**Chalamalasetti Kranti**

University of Potsdam
Karl-Liebknecht-Straße 24-25
14476 Potsdam
Germany

kranti.chalamalasetti@uni-potsdam.de

## 1 Research interests

Spoken dialogue systems (SDSs) aims to enable natural, interactive and collaborative conversations. My research interest lies in leveraging these **situated collaborative conversations** to teach new concepts (*skills*) to collaborative robots (cobots). These **cobots**, when operating in manufacturing environments such as assembly lines, are envisioned to converse with humans, reach common ground, and learn new skills in "one shot" without the need for multiple demonstrations. Unlike SDSs in consumer domains, these cobot-based systems must handle conversations in noisy, time-sensitive industrial settings.

Motivated by these challenges, my research focuses on building **collaborative dialogue systems** capable of integrating **conversational programming** (Brummelen et al., 2020) to translate situated dialogue into modular programs (Paetzel-Prüsmann et al., 2022), knowing **when to ask for clarifications** (Shi et al., 2022; Rahmani et al., 2023; Madge and Poesio, 2024), and adapting the **program based on corrections** (Fakhoury et al., 2024).

### 1.1 Conversational Programming

Conversational programming for cobots introduces challenges related to natural language understanding and translation. In this setting, human instructors are often novices, unfamiliar with programming concepts or the system's internal workings. As a result, their instructions may be ambiguous, unstructured, and span multiple turns involving clarifications and corrections. This contrasts with traditional robotic commands, where the goal is immediate execution (*e.g., place the fruits into a bowl*); in cobot programming, the objective is to generate modular programs that can be reused in future contexts.

In this direction, we first explored the program synthesis capabilities of large language models (LLMs) using a retrieval-augmented approach (Kranti et al., 2024a) in a Minecraft building task (Narayan-Chen et al., 2019). Although our approach outperformed the baseline (Jayannavar et al., 2020), it also highlighted several challenges inherent in the dataset. To narrow the focus toward controlled action generation in industrial scenarios, we proposed a 2.5D [1] structure-building task (Kranti et al., 2024b) involving assembly components. The objective is to reconstruct a target structure with specific spatial arrangements on a 2.5D grid. This environment also supports the construction of complex compositional structures (*e.g., Build an X5 bridge by connecting a T3 and B5 vertically*).

We evaluate instruction-tuned LLMs on this task, prompting models to generate both first-order code (a sequence of atomic code) from multi-turn instructions, and higher-order code (modular functions) from single-turn natural language instructions. Results indicate that while models perform well on first-order code generation, they struggle with higher-order code, particularly when repetition is required in synthetic instructions. We also curated a dataset of human-written instructions and observed similar difficulties in producing correct higher-order code. With fine-tuning (Kranti et al., 2025b), the models show improved performance on human-written instructions that do not involve repetition, but they continue to struggle when repetition is required. These findings highlight both the potential and current limitations of LLMs in supporting modular, reusable program generation, suggesting the need for further research.

In addition to this, I was involved in developing tools to capture human-human interactions in collaborative structure-building tasks. I also developed interfaces for writing stand-alone instructions and replicating structures from them, enabling the study of both instruction generation and interpretation in isolation.

Future work will focus on integrating iterative feedback mechanisms, generating modular functions from multi-turn conversations, and evaluating the compositional capabilities of the models.

### 1.2 Collaborative Dialogue System

While the evaluation of LLMs for conversational programming was conducted in a virtual setup, I developed a collaborative dialogue system to better understand and test these capabilities in an interactive setting. At present,

[1] A 2D grid where components can also be stacked vertically, without the complexity of full 3D simulation.



*The 21th Annual Meeting of the Young Researchers' Roundtable on Spoken Dialogue Systems*, pages 6–9
Aug 28–29, 2025. 

user input is collected via text, but we plan to integrate a spoken dialogue interface in future iterations. The system includes a user interface (built using SLURK (Götze et al., 2022)) for receiving human instructions, a hybrid dialogue manager that integrates both RASA and LLM-based components, and a PyBullet [2]-based robot arm simulator for visualizing the execution of instructions. The system follows a modular approach to manage interactions with the LLM (for conversational programming) and RASA [3] (for dialogue control).

The system is designed to serve two purposes: (a) to collect human instructions for evaluating collaborative dialogue strategies, and (b) to compare the usability of our dialogue-based interface with traditional robot programming approaches, such as using teach pendants [4]. This work is currently in progress. The addition of speech could introduce challenges in ASR errors and interrupted speech, especially in noisy industrial settings.

While this system supports natural instruction and user studies, it currently does not address interactive repair strategies such as clarification or correction. These aspects are the focus of the next subsection.

### 1.3 Dealing with clarifications and corrections

In order to translate instructions into modular programs, the models used by the cobot must be capable of accurately inferring user intent. This includes distinguishing between different types of utterances, such as elaborations (*e.g., "stack it with a horizontal bridge; the bridge is placed in such a way that the structure looks like a T"*), acknowledgements (*e.g., "yes, good job!"*), which may not require code generation, and corrections (*e.g., "no, move that red washer to its right"*), which necessitate updating the code based only on the corrected instruction. Handling such utterances demands a nuanced understanding of conversational context and user intent.

In ongoing work, I am developing a simulation framework in which an instruction-giving LLM (to be later replaced by a human) communicates with an agent LLM responsible for generating executable code to build a goal structure. At each turn, the instruction-giver observes the current status and generates follow-up instructions or corrections. The agent LLM either generates code (if the instruction is clear) or requests clarification. This setup enables the simulation of diverse instruction styles and linguistic variations while keeping environmental conditions constant, thereby allowing for robust evaluation of collaborative behavior.

