
IGLU 2022: Interactive Grounded Language Understanding in a Collaborative Environment at NeurIPS 2022

Julia Kiseleva * Alexey Skrynnik* Artem Zholus* Shrestha Mohanty *
 Negar Arabzadeh * Marc-Alexandre Côté * Mohammad Aliannejadi
 Milagro Teruel Ziming Li Mikhail Burtsev Maartje ter Hoeve
 Zoya Volovikova Aleksandr Panov Yuxuan Sun Kavya Srinet Arthur Szlam
 Ahmed Awadallah
 julia.kiseleva@microsoft.com

Abstract

Human intelligence has the remarkable ability to quickly adapt to new tasks and environments. Starting from a very young age, humans acquire new skills and learn how to solve new tasks either by imitating the behavior of others or by following provided natural language instructions. To facilitate research in this direction, we propose *IGLU: Interactive Grounded Language Understanding in a Collaborative Environment*. The primary goal of the competition is to approach the problem of how to develop interactive embodied agents that learn to solve a task while provided with grounded natural language instructions in a collaborative environment. Understanding the complexity of the challenge, we split it into sub-tasks to make it feasible for participants. This research challenge is naturally related, but not limited, to two fields of study that are highly relevant to the NeurIPS community: Natural Language Understanding and Generation (NLU/G) and Reinforcement Learning (RL). Therefore, the suggested challenge can bring two communities together to approach one of the important challenges in AI. Another important aspect of the challenge is the dedication to perform a human-in-the-loop evaluation as a final evaluation for the agents developed by contestants.

Keywords

Natural Language Understanding (NLU), Reinforcement Learning (RL), Grounded Learning, Interactive Learning, Embodied RL.

1 Competition Description

1.1 Background and Impact

Humans have the remarkable ability to adapt to new tasks and environments quickly. Starting from a very young age, humans acquire new skills and learn how to solve new tasks either by imitating the behavior of others or by following provided natural language instructions [An, 1988, Council, 1999]. Studies in developmental psychology have shown evidence of natural language communication being an effective method for transmission of generic knowledge between individuals as young as infants [Csibra and Gergely, 2009]. This form of learning can even accelerate the acquisition of new skills by avoiding trial-and-error when learning only from observations [Thomaz et al., 2019]. Inspired by this, the AI research community attempts to develop grounded interactive embodied

*Leader Organizers

agents that are capable of engaging in natural back-and-forth dialog with humans to assist them in completing real-world tasks [Winograd, 1971, Narayan-Chen et al., 2017, Levinson, 2019, Chen et al., 2020, Abramson et al., 2020, ?, ?, ?]. Notably, the agent needs to understand when to initiate feedback requests if communication fails or instructions are not clear and requires learning new domain-specific vocabulary [Aliannejadi et al., 2020, 2021, Rao and Daumé III, 2018, Narayan-Chen et al., 2019, Jayannavar et al., 2020]. Despite all these efforts, the task is far from solved. For that reason, we propose the IGLU competition, which stands for Interactive Grounded Language Understanding (IGLU) in a collaborative environment.

Specifically, the goal of our competition is to approach the following scientific challenge: *How to build interactive embodied agents that learn to solve a task while provided with grounded natural language instructions in a collaborative environment?* By ‘interactive agent’ we mean that the agent can: (1) follow the instructions correctly, (2) ask for clarification when needed, and (3) quickly adapt newly acquired skills. The IGLU challenge is naturally related to two fields of study that are highly relevant to the NeurIPS community: Natural Language Understanding and Generation (NLU / NLG) and Reinforcement Learning (RL).

Relevance of NLU/G Natural language interfaces (NLIs) have been the “holy grail” of human-computer interaction for decades [Woods et al., 1972, Codd, 1974, Hendrix et al., 1978]. The recent advances in language understanding capability [Devlin et al., 2018, Liu et al., 2019, Clark et al., 2020, Adiwardana et al., 2020, Roller et al., 2020, Brown et al., 2020, Ouyang et al., 2022] powered by large-scale deep learning has led to a major resurgence of natural language interfaces such as virtual assistants, dialog systems, semantic parsing, and question answering systems etc [Liu and Lane, 2017, 2018, Dinan et al., 2020, Zhang et al., 2019]. Recent efforts have also focused on interactivity and continuous learning to enable agents to interact with users to resolve the knowledge gap between them for better accuracy and transparency. This includes systems that can learn new task from instructions [Li et al., 2020], assess their uncertainty [Yao et al., 2019], ask clarifying questions [Aliannejadi et al., 2021] and seek and leverage feedback from humans to correct mistakes [Elgohary et al., 2020].

Relevance of RL Recently developed RL methods celebrated successes for a number of tasks [Belle-mare et al., 2013, Mnih et al., 2015, 2016, Silver et al., 2017, Hessel et al., 2018, ?]. One of the aspects that helped to speed up RL methods development is game-based environments, which provide clear goals for an agent to achieve in flexible training settings. However, training RL agents that can follow human instructions has attracted fewer exploration [Chevalier-Boisvert et al., 2019, Cideron et al., 2019, Hu et al., 2019, Chen et al., 2020, Shu et al., 2017], due to complexity of the task and lack of proper experimental environments. Another interesting aspect of language in the context of language conditioned RL is its compositionality. Conditioning on language abstractions might help the agent to better understand visually different but semantically similar situations [Mu et al., 2022]. Shu et al. [2017] proposed a hierarchical policy modulated by a stochastic temporal grammar for efficient multi-task reinforcement learning where each learned task corresponds to a human language description in Minecraft environment. The BabyAI platform [Chevalier-Boisvert et al., 2019] aims to support investigations towards learning to perform language instructions with a simulated human in the loop. Chen et al. [2020] demonstrated that using step-by-step human demonstrations in the form of natural language instructions and action trajectories can facilitate the decomposition of complex tasks in a crafting environment.

Relevance to Real Live Scenarios and Societal Impact Our competition can drive positive impact for real-life scenarios such as:

- **Education:** *Minecraft: Education Edition*² is a game-based learning platform that promotes creativity, collaboration, and problem-solving in an immersive digital environment. As of 2021, educators in more than 115 countries are using Minecraft across the curriculum. As stated in Ayoub, adding AI elements to this educational platform will move its potential to a new level. AI applications have the power to become a great equalizer in education. Students can get personalized education and scaffolding while being less dependent on uncontrollable factors such as the quality of their teachers or the amount of help they receive from their caregivers.

