EXTENDED-ABSTRACT

Synopsis: Using (Not-so) Large Language Models to Generate Simulation Models in a Formal DSL: A Study on Reaction Networks

JUSTIN NOAH KREIKEMEYER, University of Rostock, Rostock, Mecklenburg-Vorpommern, Germany

MIŁOSZ JANKOWSKI, University of Rostock, Rostock, Mecklenburg-Vorpommern, Germany

PIA WILSDORF, University of Rostock, Rostock, Mecklenburg-Vorpommern, Germany

ADELINDE M UHRMACHER, University of Rostock, Rostock, Mecklenburg-Vorpommern, Germany

Published: 23 June 2025

Citation in BibTeX format

SIGSIM-PADS '25: 39th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation
June 23 - 26, 2025
Santa Fe, USA

Conference Sponsors:
SIGSIM

https://doi.org/10.1145/3726301.3728415
ISBN: 9798400715914

Synopsis: Using (Not-so) Large Language Models to Generate Simulation Models in a Formal DSL: A Study on Reaction Networks

Justin Noah Kreikemeyer
University of Rostock
Institute for Visual and Analytic Computing
Rostock, Germany
justin.kreikemeyer@uni-rostock.de

Miłosz Jankowski
University of Rostock
Institute for Visual and Analytic Computing
Rostock, Germany
milosz.jankowski@uni-rostock.de

Pia Wilsdorf
University of Rostock
Institute for Visual and Analytic Computing
Rostock, Germany
pia.wilsdorf@uni-rostock.de

Adelinde M. Uhrmacher
University of Rostock
Institute for Visual and Analytic Computing
Rostock, Germany
adelinde.uhrmacher@uni-rostock.de

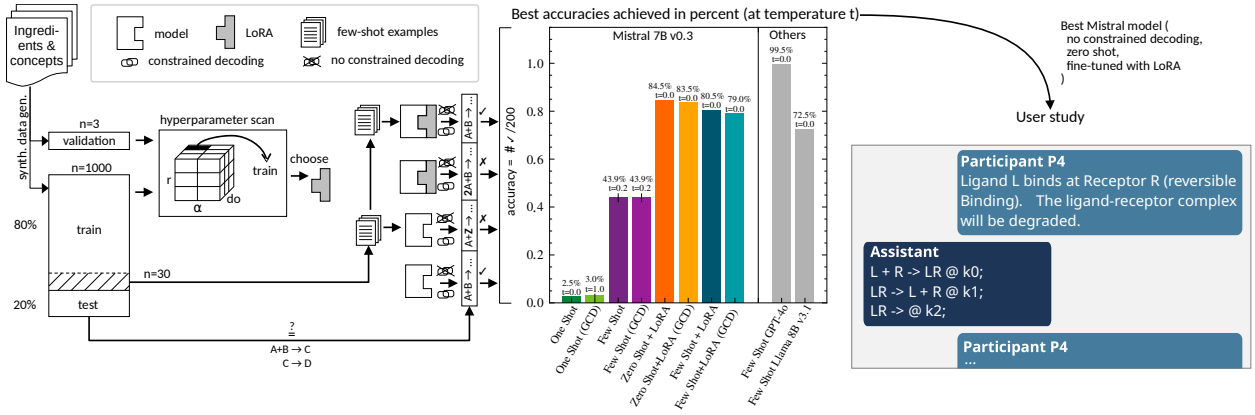

Figure 1: Overview of the evaluation procedure and main results. Left: testing few-shot prompting, low-rank adaptation (LoRA), and grammar-constrained decoding (GCD). Center: accuracies of the method combinations in translating from natural language to reaction networks. Right: excerpt from the small-scale user study with the selected best obtained combination.

CCS Concepts

• **Computing methodologies** → *Modeling methodologies*; **Machine translation**.

Keywords

simulation model generation, natural language processing, language model, constrained decoding, knowledge extraction

ACM Reference Format:
Justin Noah Kreikemeyer, Miłosz Jankowski, Pia Wilsdorf, and Adelinde M. Uhrmacher. 2025. Synopsis: Using (Not-so) Large Language Models to Generate Simulation Models in a Formal DSL: A Study on Reaction Networks. In *39th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS '25), June 23–26, 2025, Santa Fe, NM, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3726301.3728415

SIGSIM-PADS '25, Santa Fe, NM, USA

ACM ISBN 979-8-4007-1591-4/25/06
https://doi.org/10.1145/3726301.3728415

1 Introduction

Modeling is the process of finding a useful abstraction to describe an observed system. Whereas typically formulated in executable domain-specific modeling languages (DSL), natural language is arguably the most accessible and intuitive means for humans to express their knowledge, as evidenced by decades of research in natural language processing (NLP) [2]. Of course, translating from natural to formal language is prone to be incomplete and ambiguous, but the approximate solution to this problem is highly relevant and we are hardly the first to tackle the question: To what extent can modeling with DSLs be supported by automatically translating from natural language to a formal model? [1, 3]. To this end, in [11] we evaluate a recent development in NLP, Large Language Models (LLM). To prepare an LLM for this task, several methods can be employed: Prompting strategies, parameter-efficient finetuning [8], and grammar-constrained decoding [7]. The few existing approaches to translate from natural language to simulation models using LLMs rely on huge, commercial models and prompting alone.

We argue that a combination of fine-tuning, few-shot prompting, and other techniques allow the use of much smaller, open-weight LLMs for the translation task, requiring only a fraction of the resources and providing full control over reproducibility [11].

