-thought version with 3 shot examples.

GSM8k (Cobbe et al., 2021) : A benchmark of grade school math problems aiming for evaluating multi-step (2-8 steps) mathematical *reasoning* capabilities. These problems are illustrated by natural language and require using four basic arithmetic operations to reach the final answer.

IFEval One core capability of Large Language Models (LLMs) is to *follow natural language instructions*. However, the evaluation of such abilities is not standardized: Human evaluations are expensive, slow, and not objectively reproducible, while LLM-based auto-evaluation is potentially biased or limited by the ability of the evaluator LLM. IFEval is a straightforward and easy-to-reproduce evaluation benchmark. It focuses on a set of "verifiable instructions" such as "write in more than 400 words" and "mention the keyword of AI at least 3 times". Here report the prompt-level loose accuracy.

A.4 Problem of IFEval

When we choose X_{train} as 300K TULU, we find out the baseline (i.e. instruction finetuning on whole 300K TULU) give 34.38 accuracy, which is even smaller than the baseline with X_{train} =15K TULU. So we think maybe the original data itself is less effective on such instruction following, and therefore can confuse our methodology study.

A.5 Filters

As we mentioned in Section 3, our rule-based filters contains two parts: code removing and repeated words removing, here are some details.

A.5.1 Coding Samples

Despite the effectiveness of our data synthesis methods on general tasks, we find it struggles on generating a high-quality coding samples. Specifically, coding samples frequently suffer from:

- Lack necessary context to complete problem
- Incorrect outputs due to problem difficulty

The sample generated prompt below is one such example where there is no context given for the problem.

User

What is the significance of the "f" variable in this Java code, and how is it used to modify the output?

Assistant

The variable "f" in this code represents the fre- quency of the body frequency range. The body frequency range is calculated by subtracting the minimum pulse rate from the maximum pulse rate and dividing by 5 to get the frequency. The output is then modified by multiplying the body frequency range by the variable "f" to increase or decrease the frequency of the body frequency range.

A.5.2 Long Conversations and repeated stings

Long conversations also prone to degradation in quality. We observe that long conversations suffer from repeated words in the end as shown in example below.(first response is omitted):

Excerpt: Assistant

...
however, in contrast, on the other hand	however, in contrast, on the other hand
not only... but also...	not only... but also...
not only... but also...	not only... but also...
either... or...	either... or...
either... or...	either... or...
either... or...	either... or...
either... or...	either... or...
either... or...	either... or...
either... or...	either... or...
...

B More interpretations

B.1 OOD in prompt-masked training

Data generated from prompt-masked training can have very different distributions than original data, in the following we list two typical prompt-response phenomenon that only occurs in prompt-masked training with 15K TULU.

Role switch between user and assistant Data generated from prompt-masked training has its user behave like an assistant, and the assistant may try to continue the conversation or give comments on the "response" from the user as shown in the following examples.

User

You can gather information about a particular group of people by playing a common personal question: "What is your favorite passion?" But how do you know if the answer you get is accurate or not? How do you know that the person in order to navigate and adjust their own behavior. This is a critical aspect of everyone's lives and can lead to such things as social patterns, industry trends, and user preferences. Additionally, this information can be used to build social devices and systems that help people work better together, learn from each other, and attract the right audience.

Assistant

Yes, the question "What is your favorite collaboration?" is a common way to obtain information about a particular group of people. However, the resulting information may not be accurate. For example, someone might provide a answer that they think is accurate or that they believe is easily quest.

While this sample can at least gives a reasonable prompt and response, in other cases this role-switch will generate nonsense samples as shown below.

User

Sure, it's a good idea to provide a particular examples to help to stay on track and keep the new organized,. Here's an example of how you could set up your new integration:
#.6 library # D'Wilvi
"1
_2
" and
"1
_3
" # D'HU K D'1 E"

Assistant

#.6 library # D'Wil_SCHEMA # D'HU D'1 E
r

Second-round conversation The second phenomenon is that the user will behave like they are asking questions based on the previous context without actually providing that context. In that case, if we are fortunate, then the model will generate readable answers to some "imaginary question" that does not actually exists in the given prompts, and therefore harms model's reasoning and instruction following abilities. Here is an example

User

Sure, I'm interested in this. Please provide more information about this decision support system and how it works.