²<https://education.minecraft.net/>

- **Robotics:** Bisk et al. [2016] proposed a protocol and framework for collecting data on human-robot interaction through natural language. The work demonstrated the potential for unrestricted contextually grounded communications between human and robots in blocks world. Developing robots to assist humans in different tasks at home has attracted much attention in the Robotics field [Stuckler et al., 2012]. In fact, the Robocup@Home³ and the Room-Across-Room⁴ have run for several years. Given that the main human-robot interaction is through dialog, and the robot is supposed to assist the human in multiple tasks, we envision IGLU to enable more effective task grounded dialog training between human and robots.

1.2 Novelty

Last year (@NeurIPS2021), we organized the first edition of the IGLU competition to tackle the task of grounded language understanding and interactive learning and bring together the NLU/G and RL research communities Kiseleva et al. [2021, 2022]. In light of the competition results, it is clear the task is far from being solved and requires more collective effort to explore promising directions. Hence, we now proposed the second edition of the IGLU competition.

Learning from the feedback we receives and from our own experiences running the IGLU in 2021, we made some significant changes for this year’s edition. This year our main focus is on *interactive agents*. To accommodate it, we made the following changes:

- *Simplifying data collection:* in order to increase the speed of the data collection and, subsequently, collect more data, we employ a single-turn data collection strategy instead of a multi-turn one (Section 1.3.1).
- *Speeding up training environment:* the critical change is the use of a new gridworld environment that allows for fast and scalable turnaround experiments while requiring a lower amount of memory and computation resources (Section 1.5.1)
- *Reformulating NLP task:* the NLP task is reformulated to tackle the generation of clarifying question that leads to truly interactive agents (Section 1.4.1).

1.3 Data

IGLU is partially motivated by the HCRC Map Task Corpus [Thompson et al., 1993], which consists of route-following multi-turn dialogs between an *Instruction Giver* (further called Architect) and a *Follower* (further called Builder). To enable data collection for IGLU we have developed a tool by leveraging the CraftAssist library, which can be easily plugged into crowdsourcing platforms, to collect the dataset of multi-turn interactions between Architect and Builder to extend the original one Jayannavar et al. [2020]. Here⁵ is a visualization of the collected interactions.⁶.

We modified the data collection strategy based on the lessons learned from last year’s contest. To make the contest appealing to a broader audience, we simplify the task by leveraging the data we collected last year. Namely, we focus on the setup where an interactive agent is dropped in the middle of an ongoing game where the structure is built partially (Fig. 1). Then, the agent is given a natural language instruction to be executed in the current world, and in case of uncertainty, they need to ask clarifying questions. Since the current data collection is limited to building-related tasks, the agent and builder are interchangeably used. We are working on enhancing the data collecting tool to allow agents to perform various tasks, such as ‘grab,’ ‘bring,’ etc., to convert a builder into an actor.

1.3.1 Data Collection

To extend the dataset collected during IGLU 2021⁷, we will follow a simplified procedure by requiring only one player per game (instead of two). The data collection task is as follows:

1. a player is dropped in the middle of an ongoing game, and they need to extend the structure in an imaginary way (Fig. 1: Left) within a certain time limit;

³<https://athome.robocup.org/>

⁴<https://ai.google.com/research/rxr/habitat>

⁵<https://youtu.be/Ls6Wv7EUDAO>

⁶The developed tool for data collection is used for IGLU 2022

⁷The previously agreed IRB is reused

- once the structure the player had in mind is done, they need to write down instructions on how to remodel what they just constructed. The instructions should be descriptive and sufficient as if they wanted to instruct someone else to redo what they have done (Fig. 1: Right).

We extend the collected single-turn dataset with clarifying questions by initializing the second data collection stage. The collected instructions are shown to players, and they are asked either to execute the instruction or ask clarifying questions when needed. The multi-turn data contains dialogs between the Architect and the Builder, the single-turn data contains the retrospective descriptions of what the Builder has done. Both types of data contains pairs of starting and ending gridworlds (per-turn). And, importantly, we also collect streams of an embodied behavior which were captured from human executing single or multi-turn tasks. These streams of events can be converted to the environment demonstration data using the data adaptation tool⁸ that we provide. We envision this data to be used in the role of the training data for imitation learning or reinforcement learning agents. An example of a behavior stream logs is present in Appendix D.

1.4 Tasks and Application Scenarios

The considered challenge of developing an ‘*interactive agent*’ naturally splits into two main tasks visualized in Fig. 1.4 suggests:

- NLP-related: to decide if the provided grounded instruction needs to be clarified and generate clarifying question if it’s a case;
- IR-related: to follow a clear grounded instruction in order to complete the described task.

Our vision suggests that the successful solutions to presented tasks can be combined into the end-to-end pipeline to develop desired interactive agents.

1.4.1 NLP Task: Asking Clarifying Questions

Inspired by Aliannejadi et al. [2021], we split the problem into the following concrete research questions:

RQ1 When to ask clarifying questions?

Given the instruction from the Architect, a model needs to predict whether that instruction is sufficient to complete the described task or further clarification is needed. Simply put, here it is decided if we need to activate the Builder.

RQ2 What clarifying question to ask?

If the given instruction from the Architect is ambiguous, a clarifying question should be raised. In this research question, we are specifically interested in “what to ask” to clarify the given instruction. The original instruction and its clarification can be used as an input for the Builder.

As a starting baseline we use implementation provided in Aliannejadi et al. [2021].

1.4.2 RL Task: Building Structures

Our experience of IGLU 2021 emphasized the complication of the builder task. Indeed, the optimal builder needs to know how to build any structure. The state-space of the possible cubes combinations is very large: (11, 11, 9) places where an agent can build a cube with six different colors. We believe that to solve such a challenging task, one needs a large-scale training procedure and the possibility

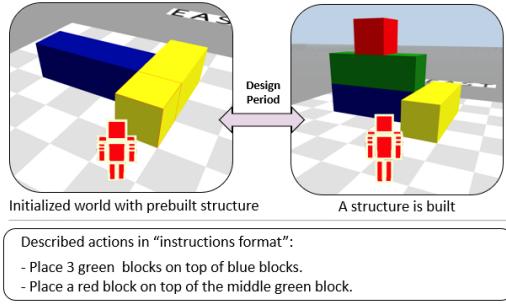


Figure 1: An example of data collection

can be converted to the environment demonstration data using the data adaptation tool⁸ that we provide. We envision this data to be used in the role of the training data for imitation learning or reinforcement learning agents. An example of a behavior stream logs is present in Appendix D.

