gComm: An environment for investigating generalization in Grounded Language Acquisition

Rishi Hazra
Indian Institute of Science, Bangalore
rishihazra@iisc.ac.in

Sonu Dixit
Indian Institute of Science, Bangalore
sonudixit@iisc.ac.in

Abstract

gComm[1] is a step towards developing a robust platform to foster research in grounded language acquisition in a more challenging and realistic setting. It comprises a 2-d grid environment with a set of agents (a stationary speaker and a mobile listener connected via a communication channel) exposed to a continuous array of tasks in a partially observable setting. The key to solving these tasks lies in agents developing linguistic abilities and utilizing them for efficiently exploring the environment. The speaker and listener have access to information provided in different modalities, i.e. the speaker's input is a natural language instruction that contains the target and task specifications and the listener's input is its grid-view. Each must rely on the other to complete the assigned task, however, the only way they can achieve the same, is to develop and use some form of communication. gComm provides several tools for studying different forms of communication and assessing their generalization.

1 Environment Description

Recently, datasets embodied in action and perception have been used to train models for various tasks (Vries et al., 2018; Mao et al., 2019). One such dataset is the grounded SCAN (gSCAN) dataset (Ruis et al., 2020) which is used for systematic generalization. We base our environment gComm on the gSCAN dataset which is a grounded version of SCAN benchmark (Lake and Baroni, 2018). While both these tasks focus on generalization with the meaning grounded in states of a grid-world, there are however, certain key differences between gComm and gSCAN: **(i)** Firstly, gSCAN focuses on rule-based generalization for navigation tasks, wherein, an agent learns to map a natural language instruction and its corresponding grid-view to a sequence of action primitives. Contrary to that, we present emergent communication as our main theme, using a pair of bots, a stationary speaker and a mobile listener, that process the language instruction and the grid-view respectively; **(ii)** Secondly, unlike the supervised framework adopted for learning gSCAN tasks, we use a more realistic RL-framework, wherein, the listener learns by exploring its environment and interacting with it. Our environment is conceptually similar to the BabyAI platform (Chevalier-Boisvert et al., 2019). However, contrary to BabyAI , which focuses on language *learning*, we intend to project gComm as a general purpose platform for investigating generalization from the perspective of grounded language *acquisition* through emergent communication.

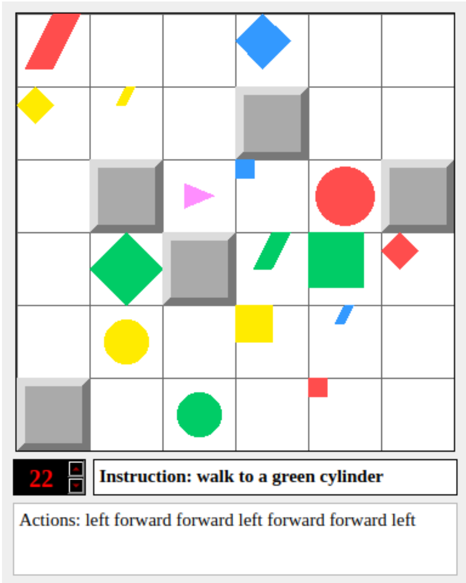

Figure 1: gComm Environment

Object Attributes: The gComm grid-world is populated with objects of different characteristics like shape, color, size and weight.

- **Shapes:** *circle, square, cylinder, diamond*
- **Colors:** *red, blue, yellow, green*
- **Sizes:** $1, 2, 3, 4$
- **Weights:** *light, heavy*

[1] codes & baselines: https//github.com/SonuDixit/gComm

The weight attribute can be fixed corresponding to the object size at the beginning of training. For instance, smaller sized objects are lighter and vice versa. Alternatively, the weight can be set as an independent attribute. In the latter option, the weight is randomly fixed at the start of each episode so that the listener cannot deduce the same from the grid information, and must rely on the speaker.