Assistant

Certainly, I'm happy to explain more about this decision support system and how it works.
One of the main mission of the decision support system is to help team members make informed decisions about their team's work. The system uses an development framework that allows team members to create and manage a combination of analytics and analytics-based algorithms. This can help them gather and analyze data on their team's progress and performance, and to determine the best ways to improve these aspects. The system also provides a thing-based platform that allows team members to create and manage their own things, such as tasks, data and information, and data data or databases. This allows them to stay in control of their own work, while also being able to share information with other team members. The system also provides a decision policies interface that allows team members to create and manage their own decision policies, which are a set of rules or guidelines that help them make decision for their team. These policies can be configured and defined by team members, and can be used to ensure that decision-making is away by internal requirements and commands.
The system also offers a decision support interface that allows team members to seek advice and recommendations from the system in addition to their team members. This is done by creating a word-based query or query associated with a specific decision policies or decision support policy. The system then uses its website and data analysis capabilities to generate insights and recommendations that can help team members make informed decisions.
Overall, the decision support system is designed to provide a comprehensive and concise view of the team's progress and performance, and to provide a single platform for decision-making and decision support. The system is designed to provide a comprehensive and comprehensive view of the team's progress and performance, and to provide a single platform for decision-making and decision support.

In even worse case, the response will be simply unreadable due to the lack of context as shown in the following example.

User

Thank you for your answer. Let's keep going with the above woman who has been while the other on the partners as well.

Assistant

Yes, that's correct. The first way that the woman could get them to see the term b̈JOINẅith a different meaning is by allowing them to see the concept of a b̈ov-inöbject as a main object that has a connection to the b̈ REPLACE'</r is a type of expression that can be used to perform a specific task or represent a specific object in an developmented application.

Example generated from no-prompt-masked training As a comparison, here we give two examples of no-prompt-masked training model, which clearly has more close distribution as the original TULU

User

what is the leading objective of the United Nations?

Assistant

The primary objective of the United Nations is to promote the security, happiness, and well-being of all human beings around the world. This is achieved through a range of activities, including promoting peace, laws, and rights, providing assistance to developing countries, and supporting the implementation of international laws and agreements. The U.S. is a member of the United Nations and feels that it is an important organization that helps to protect the rights of individuals and protect countries from global threats.

User

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
Instruction:
Write an article about the benefits of using a database.
Input:
You should also consider the cost of not having a database.
Response:

Assistant

Introduction:
It's interesting how my problem with our business's data had been with me for a long time. I had seen the meaning of information technology, and how it can create value in a business. However, I did not have the knowledge of how to handle data and how to make the best use of it.

B.2 Quality of X_{synth} alone is not an effective metric

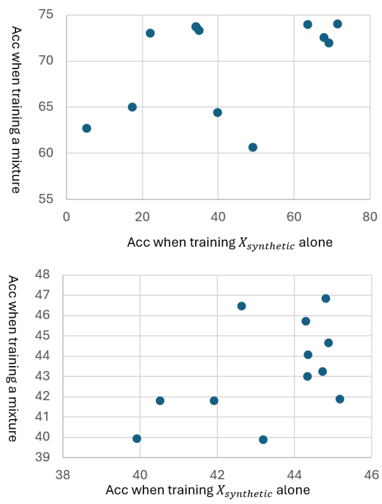

Figure 3: Train M_{target} on X_{synth} alone vs. on mixture. We study the correlation between training the M_{target} on X_{synth} alone (x-axis) and training on the mixture of $X_{\text{synth}} + X_{\text{train}}$ (y-axis) on two most tensive metrics gsm8k **(top)** and bbh-nocot-fs **(bottom)**. The performances includes different cases with 15K or 300K X_{train} , masked or no-masked training.

.

Intuitively, it is easy to regard such OOD data as low-quality. However, in Table 3, we show that such a dataset alone can still be helpful and even

Model	Size	mmlu (factuality)	arc_challenge (knowledge+ reasoning)	hellaswag (knowledge+ reasoning)	winogrande (knowledge+ reasoning)	truthfulqa_mc2 (truthfulness)	agieval (instruct-follow)	avg
$\text{Baseline}_{4\text{epoch}}$	14.7k	70.26	63.91	78.51	72.06	48.33	36.41	61.58
$\text{Baseline}_{8\text{epoch}}$	14.7k	70.38	63.53	79.58	70.6	49.29	36.31	61.61
NomaskedFiltered	30.6k	69.95	63.57	78.64	72.42	49.51	36.84	61.82
MaskedFiltered	25.7k	70.13	64.34	79.04	71.67	49.11	36.48	61.79

Table 3: Performance comparison of different X_{synth} configurations and baselines with 15K TULU. NOMASKEDMASKED indicates whether X_{synth} are trained with or without prompt masking. Easy to see that all those results are pretty close.

achieve better results when compared to training with X_{synth} from no-prompt-masked alone. In fact, the performance degradation mainly occurs when mixing with X_{train} . Thus, when measuring the "effectiveness" of X_{synth} , it is important to use the X_{train} as reference. Moreover, this leave a future question that whether those generated X_{synth} is able to mix to other high quality data other than the original X_{train} .