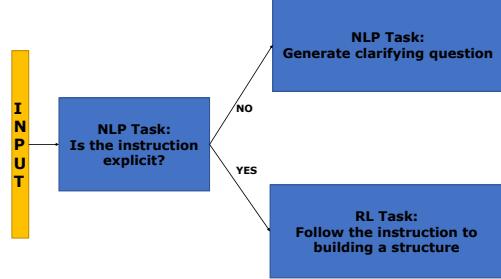


Figure 2: IGLU’s general overview

⁸<https://github.com/iglu-contest/iglu/blob/master/iglu/data/>

to run a spectrum of various experiments. Unfortunately, the last part was restricted by the slow sampling rate of the Malmo environment Kanervisto et al. [2022]. Thus, for IGLU 2022 we worked in two directions: speeding up the environment and creating a baseline, which can be generalized to build various structures.

1.5 Baselines, Code, and Material Provided

1.5.1 IGLU Gridworld RL Environment

For the builder task, we created a novel RL environment⁹ where an embodied agent can build spatial structures of blocks of different colors. The agent’s goal is to complete a task expressed as an instruction written in natural language.

We implement the environment in pure Python using a simplified version of an open-source Minecraft clone¹⁰. The renderer is decoupled from the core environment, it can run headless with GPU-acceleration, and can be disabled. The simplicity of the environment makes it fast and scalable for highly loaded RL experiments. Without a renderer, the environment runs at 17000 steps-per-second (SPS) and at 4000 SPS with rendering enabled, which is 85 (for vector observations) and 20 (for image-based observations) times faster than the previous version of the IGLU @2021 environment. Such significant speedup is possible by removing the complexity of using Minecraft, which comes with a massive amount of (for now) unused features. The latest version, in turn, is as simple as it is possible, that is why it is much faster.

The observation space consists of point-of-view (POV) image (64, 64, 3), inventory item counts (6,), a snapshot of the building zone (11, 11, 9), and the agent’s (X, Y, Z) position with pitch and yaw angles (5,). The building zone is represented via a 3D tensor with block ids (e.g., 0 for air, 1 for blue block, etc.). The agent can navigate over the building zone, place and break blocks, and switch between block types. Additionally, the environment provides a dialog from the dataset, which defines a task for building. The dialog is split into two parts: the first one is a context for the task, and the second one defines the target. The context utterances define blocks that are placed before the beginning of the episode. We thus initialize the episode with such blocks placed at their correct positions. Target utterances define the rest of the blocks needed to be added. The action space combines discrete and continuous subspaces: a discrete space of 14 actions: *noop*, *step forward*, *step backward*, *step right*, *step left*, *jump*, *brake block*, *place block*, *choose block type 1-6* and two-dimensional continuous camera movement space.

The reward in the environment reflects how complete the built structure is agnostic to its location. We run a convolution-like procedure between the state grid and the target grid. We calculate the per-block intersection for each alignment and then maximize it over alignments. The pseudocode for the reward calculation is presented in Appendix A.

1.5.2 Builder Baseline

Training an agent to build any user-defined structure is a challenging task. Even if the task given to the agent is not described in natural language (e.g., by providing the target structure of the grid), it is still nontrivial. Thus, in addition to last year’s single-task baselines¹¹¹², we are providing a multi-task one. We focus on an approach that can solve multiple problems for that new baseline, including problems not seen during the training phase. The whole builder problem can be divided into two

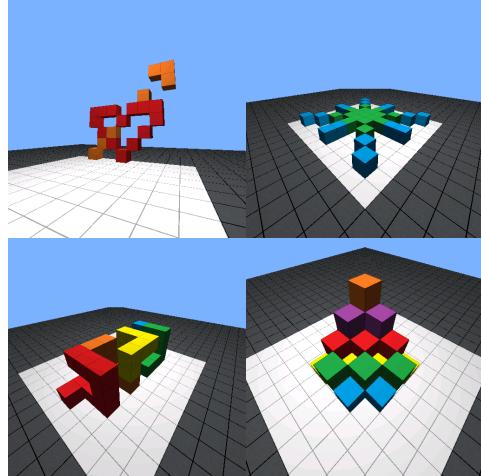


Figure 3: Examples of target structures rendered by the gridworld environment

⁹<https://github.com/iglu-contest/gridworld>

¹⁰<https://github.com/fogleman/Minecraft>

¹¹<https://github.com/iglu-contest/iglu-builder-baseline-dreamer>

¹²<https://github.com/iglu-contest/iglu-builder-baseline-rllib>

main parts, which could be solved by NLU and RL methods (see Fig. 5 in Appendix). This section provides an approach to building structure given a target grid.

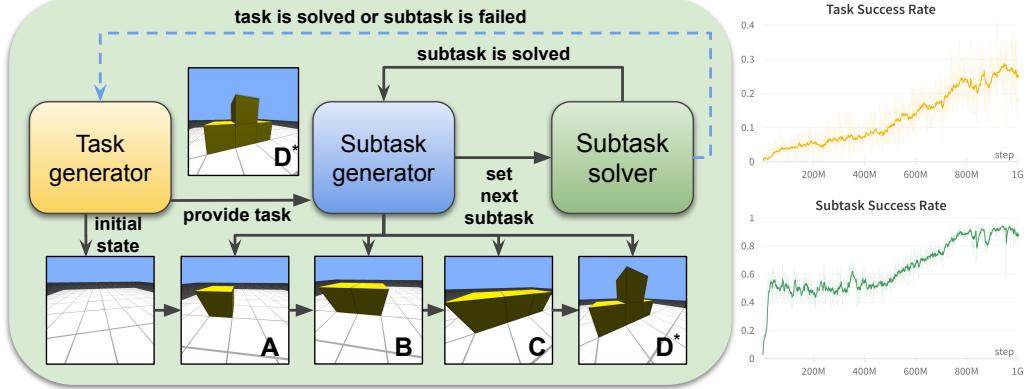


Figure 4: Scheme of the multi-task research baseline. The agent is trained to build any structure using a Task generator and heuristic approach, which determines the following block to place for the agent (Subtask generator). The episode is restricted to adding or removing only a single cube, which significantly simplifies the global task. Training curves for the PPO are presented on the right. The agent solves subtask in almost 90% of episodes and solves the whole task in 25% of trials.

