Cross-Instruction: Improving Pretrained Language Models using Domain Expert Models

Fabian Karl
fabian.karl@uni-ulm.de

1 INTRODUCTION

General-purpose Language Models (LMs) have made remarkable advancements in generating coherent and contextually relevant text. However, these models often lack domain-specific knowledge and may not perform optimally in specialized contexts [10, 12]. Gathering domain-specific instructions from human experts can be a costly and time-consuming process, making it challenging to develop LMs with expertise in multiple domains. Recent research by Wang et al. [21] explore an alternative approach to instruction tuning by leveraging LMs to generate the instructions themselves. This method, known as self-instruction, shows promise in aligning LMs by utilizing their own generated instructions. Motivated by the potential of self-instruction as a cost-effective and scalable technique for aligning LMs with domain-specific knowledge, we introduce *Cross-Instruction*, a novel procedure that leverages the expertise of domain-specific models to transfer their knowledge to a general-purpose LM. By incorporating domain-specific knowledge into the model, we can enhance its adaptability and performance in various specialized contexts. Expertise in specialized domains is often limited to a small group of individuals or organizations, making it challenging to access and leverage such knowledge effectively. Traditional instruction tuning methods rely on human experts to manually create instructions, which can be prohibitively costly and time-consuming in domains with limited available expertise. By employing self-instruction techniques, we can leverage existing domain-specific models as instructors, enabling the transfer of rare and expensive domain knowledge to a general-purpose LM. This approach offers a more efficient and cost-effective way to incorporate domain expertise into the model, bypassing the need for extensive manual instruction creation.

To address this challenge, we propose a methodology that leverages multiple instructor models, each specialized in different domains, to provide a broader and more diverse range of instructions for the self-instruction process. By aggregating knowledge from multiple sources, we aim to enhance the domain expertise and overall performance of the resulting LM.

Our primary contributions are as follows:

(1) We create and publish a new synthetic dataset specifically designed for the self-instruction process, focusing on rare and domain-specific knowledge. The dataset is generated in three different sizes, allowing us to investigate the impact of dataset scale on the resulting LM.
(2) We propose a methodology that leverages multiple instructor models, each specialized in a different domain, to provide domain-specific instructions for the self-instruction process.
(3) We evaluate the performance of the resulting LM and compare its performance with commonly used LMs on various benchmarks

2 RELATED WORK

In this section, we discuss the related work on generative LMs, instruction tuning, and knowledge distillation. Finally, we briefly touch upon the criticism of the self-instruction process and its relevance to our work.

2.1 Generative Language Models

Generative LMs have been at the forefront of natural language processing (NLP) research. The breakthrough in this field came with the introduction of the Transformer architecture [20]. This architecture allowes for efficient processing of sequential data, leading to significant advancements in generative language modeling. Prominent among the generative LMs are the GPT series, including GPT-1 to GPT-4. These models, known for their simplicity yet powerful performance at a large scale, have been instrumental in various NLP tasks [2, 13, 15, 16]. Another notable model is LLaMA which comprises a collection of foundation LMs ranging from 7B to 65B parameters [19].

In addition to general-purpose LMs, there has been significant progress in developing domain-specific LMs. For example, BioGPT is a LM specifically designed for the biomedical domain, enabling researchers to generate biomedical texts with improved domain relevance [12]. Similarly, StarCoder focuses on code generation for multiple programming languages, aiding developers in automating code writing tasks [10].

2.2 Instruction Tuning

Instruction tuning techniques have emerged as a means to enhance the performance and adaptability of generative LMs. These techniques involve fine-tuning models based on instructions or demonstrations.InstructGPT utilizes reinforcement learning from human feedback (RLHF) to fine-tune GPT-3. By leveraging RLHF, InstructGPT produces 1.3B parameter models whose outputs are preferred over those generated by the larger 175B GPT-3 model [14]. Another approach to instruction tuning is self-instruction, as demonstrated by Wang et al. [21]. This approach enhances the model by generating instructions using the vanilla model and subsequently training it to adhere to those instructions ALPACA [18] is a fine-tuned model derived from a 7B LLaMA model [19], using 52K instruction-following data generated by the Self-Instruct techniques. The objective of ALPACA is to develop a robust and cost-effective model, enhancing the capabilities of LLaMA within a budget constraint of less than $600. Other notable works include OpenAssistant [9], an supervised-fine-tuning (SFT) model based on Pythia 12B [1], and Vicuna, an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations [5].