1.1 Reinforcement Learning framework

Setup: In each round, a task is assigned to a stationary Speaker-Bot, the details of which (task and target information) it must share with a mobile Listener-Bot by transmitting a set of messages $m_{i=1}^{n_m}$, via a communication channel. At each time-step t, the listener agent selects an action from its action space $\mathcal{A}$, with the help of the received messages $m_{i=1}^{n_m}$ and its local observation (grid-view) $o_t \in \mathcal{O}$. The environment state is updated using the transition function $\mathcal{T}$: $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. The environment provides a reward to the agent at each time-step using a reward function r: $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The goal of the agent is to find a policy $\boldsymbol{\pi}_\theta : (\mathcal{O}, m_{i=1}^{n_m}) \rightarrow \Delta(\mathcal{A})$ that chooses optimal actions so as to maximize the expected reward, $\mathcal{R} = \mathrm{E}_{\boldsymbol{\pi}}[\sum_t \gamma^t r^{(t)}]$ where r^t is the reward received by the agent at time-step t and $\gamma \in (0, 1]$ is the discount factor. At the beginning of training, their semantic repertoires are empty, and the speaker and listener must converge on a systematic usage of symbols to complete the assigned tasks thus, giving rise to an original linguistic system.

Observation Space: To encourage communication, gComm provides Speakera partially observable setting in which neither the speaker nor the listener has access to the complete state information. The speaker knows the task and target specifics through the natural language instruction whereas, the listener has access to the grid representation. However, the listener is unaware of either the target object or the task, and therefore must rely on the speaker to accomplish the given task. The observation space of the listener comprises (i) the grid representation; (ii) the messages transmitted by the speaker.

The natural language instruction is parsed to $\langle \text{VERB}, \{\text{ADJ}_i\}_{i=1}^{3}, \text{NOUN} \rangle$ using an ad hoc semantic parser[2]. It is then converted to the following 18-d vector representation before being fed to the speaker: {*1, 2, 3, 4, square, cylinder, circle, diamond, r, b, y, g, light, heavy, walk, push, pull, pickup*}. Each position represents a bit and is set or unset according to the target object attributes and the task. The breakdown of the vector representation is as follows: bits $[0-3]$: target size; $[4-7]$: target shape; $[8-11]$: target color; $[12-13]$: target weight; $[14-17]$: task specification.

The grid information can either be a image input of the whole grid or a predefined cell-wise vector representation of the grid. In the latter case, each grid cell in is specified by a 17-d vector representation given by: {1, 2, 3, 4, *square, cylinder, circle, diamond, r, b, y, g, agent, E, S, W, N*}. The breakdown is as follows: bits $[0-3]$: object size; $[4-7]$: object shape; $[8-11]$: object color; $[12]$: agent location (is set $= 1$ if agent is present in that particular cell, otherwise 0); $[13-16]$: agent direction. For an $obstacle$ or a $wall$, all the bits are set to 1.

Action Space: The action space comprises eight different actions that the listener agent can perform: {*left, right, forward, backward, push, pull, pickup, drop*}. In order to execute the 'push', 'pull', and 'pickup' actions, the agent must navigate to the same cell as that of the object. Upon executing a *pickup* action, the object disappears from the grid. Conversely, an object that has been picked up can reappear in the grid only if a 'drop' action is executed in the same episode. Also refer Section 1.2 for further details about task descriptions.

Rewards: gComm generates a 0-1 (sparse) reward, i.e., the listener gets a reward of $r = 1$ if it achieves the specified task, otherwise $r = 0$.

Communication: Recall that the listener has incomplete information of its state space and is thus unaware of the task and the target object. To address the information asymmetry, the speaker must learn to use the communication channel for sharing information. What makes it more challenging is the fact that the semantics of the transmitted information must be learned in a sparse reward setting, i.e. to solve the tasks, the speaker and the listener must converge upon a common protocol and use it systematically with minimal feedback.

1.2 Task Description

(i) Walk to a target object **(ii) Push** a target object in the forward direction. **(iii) Pull** a target object in the backward direction. **(iv) Pickup** a target object. **(v) Drop** the picked up object.

[2]VERB: task; ADJ: object attributes like color, size and weight; NOUN: object shape

Additionally, there are modifiers associated with verbs, for instance: *pull the red circle twice*. Here, *twice* is a numeral adverb and must be interpreted to mean two consecutive 'pull' actions. When an object is picked up, it disappears from the grid and appears only if a 'drop' action is executed in the subsequent time-steps. However, no two objects can overlap. It should be noted that while defining tasks, it is ensured that the target object is unique.