There are many ways to build a structure in IGLU environment. To overcome this, we split the whole system from the agents’ perspective into three modules: a task generator module, a subtask generator module, and a subtask solving module (see left part of Fig. 4). We define the subtask as an episode of adding or removing a single cube. It allows us to train an agent with a dense reward signal in episodes with a short horizon.

The right part of Fig. 4 reports the learning curves of the PPO algorithm. The agent was trained for one billion steps using the Sample Factory framework Petrenko et al. [2020] on a single GPU machine in 15 hours. We use randomly generated compact structures as tasks and a subtask generating module. This module determines the order of blocks for subtasks. In this experiment, we used a heuristic one, which samples the next cube sequentially going through the 3D voxel of a target, but this module can be trained with a learning model. The baseline uses vector-based observation space (environment without rendering) and shows the training framerate of 19000 SPS. The PPO input is the agent’s position, a compass angle, and target/current grid (3D voxels).

To train the agent, we design a reward function, which defines the reward signal based on the Manhattan distance of the placed cube to the target position (see Table 1). In addition, the agent receives an extra reward, which is equal to 0.5 if the agent puts the cube in the right place, under his feet. This addition sufficiently increased the agent’s performance in tasks with high structures.

Table 1: The design of the reward function is based on Manhattan distance. The agent would receive a negative reward if the distance of the placed cube from the target was more than three blocks.

Distance	0	1	2	3	4	5	$i \in \{6, +\infty\}$
Reward	1	0.25	0.05	0.001	-0.0001	-0.001	$-0.01 \times (i - 5)$

1.6 Metrics

NLP Task We evaluate this task in two distinct steps. The first step is a classification problem on whether we need clarification question or not. To compare the performance, we rely on precision, recall, F_1 score, and accuracy. Aliannejadi et al. [2021] used a similar approach for open-dialogue question-answering and noted that it is rather difficult and potentially can benefit from more exploration. For the second research question (i.e., “what to ask”), we evaluate how closely the generated questions resemble the human issued clarifying questions. We adopt standard word-overlap metrics such as BLEU Papineni et al. [2002] score as well as word embedding based metrics that captures the

contextualized and semantic similarity of the generated clarifying questions w.r.t the question issued by human annotators Liu et al. [2016].

RL Task For the Builder task, the main metric is F_1 score where the “ground truth” is the target structure (a 3d tensor) and the “prediction” is the snapshot of the building zone at the end of the episode (also a 3d tensor). This is different from IGLU 2021 where we made an additional maximization over timesteps of the metric. That maximization led to a bias towards specific types of behavior (e.g., uniformly random over actions). This year, we neglect this bias by reporting the metric at the last step of the episode. The episode terminates either when the structure is complete, or when the time limit is reached. The case when the episode ends at time limit is outside of the control of an agent. Therefore, we let the agent to decide when to end an episode (as a separate action). We report the performance by calculating the intersection between the last step snapshot of the building zone and the target zone. This value is then used to calculate the F_1 score.

Additionally, to provide more information to participants, we provide a skill labeling of tasks. Each label indicate a specific skill that the agent should possess in order to build a particular structure. For example, a structure that has blocks (or groups of block) that fly in the air without touching any other blocks side-by-side. This kind of structures requires to understand that one should place a supporting block and eventually remove it (as block placement is done side-by-side). We believe this information could help the participants to better understand their agents’ capabilities. At the same time, the final metric is not related to the skills, it is just an average taken over all tasks. The whole set of highlighted skills is present in Appendix C.

1.7 Website, Tutorial and Documentation

Competition website is hosted at iglu-contest.net. The main website contains links to all materials such as the competition slack channel, documentation for the RL environment¹³, the Codalab competitions links. A dedicated email address for the competition is info@iglu-contest.net. To ease reaching out the participants, we created a twitter account dedicated to the competition announcements¹⁴.

2 Resources

2.1 Organizing Team

Our team can be split into two main groups based on expertise: NLU/G and RL researchers.

The RL sub-team includes the following members:

Artem Zholus is an incoming summer research intern at EPFL and a Master’s student at MIPT. His research focuses on model-based reinforcement learning with application to robotics. Artem has worked on many machine learning projects, such as reinforcement learning for drug discovery and generative models. In this project, he leads the RL environment development and contributes to the data collection, baselines development, and the competition infrastructure development.

Zoya Volovikova is a Master’s student at ITMO. Zoya worked on the development of scientific software using machine learning algorithms. As an example, a system for predicting the phase of the wavefront for an adaptive optical system. In the IGLU contest, she takes part in the development of RL baselines.

Alexey Skrynnik is an individual contributor. His current research focused on multi-agent, object oriented and hierarchical reinforcement learning. He is highly experienced in practical RL. Moreover, Alexey previously won MineRL Diamond Competition 2019 as a leader of the CDS team. In the IGLU contest, he leads the development of RL baselines.

¹³<https://iglu-contest.github.io/>

¹⁴<https://twitter.com/IgluContest>

Aleksandr Panov has managed the CDS team that won the MineRL Diamond Competition 2019. Currently, Aleksandr works in model-based RL and the use of RL methods in the task of navigation in indoor environments.

Marc-Alexandre Côté is a Principal Researcher at Microsoft Research Montréal. His research interests lie at the intersection of reinforcement learning and language understanding. He leads the Microsoft TextWorld project which aims at developing new RL agents capable of navigating and interacting with text environments (e.g., text-based adventure games). Such agents should possess skills like reading and understanding natural language text, information/knowledge-gathering, planning, dealing with combinatorially large action-space. In terms of past experiences organizing competition, in 2019 Marc initiated the First TextWorld Problem competition presented at CoG 2019 as a way of promoting the TextWorld framework.

The NLU/G sub-team consists of the following members:

Julia Kiseleva is a Senior Researcher at Microsoft Research. Her research interests are in natural language processing, information retrieval, and machine learning, with a strong focus on continuous learning from user feedback and interactions. Julia has been involved in many initiatives such as a series of Search-Oriented Conversational AI (SCAI) workshop, which was continuously organized from 2015 - 2020. She also co-organized the recent challenge ConvAI3: Clarifying Questions for Open-Domain Dialogue Systems (ClariQ) at EMNLP2020.