2.3 Knowledge Distillation

Knowledge distillation is a technique where a model is trained to mimic the behavior and outputs of another model, often referred to as the teacher model [6, 8]. In the context of instruction tuning, knowledge distillation can be seen as transferring the knowledge or expertise of a pretrained model to improve the performance of another model. By leveraging the knowledge acquired by the teacher model, the student model can learn from its outputs and generate more accurate and contextually appropriate responses.

2.4 Criticism of the Approach

While instruction tuning has shown promising results in improving the capabilities of LMs, there have been criticisms and limitations associated with this approach.

LIMA (Less Is More for Alignment) [23] highlights the importance of quality over quantity in instruction tuning. It argues that the effectiveness of instruction tuning methods can be significantly improved by focusing on generating high-quality instructions rather than increasing the quantity of training data.

"The False Promise of Imitating Proprietary LLMs" [7] discusses the limitations of imitating proprietary LMs using weaker open-source models. It suggests that there exists a substantial capabilities gap between open and closed LMs, and simply imitating the style of a proprietary model may not guarantee similar factuality or performance. The paper emphasizes the need for developing better base LMs as the most effective strategy for improving open-source models, rather than relying on imitation techniques.

3 MODELS

In this section, we describe the models used in our approach, including the LM for training and the domain-specific models for instruction generation.

Base Model: To train our LM, we utilize the LLaMA-65B model [19]. LLaMA-65B is a large-scale LM with 65 billion parameters that is openly available for research purposes. We choose this model as our base LM due to its extensive capacity and proven performance in various natural language processing tasks.

General Instruction Model: For generating general instructions during the self-instruction process, we employ the GPT-4 API[1] as our general instruction model. GPT-4 is a powerful LM that can generate coherent and contextually relevant text.

Domain Expert Models: In order to incorporate domain-specific knowledge into our LM, we employ two domain expert models: Starcoder and BioGPT.

Starcoder [10]: The Starcoder model is a domain-specific LM designed specifically for code generation across multiple programming languages. It leverages its expertise in code-related tasks.

BioGPT [12]: BioGPT is a domain-specific LM that focuses on the biomedical domain. It has been trained on a large corpus of biomedical literature and exhibits strong performance in understanding and generating text specific to biomedical concepts and terminology.

4 EXPERIMENTAL APPARATUS

4.1 Datasets

For evaluation purposes, we utilize several benchmark datasets that cover both general instructions and domain-specific tasks. These datasets include:

- **Super-NaturalInstructions [22]:** This benchmark dataset contains 1,616 diverse NLP tasks accompanied by expert-written instructions. It serves as a general instruction benchmark.
- **HumanEval [4]:** This dataset consists of 164 samples of Python code, each accompanied by hand-written instructions.
- **MultiPL-E [3]:** Building upon the HumanEval dataset, MultiPL-E extends the evaluation to 18 programming languages.
- **MedNLI [17]:** This dataset contains 14,049 unique sentence pairs, annotated by medical professionals. It serves as a domain-specific benchmark in the biomedical domain.

By utilizing these benchmark datasets, we can evaluate the LM's performance on both general and domain-specific tasks.

4.2 Assumptions

(1) **Strong Overall Capability of Large Language Models (LLMs):** We assume that large language models, such as GPT-4 and LLaMA, possess a strong overall capability that enables them to generate high-quality instructions, identify the appropriate domain model in a few-shot setup, and effectively learn new content that was not extensively present during the pretraining phase. This assumption forms the basis for leveraging the self-instruction process and integrating domain expertise into the LM.
(2) **Domain Models' Domain-specific Superiority:** We assume that domain models are stronger within their respective domains compared to general models. This is because domain models are specifically trained and fine-tuned on domain-specific data, enabling them to possess a deeper understanding and expertise in their designated domains.

4.3 Procedure

To develop and evaluate our approach, we follow the following procedure:

(1) **Dataset Creation:** We start by creating datasets that encompass a wide range of domain-specific tasks. We incorporate three different domains: general, biomedical, and code generation. For each domain, we collect a seed set of tasks and their corresponding domain labels.
(2) **Training LLaMA:** LLaMA-65B is trained on the generated datasets, and we iterate the process of dataset creation and LLaMA training across three different dataset sizes. This iterative approach allows us to investigate the influence of dataset scale on the model's performance.
(3) **Evaluation:** We evaluate every LM performance on both general and domain-specific benchmark tasks: Super-NaturalInstructions [22], HumanEval [4], MultiPL-E [3] and MedNLI [17].