Target and Distractor objects: Cells in the grid-world are populated with objects divided into two classes: the *target* object and the *distractor* objects. The distractors either have the same color or the same shape (or both) as that of the target. Apart from these, some random objects distinct from the target can also be sampled using a parameter *other_objects_sample_percentage*. The listener and the objects may spawn at any random location on the grid.

Levels: In addition to the simple grid-world environment comprising target and distractor objects, the task difficulty can be increased by generating obstacles and mazes. The agent is expected to negotiate the complex environment in a sparse reward setting. The number of obstacles and the maze density can be adjusted.

Instruction generation: Natural language instructions are programmatically generated based on predefined lexical rules. At the beginning of training, the user specifies the kind of verb (transitive or intransitive), noun (object shape), and adjectives (object weight, size, color).

1.3 Communication

Communication Channel: The communication can be divided into two broad categories.

- **Discrete**: Discrete messages can either be binary (processed using Gumbel-Softmax (Jang et al., 2017)) or one-hot (processed using Categorical distribution)[3].
- **Continuous**: As opposed to discrete messages, continuous signals are real-valued. Theoretically speaking, each dimension in the message can carry 32-bits of information (32-bit floating point). These messages don't pose the same kind of information bottleneck as their discrete counterpart, however, they are not as interpretable.

[3]The use of discrete latent variables render the neural network non-differentiable. The Gumbel Softmax gives a differentiable sample from a discrete distribution by approximating the hard one-hot vector into a soft version. For one-hot vectors, we use Relaxed one-hot Categorical sampling. Since we want the communication to be discrete, we employ the *Straight-Through* trick for both binary and one-hot vectors.

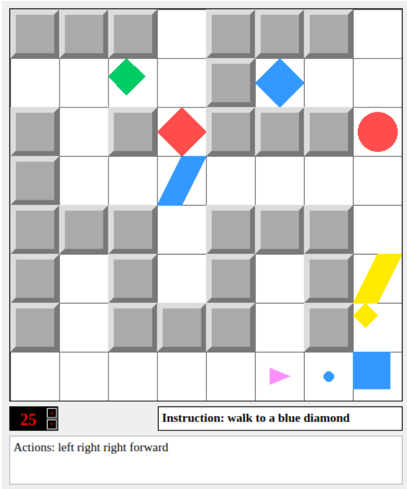

Figure 2: Maze-grid. The maze complexity and density are user-defined parameters. The agent is required to negotiate the obstacles while performing the given task.

The following baseline implementations are also readily available in the gComm environment. These baselines not only enable us to investigate the efficacy of the emergent communication protocols, but also provides quantitative insights into the learned communication abilities (Table 1).

- **Random Speaker**: In this baseline, the speaker transmits a set of random symbols to the listener which it must learn to ignore (and focus only on its local observation).
- **Fixed Speaker**: Herein, the speaker's transmissions are masked with a set of *ones*. Intuitively, this baseline provides an idea of whether communication is being used in the context of the given task (whether the speaker actually influences the listener or just appears to do so).
- **Perfect Speaker**: This baseline provides an illusion of a perfect speaker by directly transmitting the input concept encoding, hence, acting as an upper bound for comparing the learned protocols.
- **Oracle Listener**: For each cell, we zero-pad the grid encoding with an extra bit, and set it $(= 1)$ for the cell containing the target object. Thus, the listener has complete information about the target in context of the distractors.

Channel parameters: The communication channel is defined using the following parameters:

- Message Length: Length of the message vector d_m sets a limit on the vocabulary size, i.e. higher

Task	Baseline	Convergence Rewards
Walk	Simple Speaker	0.70
	Random Speaker	0.40
	Fixed Speaker	0.43
	Perfect Speaker	0.95
	Oracle Listener	0.99
PUSH & PULL	Simple Speaker	0.55
	Random Speaker	0.19
	Fixed Speaker	0.15
	Perfect Speaker	0.85
	Oracle Listener	0.90

Table 1: Comparison of baseline convergence rewards. communication: one-hot; grid-size: 4×4; episode length: 10; n_m: 3, d_m: 4; distractors: 4 (walk), 2 (push and pull)

the message length, larger is the vocabulary size. For instance, for discrete (binary) messages, the vocabulary size is given by $|\mathcal{V}| = 2^{d_m}$. Note, that a continuous message can transmit more information compared to a discrete message of the same length.