In parallel, we have also developed an evaluation framework (Kranti et al., 2025a) to assess dialogue systems under controlled interaction conditions. We evaluated this framework using a task-oriented dialogue setting based on the MultiWOZ (Budzianowski et al., 2018) dataset, providing insights into the robustness of different dialogue systems. These insights inform the design of collaborative dialogue systems, particularly with respect to the choice of instruction simulators, model sizes, and dialogue system architectures.

[2] https://pybullet.org/wordpress/

[3] `https://rasa.com/`

[4] https://en.wikipedia.org/wiki/Industrial_robot

## 2 Spoken dialogue system (SDS) research

In the era of LLMs, the future of SDS research is difficult to predict but looks promising. These models lower the barrier to creating culturally adaptive and accessible dialogue systems, while also raising important ethical and regulatory concerns.

SDS applications are expanding beyond traditional domains into areas like mental health support, creative learning, and industrial automation. This calls for interdisciplinary collaboration. Future efforts should focus on making SDSs robust to changes in underlying APIs and external dependencies. Improving automatic evaluation metrics and methodologies is also necessary, as they remain a major bottleneck.

Despite their impressive fluency, LLMs often rely on surface-level pattern matching, which may limit their ability to engage in deeper goal-driven, spoken dialogue interactions involving common ground. This makes it essential to investigate and mitigate these limitations to ensure safe and trustworthy dialogue systems. While curating human evaluation data for SDS remains resource-intensive, LLMs could be leveraged to help automate and scale this process. Additionally, improving the performance of smaller LLMs, especially in low-resource settings, will be important for making SDS systems widespread and inclusive.

## 3 Suggested topics for discussion

- Evaluation of Dialogue Quality: How can we automatically evaluate the quality of dialogues? What metrics or benchmarks are effective for meaningful assessment? How can such evaluations support the development of more advanced, possibly Turing-complete, dialogue systems?
- Multimodal Dialogue Systems: What are the implications of integrating additional modalities such as vision, gestures, or environmental context? When is multimodality essential, and how can we assess its effectiveness and necessity in various applications?
- Necessity of Architecture in the Age of LLMs: With LLMs now capable of solving a wide range of downstream tasks, is there still a need for specialized architectures in dialogue systems? If so, what should they look like, and how can we design them?

## References


Jessica Van Brummelen, Kevin Weng, Phoebe Lin, and Catherine Yeo. 2020. Convo: What does conversational programming need? an exploration of machine learning interface design. *CoRR* abs/2003.01318. https://arxiv.org/abs/2003.01318.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, pages 5016–5026. https://aclanthology.org/D18-1547/.

Sarah Fakhoury, Aaditya Naik, Georgios Sakkas, Saikat Chakraborty, and Shuvendu K. Lahiri. 2024. Llm-based test-driven interactive code generation: User study and empirical evaluation. *IEEE Trans. Software Eng.* 50(9):2254–2268. https://doi.org/10.1109/TSE.2024.3428972.

Jana Götze, Maike Paetzel-Prüsmann, Wencke Liermann, Tim Diekmann, and David Schlangen. 2022. The slurk interaction server framework: Better data for better dialog models. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*. European Language Resources Association, pages 4069–4078. https://aclanthology.org/2022.lrec-1.433.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a minecraft dialogue. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, pages 2589–2602. https://doi.org/10.18653/V1/2020.ACL-MAIN.232.

Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. 2024a. Retrieval-augmented code generation for situated action generation: A case study on Minecraft. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, pages 11159–11170. https://doi.org/10.18653/v1/2024.findings-emnlp.652.

Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. 2024b. Towards no-code programming of cobots: Experiments with code synthesis by large code models for conversational programming. *CoRR* abs/2409.11041. https://doi.org/10.48550/ARXIV.2409.11041.

Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. 2025a. clem:todd: A framework for the systematic benchmarking of llm-based task-oriented dialogue system realisations. *CoRR* abs/2505.05445. https://doi.org/10.48550/ARXIV.2505.05445.

Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. 2025b. From templates to natural language: Generalization challenges in instruction-tuned llms for spatial reasoning. *arXiv preprint arXiv:2505.14425* .

Chris Madge and Massimo Poesio. 2024. Large language models as minecraft agents. *CoRR* abs/2402.08392. https://doi.org/10.48550/ARXIV.2402.08392.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in minecraft. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, pages 5405–5415. https://doi.org/10.18653/V1/P19-1537.

Maike Paetzel-Prüsmann, Julie Hunter, Kranti Chalamalasetti, Kate Thompson, Alexandros Nicolaou, Ozan Güngör, David Schlangen, and Nicholas Asher. 2022. Conversational Programming for Collaborative Robots. In *en ligne*. Philadelphia, United States, pages 1–5. 39th edition of the IEEE International Conference on Robotics and Automation (ICRA 2022). https://hal.science/hal-04220030.

Hossein A. Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. A survey on asking clarification questions datasets in conversational systems. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, pages 2698–2716. https://doi.org/10.18653/V1/2023.ACL-LONG.152.

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to execute actions or ask clarification questions. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz,

editors, *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Association for Computational Linguistics, pages 2060–2070. https://doi.org/10.18653/V1/2022.FINDINGS-NAACL.158.

## Biographical sketch

Chalamalasetti Kranti is a PhD student in the Foundations of Computational Linguistics lab, at the University of Potsdam, Germany, supervised by Prof. David Schlangen. Her research focuses on how collaborative robots can learn new skills through natural language interactions, particularly in industrial assembly and manufacturing environments.