Negar Arabzadeh is a Research Scientist at Spotify Research working on podcast retrieval evaluation. She is also completing her PhD studies at the University of Waterloo. Her research is aligned with Ad hoc Retrieval and Conversational Search in IR and NLP domains.

Shrestha Mohanty is a Machine Learning Engineer at Microsoft. She primarily works in areas of machine learning, deep learning, and natural language processing, including topics such as personalization, dialogue systems and multilingual language models. She is also interested in and has worked on problems at the intersection of machine learning and healthcare. Prior to Microsoft, she completed her master's in information science from University of California, Berkeley.

Mohammad Aliannejadi is an assistant professor at the University of Amsterdam (The Netherlands). His research interests include single- and mixed-initiative conversational information access and recommender systems. Previously, he completed his Ph.D. at Università della Svizzera italiana (Switzerland), where he worked on novel approaches of information access in conversations. He co-organized the Conversational AI Challenge, ConvAI 3 (EMNLP 2020).

Ziming Li is Research Scientist in Amazon Alexa, and he was previously a Postdoc at University of Amsterdam. His main research interest is developing advanced dialogue systems, including dialogue policy learning and evaluation. He is also interested in the fields of conversational search and reinforcement learning.

Maartje ter Hoeve is a PhD candidate at the University of Amsterdam. Her main research interest is how we can learn from humans and their cognition to improve our NLP and IR systems. She has a background in both Linguistics and Artificial Intelligence. She co-organized the previous IGLU challenge (NeurIPS 2021).

Mikhail Burtsev's current research interests: application of neural nets and reinforcement learning in the NLP domain. He was a faculty advisor of MIPT team participating in Alexa Prize Challenges 3 and 4. He proposed and organised Conversational AI Challenges: ConvAI 1 (NIPS 2017), ConvAI 2 (NeurIPS 2018), ConvAI 3 (EMNLP 2020).

Yuxuan Sun is a Research Engineer in Facebook AI Research (FAIR). His research interests lie in natural language processing, neural symbolic learning and reasoning, and human-in-the-loop learning for embodied agents.

Arthur Szlam is a Research Scientist at Facebook AI Research. He works on connecting perception, memory, language, and action in artificial agents. Prior to joining Facebook, he was on the faculty of the City College of New York (CCNY), and was the recipient of a Sloan research fellowship. He has been a co-organizer of previous ConvAI challenges.

Kavya Srinet is a Research Engineering Manager at Facebook AI Research working towards a long-term goal of developing interactive assistants. She works on building assistants that can learn from interactions with humans. Prior to FAIR, she was a Machine Learning Engineer at the AI Lab at Baidu Research, where she worked on speech and NLP problems. Kavya was at the Allen Institute for Artificial Intelligence for a summer before that working on learning to rank for Semantic Scholar. Kavya did her graduate school from Language Technology Institute at Carnegie Mellon University, where she worked on areas of machine translation, question answering and learning to rank for graphs and knowledge bases.

Milagro Teruel is a Research Software Engineer at Microsoft Research. She works in developing new platforms to facilitate research and evaluation of machine learning models from human interactions in natural language. Previous to joining Microsoft, Milagro has worked in learning complex relations in text, such as argument mining on legal documents and Covid-19 diagnosis from medical histories, with focus on data collection in low-resource scenarios.

Ahmed Awadallah is a Senior Principal Research Manager at Microsoft Research where he leads the Language & Information Technologies Group. His research has sought to understand how people interact with information and to enable machines to understand and communicate in natural language (NL) and assist with task completion. More recently, his research has focused learning from limited annotated data (e.g., few-shot learning and transfer learning) and from user interactions (e.g. interactive semantic parsing). Ahmed's contributions to NLP and IR have recently been recognized with the 2020 Karen Spärck Jones Award from the British Computer Society. Ahmed regularly serves as (senior) committee, area chair, guest editor and editorial board member at many major NLP and IR conferences and journals.

2.2 Advisory Board

Tim Rocktäschel is an Associate Professor at the Centre for Artificial Intelligence in the Department of Computer Science at University College London (UCL), where he is leading the UCL Deciding, Acting, and Reasoning with Knowledge (DARK) Lab. He is also a Scholar of the European Laboratory for Learning and Intelligent Systems (ELLIS). He was a Manager and Research Scientist at Facebook AI Research (FAIR), a Postdoctoral Researcher in Reinforcement Learning at the Whiteson Research Lab at the University of Oxford, a Junior Research Fellow in Computer Science at Jesus College, and a Stipendiary Lecturer in Computer Science at Hertford College. Tim obtained his Ph.D. from UCL under the supervision of Sebastian Riedel, where he was awarded a Microsoft Research Ph.D. Scholarship in 2013 and a Google Ph.D. Fellowship in 2017. Tim's research focuses on Reinforcement Learning and Open-ended Learning.

Julia Hockenmaier is an associate professor at the University of Illinois at Urbana-Champaign. She has received a CAREER award for her work on CCG-based grammar induction and an IJCAI-JAIR Best Paper Prize for her work on image description. She has served as member and chair of the NAACL board, president of SIGNLL, and as program chair of CoNLL 2013 and EMNLP 2018.

Katja Hofmann is a Senior Principal Researcher at the Machine Intelligence and Perception group at Microsoft Research Cambridge. Her research focuses on reinforcement learning with applications in video games, as she believes that games will drive a transformation of how people interact with AI technology. She is the research lead of Project Malmo, which uses the popular game Minecraft as an experimentation platform for developing intelligent technology, and has previously co-organized two competitions based on the Malmo platform. Her long-term goal is to develop AI systems that learn to collaborate with people, to empower their users and help solve complex real-world problems.

Bill Dolan is Partner Researcher Manager at Microsoft Research, where he manages the Natural Language Processing group. He has worked on a wide variety of problems, including the acquisition

of structured common-sense knowledge from free text, paraphrasing, text rewriting to improve grammar and style, and most recently on data-driven, grounded approaches to handling dialog. He has helped organize a number of research community events over the years, including the RTE challenges and the “Understanding Situated Language in Everyday Life” summer research institute with the University of Washington, as well as running the Microsoft Research graduate fellowship program from 2014-2017.