- Information Rate or the number of messages n_m transmitted per round of communication.

These constitute the channel capacity, $|\mathrm{C}| = \mathrm{c}_{d_m}^{n_m}$.

Setting: Communication can either be modelled in form of *cheap talk* or *costly signalling*. In the latter case, each message passing bears a small penalty to encourage more economic and efficient communication protocols. Alternatively, the communication can either be unidirectional (message passing from speaker to listener only) or bidirectional (an interactive setting wherein message passing happens in either direction). gComm uses an unidirectional cheap talk setting.

1.4 Metrics:

In order to induce meaningful communication protocols, the speaker must transmit useful information, correlated with its input (*positive signalling*). At the same time, the listener must utilize the received information to alter its behavior and hence, its actions (*positive listening*). In alignment with the works of (Lowe et al., 2019), we incorporate the following metrics in our environment to assess the evolved communication protocols.

- **Positive signalling**: Context independence (CI) is used as an indicator of positive signalling. It captures the statistical alignment between the input concepts and the messages transmitted by the speaker and is given by:

$$\forall c \in \mathcal{C} : m_c = \arg\max_m p_{cm}(c|m)$$

$$CI(p_{mc}, p_{cm}) = \frac{1}{|\mathcal{C}|} \sum_c p_{cm}(c|m_c) p_{mc}(m_c|c)$$

Both $p_{cm}(c|m)$ and $p_{mc}(m|c)$ are calculated using a translation model by saving (m, c) pairs and running it in both directions. Since each concept element c should be mapped to exactly one message m, CI will be high when the $p_{cm}(c|m)$ and $p_{mc}(m|c)$ are high.

- **Positive listening**: We use Causal Influence of Communication (CIC) of the speaker on the listener as a measure of positive listening. It is defined as the mutual information between the speaker's message and the listener's action $I(m, a_t)$. Higher the CIC, more is the speaker's influence on the listener's actions, thus, indicating that the listener is utilizing the messages.
- **Compositionality**: Compositionality is measured using the topographic similarity (topsim) metric (Brighton and Kirby, 2006). Given two pairwise distance measures, i.e. one in the concept (input) space $\Delta_{\mathcal{C}}^{ij}$ and another in the message space $\Delta_{\mathcal{M}}^{ij}$, topsim is defined as the correlation coefficient calculated between $\Delta_{\mathcal{C}}^{ij}$ and $\Delta_{\mathcal{M}}^{ij}$. Higher topsim indicates more compositionality.

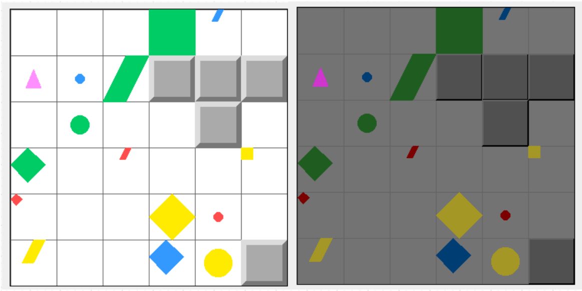

Figure 3: Lights Out

1.5 Additional features

We introduce a *lights out* feature in the gComm environment through which the grid (including all its objects) is subjected to varying illuminations (Figure 3). The feature can be activated randomly in each episode and presents a challenging situation for the agent where it is required to navigate the grid using its memory of the past observation. Note that this feature is useful only when used with an image input as the grid representation.

References

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artif. Life*.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR) 2017*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. 2019. On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 693–701. International Foundation for Autonomous Agents and Multiagent Systems.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations (ICLR) 2019*.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. 2020. A benchmark for systematic generalization in grounded language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 19861–19872. Curran Associates, Inc.

H. D. Vries, Kurt Shuster, Dhruv Batra, D. Parikh, J. Weston, and Douwe Kiela. 2018. Talk the walk: Navigating new york city through grounded dialogue. *ArXiv*.