2 Background and Methods

Large Language Models. Most of today's LLMs are based on the *transformer* neural network architecture [6]. They are highly parameterized networks trained on vast amounts of text. Fine-tuning them on conversations between a "user" and an "assistant" further enables interaction with a human end-user. We evaluate the 7.3 billion parameter, Apache 2.0 licensed "Mistral Instruct v0.3" [9] ($\text{Mistral}^{\text{7B}}_{\text{v0.3}}$) and also compare to the 8 billion parameter, proprietary licensed "Llama Instruct v3.1" [12] ($\text{Llama}^{\text{8B}}_{\text{v3.1}}$) and the very large, commercial "GPT-4o" by OpenAI [4] (GPT_{4o}).

Prompting Techniques. LLMs are foundation models: they can be tailored to new tasks without re-training. This is done by providing specific instructions and examples. Depending on the number of examples provided, *zero-* (no examples, just instructions) or *few-shot* (multiple examples) prompting are distinguished. Instructions are often also supplied via a special *system prompt*. However, the generalization capability of small LLMs is limited, such that prompting is often insufficient to yield good performance on a task, necessitating fine-tuning.

Synthetic Data Generation. As training data in our use-case, but also in general for DSLs [10], is scarce, we built a *synthetic data generator* (Figure 1, top left). It composes high-level concepts, such as "production", "death", or "chain reactions", and ingredients, such as species names and attributes, into models. Model generation is done at random, as for translation tasks the scientific relevance of the output is not important. Each concept is linked to a variety of natural language template formulations, from which one is chosen and instantiated to match. One entry in the dataset comprises a model (the "output") built from multiple concept instantiations, and their varying verbal description (the "input"). We generate a training (testing) corpus of 800 (200) examples (on disjunct template formulations/ingredients).

Parameter-Efficient Fine-tuning. Tuning all parameters of an LLM is time- and memory-intensive. Instead, efficient ways that do not adjust all parameters, such as low-rank adaptation (LoRA) [8] are often used in practice. We here employ LoRA to fine-tune the $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$ model to our data.

3 Quantitative Results

In our evaluation, we tested the combination of the basic techniques above when applied to the $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$ model (Figure 1, left). Our results (Figure 1, center) show that LoRA fine-tuning can significantly boost the performance of $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$ compared to just few-shot prompting. Still, it cannot reach the accuracy of the much larger GPT_{4o} (84.5% vs. 99.5%). However, considering the (estimated) three to five orders-of-magnitude difference in parameters, it comes remarkably close. Constrained decoding did not have a large impact. We postulate this technique might become more important for more expressive DSLs. The only 1B larger $\text{Llama}^{\text{8B}}_{\text{v3.1}}$ model's few-shot accuracy is much higher than $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$'s. Thus, fine-tuning $\text{Llama}^{\text{8B}}_{\text{v3.1}}$ presents an interesting avenue for future work.

4 Small-scale User Study

In a small study involving five scientists, we tested how well our prototype performs under practical circumstances (Figure 1, right). We found that the fine-tuned $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$ provided reasonable answers even in some difficult situations. However, the current prototype often fails at interaction, suggesting that our training did not retain all capabilities of the original model. The augmentation of the synthetic data generation in this regard, and also including additional domain concepts desired in the user study [11], is thus an interesting future direction.

5 Conclusion

We think that future developments of small and open-weights LLMs, like DeepSeek's recent distill models [5], and higher-quality data can close the gap to much larger LLMs for specific tasks, such as translating natural language to formal DSLs. With $\text{Mistral}^{\text{7B}}_{\text{v0.3}}$ both, training and inference, could be achieved with around 24GB of VRAM on a consumer NVIDIA RTX 3090Ti, enabling local use and adaptation of the LLM. In a broader scope, our findings suggest that LLMs will enable entirely new opportunities for modeling by interacting with (locally executable) computer assistants.

Acknowledgments

Partially funded by the Deutsche Forschungsgemeinschaft, Grant No.: 320435134 (https://gepris.dfg.de/gepris/projekt/320435134).

References

[1] John Chen, Xi Lu, Yuzhou Du, Michael Rejtig, Ruth Bagley, Mike Horn, and Uri Wilensky. 2024. Learning Agent-based Modeling with LLM Companions: Experiences of Novices and Experts Using ChatGPT & NetLogo Chat. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 141, 18 pages. doi:10.1145/3613904.3642377

[2] K. R. Chowdhary. 2020. Natural language processing. *Fundamentals of Artificial Intelligence* (2020), 603–649. doi:10.1007/978-81-322-3972-7_19

[3] W.R. Cyre, J.R. Armstrong, and A.J. Honcharik. 1995. Generating Simulation Models from Natural Language Specifications. *SIMULATION* 65 (10 1995), 239–251. Issue 4. doi:10.1177/003754979506500402

[4] OpenAI API Documentation. last accessed 2024-11-12 16:30 UTC+1. GPT-4o. https://platform.openai.com/docs/models#gpt-4o

[5] DeepSeek-AI et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[6] Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. 2024. A Bibliometric Review of Large Language Models Research from 2017 to 2023. *ACM Trans. Intell. Syst. Technol.* 15, 5, Article 91 (Oct. 2024), 25 pages. doi:10.1145/3664930

[7] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore. doi:10.18653/v1/2023.emnlp-main.674

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[9] Albert Q. Jiang and et al. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[10] Sathvik Joel, Jie JW Wu, and Fatemeh H. Fard. 2024. A Survey on LLM-based Code Generation for Low-Resource and Domain-Specific Programming Languages. arXiv:2410.03981 [cs.SE] https://arxiv.org/abs/2410.03981

[11] Justin N. Kreikemeyer, Miłosz Jankowski, Pia Wilsdorf, and Adelinde M. Uhrmacher. 2025. Using (Not-so) Large Language Models to Generate Simulation Models in a Formal DSL: A Study on Reaction Networks. *ACM Transactions on Modeling and Computer Simulation* 35, 4 (2025).

[12] AI @ Meta Llama Team. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783