Ryen W. White is a Partner Research Area Manager at Microsoft Research, where he leads the Language and Intelligent Assistance research area. He led the applied science organization at Microsoft Cortana, and he was chief scientist at Microsoft Health. Ryen has authored hundreds of publications in areas such as information retrieval, computational health, and human-computer interaction - including many that received awards. He was program chair for SIGIR 2017 and The Web Conference 2019. Ryen is editor-in-chief of ACM Transactions on the Web.

Maarten de Rijke is Distinguished University Professor of Artificial Intelligence and Information Retrieval at the University of Amsterdam. He is VP Personalization and Relevance and Senior Research Fellow at Ahold Delhaize. His research strives to build intelligent technology to connect people to information. His team pushes the frontiers of search engines, recommender systems and conversational assistants. They also investigate the influence of the technology they develop on society. De Rijke is the director of the Innovation Center for Artificial Intelligence.

Oleg Rokhlenko is a Senior Science Manager at Amazon Alexa Shopping, leading a science team specializing in Natural Language Processing (NLP) and Conversational AI for voice-activated shopping assistant agents. Before joining Amazon, Oleg was a Senior Research Scientist in Yahoo Labs, focusing on ML, NLP and Community Question Answering. Prior to that, Oleg was a Research Staff Member in IBM Research. Oleg holds a PhD from the Technion - Israel Institute of Technology and has numerous publications in the top-tier conferences in the relevant fields. Oleg also serves on the program committee of the leading ML, NLP, and IR conferences such as NIPS, ICML, ACL, EMNLP, NAACL, SIGIR, WSDM, The Web Conference, and more.

Sharada Mohanty is the CEO and Co-founder of AIcrowd, a community of AI researchers built around a platform encouraging open and reproducible artificial intelligence research. He was the co-organizer of many large-scale machine learning competitions, such as NeurIPS 2017: Learning to Run Challenge, NeurIPS 2018: AI for Prosthetics Challenge, NeurIPS 2018: Adversarial Vision Challenge, NeurIPS 2019: MineRL Competition, NeurIPS 2019: Disentanglement Challenge, NeurIPS 2019: REAL Robots Challenge, NeurIPS 2020: Flatland Competition, NeurIPS 2020: Progen Competition. He is extremely passionate about benchmarks and building communities. During his Ph.D. at EPFL, he worked on numerous problems at the intersection of AI and health, with a strong interest in reinforcement learning. In his previous roles, he has worked at the Theoretical Physics department at CERN on crowdsourcing compute for PYTHIA powered Monte-Carlo simulations; he has had a brief stint at UNOSAT building GeoTag-X—a platform for crowdsourcing analysis of media coming out of disasters to assist in disaster relief efforts. In his current role, he focuses on building better engineering tools for AI researchers and making research in AI accessible to a larger community of engineers.

References

Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, et al. Imitating interactive intelligence. [arXiv preprint arXiv:2012.05672](https://arxiv.org/abs/2012.05672), 2020.

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. [arXiv preprint arXiv:2001.09977](https://arxiv.org/abs/2001.09977), 2020.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq). 2020.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. Building and evaluating open-domain dialogue corpora with clarifying questions. pages 4473–4484, 2021.

Meltzoff An. Imitation, Objects, Tools, and the Rudiments of Language in Human Ontogeny, February 1988. URL <https://pubmed.ncbi.nlm.nih.gov/23997403/>. ISSN: 0393-9375 Issue: 1-2 Publisher: Hum Evol Volume: 3.

Dan Ayoub. Unleashing the power of AI for education. URL <https://www.technologyreview.com/2020/03/04/905535/unleashing-the-power-of-ai-for-education/>.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. Natural language communication with robots. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 751–761, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1089. URL <https://www.aclweb.org/anthology/N16-1089>.

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. Language models are few-shot learners. 2020.

Valerie Chen, Abhinav Gupta, and Kenneth Marino. Ask your humans: Using human instructions to improve generalization in reinforcement learning. *arXiv preprint arXiv:2011.00517*, 2020.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop. In International Conference on Learning Representations, 2019.

Geoffrey Cideron, Mathieu Seurin, Florian Strub, and Olivier Pietquin. Self-educated language agent with hindsight experience replay for instruction following. *arXiv preprint arXiv:1910.09451*, 2019.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Edgar F Codd. Seven steps to rendezvous with the casual user. IBM Corporation, 1974.

National Research Council. How People Learn: Brain, Mind, Experience, and School: Expanded Edition. August 1999. ISBN 978-0-309-07036-2. doi: 10.17226/9853. URL <https://www.nap.edu/catalog/9853/how-people-learn-brain-mind-experience-and-school-expanded-edition>.

Gergely Csibra and György Gergely. Natural pedagogy. *Trends in cognitive sciences*, 13(4):148–153, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 2018.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. The second conversational intelligence challenge (convai2). In The NeurIPS'18 Competition, pages 187–208. Springer, Cham, 2020.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. Speak to your parser: Interactive text-to-SQL with natural language feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2065–2077, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.187. URL <https://www.aclweb.org/anthology/2020.acl-main.187>.

Gary G Hendrix, Earl D Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems (TODS)*, 3(2):105–147, 1978.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. Hierarchical decision making by generating and following natural language instructions. *arXiv preprint arXiv:1906.00744*, 2019.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.232. URL <https://www.aclweb.org/anthology/2020.acl-main.232>.

Anssi Kanervisto, Stephanie Milani, Karolis Ramanauskas, Nicholay Topin, Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, Wei Yang, Weijun Hong, Zhongyue Huang, Haicheng Chen, Guangjun Zeng, Yue Lin, Vincent Michel, Eloi Alonso, François Fleuret, Alexander Nikulin, Yury Belousov, Oleg Svidchenko, and Aleksei Shpilman. Minerl diamond 2021 competition: Overview, results, and lessons learned, 2022. URL <https://arxiv.org/abs/2202.10583>.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, Arthur Szlam, Yuxuan Sun, Katja Hofmann, Michel Galley, and Ahmed Awadallah. Neurips 2021 competition iglu: Interactive grounded language understanding in a collaborative environment. *arXiv preprint arXiv:2110.06536*, 2021.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, et al. Interactive grounded language understanding in a collaborative environment: Iglu 2021. *arXiv preprint arXiv:2205.02388*, 2022.

Stephen C Levinson, Tom m. mitchell, simon garrod, john e. laird, stephen c. levinson, and kenneth r. koedinger. *Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions*, 26:9, 2019.