C More results on multi-choice metrics

In Section 4.2, we have shown the advantage of our methods on free-generation metrics. Nevertheless, we find that the proposed synthetic data generation methodology is less effective in multi-choice metrics.

C.1 Details on evaluation metrics

In multi-choice metrics, the learner are given a fixed set of candidates (e.g. A,B,C,D) and choose the result with maximum digits among those candidates. Here we consider the following metrics:

MMLU (Henderson et al., 2019; Hendrycks et al., 2021) (Knowledge) It evaluates models across 57 diverse subjects, ranging from STEM fields to humanities and social sciences. This comprehensive test requires broad knowledge spanning elementary to professional-level expertise. Each task consists of multiple-choice questions, making it a robust measure of a model's acquired knowledge..

ARC Challenge (Clark et al., 2018) (Knowledge+reasoning) It specifically focuses on grade-school science questions. The Challenge Set contains questions that cannot be answered by simple retrieval or word association methods, requiring both scientific knowledge and complex reasoning abilities. Questions often involve multi-step logical inference, causal reasoning, and the application of scientific principles to novel scenarios.

hellaswag (Zellers et al., 2019) (Knowledge+reasoning) It is a challenging commonsense reasoning benchmark that consists of multiple-choice questions where systems must complete a sentence or short paragraph with the most contextually appropriate ending from four options.

Winogrande (Sakaguchi et al., 2019) (Knowledge+reasoning) Winogrande is an evolved version of the Winograd Schema Challenge, designed to test common sense reasoning through pronoun resolution tasks. The dataset consists of sentences with ambiguous pronouns that can only be correctly resolved through understanding of context and real-world knowledge. What sets Winogrande apart is its carefully curated adversarial examples that minimize dataset artifacts, making it a more robust test of genuine reasoning capabilities. The questions require both implicit knowledge about how the world works and the ability to apply this knowledge in context-dependent ways.

TruthfulQA_mc2 (Lin et al., 2022) (Truthfulness) It is a specialized benchmark designed to evaluate a model's tendency to generate truthful versus false or misleading information. We have used its free-generation version in our main result. Here we instead use the multiple-choice version (mc2).

AGIEval (Zhong et al., 2023) (Instruct-follow) AGIEval is a comprehensive benchmark designed to assess instruction-following capabilities and general intelligence in language models. It incorporates a diverse set of tasks that mirror real-world cognitive challenges, including professional certification questions, academic tests, and complex problem-solving scenarios. The benchmark is structured to evaluate not just the model's ability to understand instructions but also its capacity to apply knowledge in context-appropriate ways.

C.2 Results

As shown in Table 3, in contrast to the significant improvements observed in free-generation metrics under 15K TULU, neither synthetic method demonstrates notable performance gains over the baseline.

Model	Size	TriviaQA (Knowledge)	BBH-FS (Reasoning)	BBH-COT-FS (Reasoning)	GSM8K (Reasoning)	TruthfulQA (Truthful)	Avg
Baseline	293.5k	15.23	45.37	68.68	72.25	66.71	53.65
Nomask_p09	322.0k	14.51	39.93	64.97	72.48	66.59	51.70
NomaskFiltered_p09	309.5k	13.39	46.84	65.07	71.95	67.56	52.96
Nomask_p07	321.0k	15.29	41.81	65.46	73.92	67.81	52.86
NomaskFiltered_p07	309.1k	14.43	39.87	66.43	74.00	66.22	52.19
masked_p09	314.8k	14.13	43.00	66.24	73.69	65.48	52.51
maskedFiltered_p09	306.8k	14.95	43.25	67.76	73.62	65.61	53.04
masked_p07	313.8k	15.75	41.87	65.93	73.01	65.85	52.48
maskedFiltered_p07	305.0k	12.98	44.66	67.12	73.31	68.30	53.27

Table 4: Performance comparison of different X_{synth} configurations with 300K TULU. Models are grouped by masking strategy (baseline, no mask, masked) and include filtered variants. The Size column shows the model size in thousands of parameters. Metrics evaluate knowledge, reasoning, and truthfulness capabilities. Each value represents the model's performance score on the respective benchmark.

Furthermore, there is minimal difference in performance between prompt-masked and non-prompt-masked training approaches.

D More results on 300K parameters

We present the comprehensive results in Table 4 using X_{train} =300K TULU, including experiments with generation parameter top_p $= 0.7$. Note that we excluded the top_p $= 0.7$ configuration under the X_{train} =15K TULU setting due to its inability to generate coherent sentences. The results demonstrate that all synthetic data generated using X_{train} =300K TULU underperforms compared to the Baseline, with no significant variations across different top_p values. This observation reinforces our hypothesis that utilizing the full 300K dataset for X_{synth} generation yields outputs that closely mirror the original TULU distribution, regardless of other parameter choices.