To generate the necessary data for training, we set up the following data creation pipeline:

(1) We start with a seed of tasks and their corresponding domains (e.g., general, bio, code) to form the task pool.
(2) Using the task pool, we employ a general instruction model, which in our case is the GPT-4 API, to generate a set of instructions.
(3) Based on the task pool, the general instruction model identifies the appropriate domain model to generate the response or instance in a few-shot setup with 10 examples.
(4) The selected domain model then generates the response or instance based on the provided instruction and task.

[1] https://platform.openai.com/docs/models/gpt-4

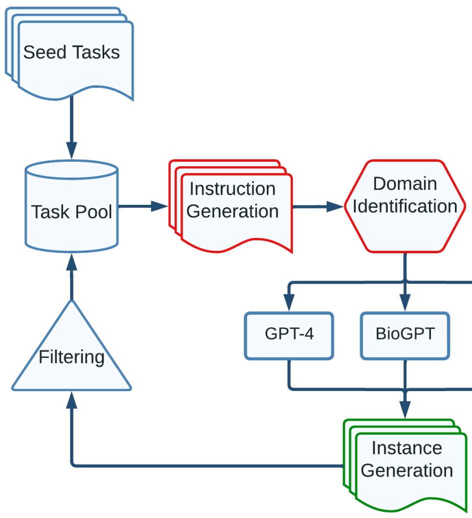

Figure 1: An illustration of the dataset creation procedure. The steps marked in red are performed by the general instruction model, while the steps marked in green are carried out by domain expert models

(5) After generating the instructions, we apply the quality filtering procedure proposed by Wang et al. [21] to ensure the inclusion of high-quality instruction-answer pairs.

The steps (2) to (5) are repeated iteratively until the desired number of instruction-answer pairs is generated, which in our case is 5,000, 52,000, and 100,000 instruction-answer pairs. An illustration of this can be seen in Figure 1.

4.4 Metrics

To evaluate the performance of our LMs, we employ the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric, specifically ROUGE-L [11]. ROUGE is a widely used metric in the field of natural language processing for assessing the quality of generated text. ROUGE-L measures the longest common subsequence of words between the generated text and the reference text. It focuses on capturing the recall aspect of the generated text by evaluating the overlapping content with the reference text.

REFERENCES

[1] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. *CoRR* abs/2304.01373 (2023). https://doi.org/10.48550/arXiv.2304.01373 arXiv:2304.01373

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[3] Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. 2022. MultiPL-E: A Scalable and Extensible Approach to Benchmarking Neural Code Generation. arXiv:2208.08227 [cs.LG]

[4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison 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, Joshua 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. *CoRR* abs/2107.03374 (2021). arXiv:2107.03374 https://arxiv.org/abs/2107.03374

[5] 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. https://lmsys.org/blog/2023-03-30-vicuna/

[6] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* 129, 6 (2021), 1789–1819. https://doi.org/10.1007/s11263-021-01453-z

[7] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The False Promise of Imitating Proprietary LLMs. *CoRR* abs/2305.15717 (2023). https://doi.org/10.48550/arXiv.2305.15717 arXiv:2305.15717

[8] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015). arXiv:1503.02531 http://arxiv.org/abs/1503.02531

[9] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant Conversations - Democratizing Large Language Model Alignment. *CoRR* abs/2304.07327 (2023). https://doi.org/10.48550/arXiv.2304.07327 arXiv:2304.07327

[10] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. StarCoder: may the source be with you! *CoRR* abs/2305.06161 (2023). https://doi.org/10.48550/arXiv.2305.06161 arXiv:2305.06161

[11] Chin-Yew Lin and Franz Josef Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, Donia Scott, Walter Daelemans, and Marilyn A. Walker (Eds.). ACL, 605–612. https://doi.org/10.3115/1218955.1219032

[12] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinform.* 23, 6 (2022). https://doi.org/10.1093/bib/bbac409

[13] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/arXiv.2303.08774 arXiv:2303.08774

[14] Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

[15] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

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

[17] Alexey Romanov and Chaitanya Shivade. 2018. Lessons from Natural Language Inference in the Clinical Domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and

Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 1586–1596. https://doi.org/10.18653/v1/d18-1187

[18] 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.

[19] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/arXiv.2302.13971 arXiv:2302.13971

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[21] 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 13484–13508. https://aclanthology.org/2023.acl-long.754

[22] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 5085–5109. https://aclanthology.org/2022.emnlp-main.340

[23] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. *CoRR* abs/2305.11206 (2023). https://doi.org/10.48550/arXiv.2305.11206 arXiv:2305.11206