Toby Jia-Jun Li, Tom Mitchell, and Brad Myers. Interactive task learning from GUI-grounded natural language instructions and demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, July 2020.

Bing Liu and Ian Lane. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE, 2017.

Bing Liu and Ian Lane. Adversarial learning of task-oriented neural dialog models. In *Proceedings of the SIGDIAL 2018 Conference*, pages 350–359, 2018.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL <http://arxiv.org/abs/1907.11692>.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktaschel, and Edward Grefenstette. Improving intrinsic exploration with language abstractions, 2022. URL <https://arxiv.org/abs/2202.08938>.

Anjali Narayan-Chen, Colin Gruber, Mayukh Das, Md Rakibul Islam, Soham Dan, Sriraam Natarajan, Janardhan Rao Doppa, Julia Hockenmaier, Martha Palmer, and Dan Roth. Towards problem solving agents that communicate and learn. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 95–103, 2017.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. Collaborative dialogue in minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, 2019.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Aleksii Petrenko, Zhehui Huang, Tushar Kumar, Gaurav Sukhatme, and Vladlen Koltun. Sample factory: Egocentric 3d control from pixels at 100000 fps with asynchronous reinforcement learning. In *International Conference on Machine Learning*, pages 7652–7662. PMLR, 2020.

Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *arXiv preprint arXiv:1805.04655*, 2018.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

Tianmin Shu, Caiming Xiong, and Richard Socher. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *arXiv preprint arXiv:1712.07294*, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Jorg Stuckler, Dirk Holz, and Sven Behnke. Robocup@ home: Demonstrating everyday manipulation skills in robocup@ home. *IEEE Robotics & Automation Magazine*, 19(2):34–42, 2012.

Andrea L Thomaz, Elena Lieven, Maya Cakmak, Joyce Y Chai, Simon Garrod, Wayne D Gray, Stephen C Levinson, Ana Paiva, and Nele Russwinkel. Interaction for task instruction and learning. In *Interactive task learning: Humans, robots, and agents acquiring new tasks through natural interactions*, pages 91–110. MIT Press, 2019.

Henry S. Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. The HCRC Map Task corpus: natural dialogue for speech recognition. In *Proceedings of the workshop on Human Language Technology - HLT '93*, page 25, Princeton, New Jersey, 1993. Association for Computational Linguistics. ISBN 978-1-55860-324-0. doi: 10.3115/1075671.1075677. URL <http://portal.acm.org/citation.cfm?doid=1075671.1075677>.

Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, 1971.

W. A. Woods, Ronald M Kaplan, and Bonnie L. Webber. The lunar sciences natural language information system: Final report. BBN Report 2378, 1972.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. Model-based interactive semantic parsing: A unified framework and a text-to-SQL case study. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5447–5458, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1547. URL <https://www.aclweb.org/anthology/D19-1547>.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536, 2019.

Appendix

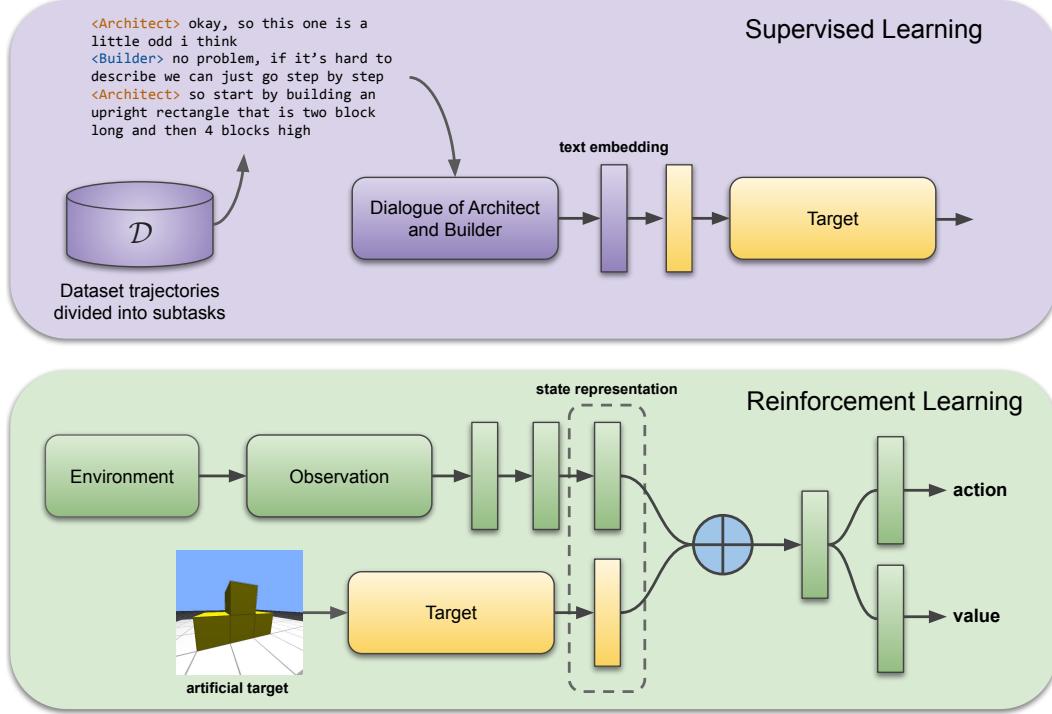


Figure 5: The decomposition scheme of silent builder. There two main parts, which should be solved separately using NLU and RL methods.

A Maximal Intersection Calculation Scheme

We calculate reward using the following algorithm. Effectively, at each step, we calculate the maximal intersection between the current and the target grids as described in Algorithm 1. The reward is then defined in terms of maximal intersections at two consecutive steps, that is

$$r_t \leftarrow \max_{\text{int}} - \max_{\text{int}_{t-1}}.$$

Algorithm 1 shows a principle version of the intersection calculation. In practice, we optimize it as it takes a lot of time to compute it using the naive version. First, we recalculate the intersection size only when we observe a change in the building zone (i.e., one block added or removed). Second, there can be some translation configuration such that the full intersection is not possible (e.g., for a goal structure with block in the four corners of the building zone, if the agent has placed a block in the center of the zone, then this means it would be unable to have full intersection since any non-zero shift would cut out at least one of these blocks). Therefore, we consider only those intersections that do not cut the target grid.

B Competition Rules

B.1 Silent Builder

1. Submissions to the IGLU competition must be open. We expect Teams to reveal details of their method, including source-code and machine learning model parameters.
2. A team can submit separate entries to both (Clarifying Questions and Builder) tracks; performance in the tracks will be evaluated separately. Submissions between the phases are not linked in any way.

Algorithm 1: Maximal Intersection Pseudocode

Input :current: Current build zone snapshot, 3d tensor (y, x, z) filled with integers in range [0, 6];
target: Target build zone, 3d tensor (y, x, z) filled with integers in range [0, 6];
X, Y, Z: sizes of the build zone over three axes

Output:max_int: Scalar intersection size

maximum $\leftarrow 0$;

// Iterate over rotations

for $i \in [1..4]$ do

 // Iterate over translations

 for $dx \in [-X..X]$ do

 for $dz \in [-Z..Z]$ do

 shifted_grid $\leftarrow translate(current, dx, dz)$;

 intersection $\leftarrow (shifted_grid = target) \wedge (target \neq 0)$;

 maximum $\leftarrow \max(maximum, intersection)$;

 end

 end

 current $\leftarrow rotate_90degrees(current)$;

end

return maximum

3. Interaction with the submission system must be through the provided interface, in particular through the “step” function. Only the iglu gym environment provided by organizers can be used. Additional information may not be extracted from the simulator in any way (unless the opposite stated for a specific phase).
4. Submissions will be evaluated in IGLUSilentBuilderVisual-v0 or IGLUSilentBuilder-v0 depending on a phase.
5. In IGLUSilentBuilderVisual-v0 only image, chat, compass, and inventory are exposed. The use of any other fields or other environment info is prohibited during the evaluation.
6. In IGLUSilentBuilder-v0 the allowed observation components are image, chat, compass, inventory, agent position, current grid
7. For training machine learning models, any information, including the agent position and the grid state can be used.
8. The trained machine learning models can use either a public dataset from the Minecraft Dialogue or public IGLU dataset or both of them. If any other sources of data are used, they should also be publicly available.
9. The submission to the Builder competition can use a trained machine learning model or the hardcoded heuristic-based or rule-based agents.
10. Participants do not have the access to the test examples which their agents are evaluated on. Any unauthorized access to the information that can reveal the test tasks contents will lead to disqualification.
11. The use of internet will be turned off for a submission to avoid cheating. Every downloadable data (e.g., packages, datasets, neural networks weights) should be included in submission or installed in the image.

C Preliminary Set of Skills in the Builder Track

A preliminary set of agent skills which are needed to build a structure is provided below. Note that these labels do not correspond to structures themselves but to a behavior (and an instruction) that led to a particular structure.

- **flat:** flat structure with all blocks on the ground
- **flying:** there are blocks that cannot be placed without removing some other blocks (i.e. one block or a group of blocks are not connected to the ground)

- **tricky**: some blocks are hidden or there should be a specific order in which they should be placed (e.g., a 27 blocks cube with the block in the center of different color than the whole cube)
- **tall**: a structure cannot be built without the agent being at a sufficient height (the placement radius is 3 blocks)

Each submission is evaluated on a set of tasks with certain skill labels associated with each task. The platform will then respond with a per skill performance that could help participants to better understand the performance of their submission.

D Example of The Behavior Data

In this section we show an example of the raw behavioral data represented as an event stream in the JSON format.

```

1  {'gameId': 19,
2   'stepId': 1,
3   'avatarInfo': {'pos': [0.16371048991651502,
4     66.19999999999993,
5     5.601672092120345],
6   'look': [-0.6139999999999998, 0.0019999999999995]},
7   'worldEndingState': {'blocks': [[0, 64, 1, 50],
8     [1, 64, 1, 50],
9     [3, 63, 1, 59],
10    [3, 65, 1, 59],
11    [3, 66, 1, 59],
12    [4, 63, 1, 59],
13    [4, 65, 1, 59],
14    [5, 63, 1, 59],
15    [5, 64, 1, 59],
16    [5, 65, 1, 59],
17    [5, 66, 1, 59]]},
18   'clarification_question': 'null',
19   'tape':
20     action start_recover_world_state
21     action select_and_place_block 59 3 63 1 6.93311074709203e-24
22       64.12000274658203 4 -1.733277824122399e-24 0 -1
23     action select_and_place_block 59 3 65 1 6.93311074709203e-24
24       64.12000274658203 4 -1.733277824122399e-24 0 -1
25     ...
26     action select_and_place_block 59 5 66 1 6.93311074709203e-24
27       64.12000274658203 4 -1.733277824122399e-24 0 -1
28   block_change (3, 63, 1, 0, 59) (3, 65, 1, 0, 59) (3, 66, 1,
29     0, 59) (4, 63, 1, 0, 59) (4, 65, 1, 0, 59) (5, 63, 1, 0,
30     , 59) (5, 64, 1, 0, 59) (5, 65, 1, 0, 59) (5, 66, 1, 0,
31     59)
32   pos_change (0.7800644185765344, 66.83999999999992, 3.3853728
33     9325922)
34   set_look (-0.40799999999997, -0.3500000000000002)
35   action finish_recover_world_state
36   ...
37   action step_backward
38   pos_change (0.780392971263885, 66.83999999999992, 3.46751062
39     41016485)
40   set_look (-0.756, 0.0059999999999988)
41   pos_change (0.7812531261284168, 66.83999999999992, 3.6108680
42     43629237)
43   set_look (-0.758, 0.0059999999999988)
44   ...

```

```
36     action step_right
37         pos_change (0.8616502838004294, 66.83999999999992, 5.3805996
38             39872034)
39         set_look (-0.758, -0.032000000000000126)
40         pos_change (1.004321268291321, 66.83999999999992, 5.40441797
41             97943865)
42         set_look (-0.754, -0.032000000000000126)
43         ...
44         pos_change (1.861788332305804, 65.39999999999995, 4.27719126
45             9822069)
46         action select_and_place_block 50 1 64 1 2.2324962615966797 6
47             9.09007263183594 7.2514328956604 -0.09427810765305872 -0
48             .6472718661639695 -0.7564064844314669
49         block_change (1, 64, 1, 0, 50)
50         pos_change (1.8616574997357698, 65.39999999999995, 4.2761415
51             81759024)
52         pos_change (1.8615528336797424, 65.39999999999995, 4.2753018
53             31308588)
54         ...
55     ;
56 }
```