Learning to Communicate and Solve Visual Blocks-World Tasks

Qi Zhang,[1] **Richard Lewis,**[2] **Satinder Singh,**[1] **Edmund Durfee**[1]
[1]Computer Science and Engineering, [2]Department of Psychology, University of Michigan
{qizhg,rickl,baveja,durfee}@umich.edu

Abstract

We study emergent communication between speaker and listener recurrent neural-network agents that are tasked to cooperatively construct a blocks-world target image sampled from a generative grammar of blocks configurations. The speaker receives the target image and learns to emit a sequence of discrete symbols from a fixed vocabulary. The listener learns to construct a blocks-world image by choosing block placement actions as a function of the speaker's full utterance and the image of the ongoing construction. Our contributions are (a) the introduction of a task domain for studying emergent communication that is both challenging and affords useful analyses of the emergent protocols; (b) an empirical comparison of the interpolation and extrapolation performance of training via supervised, (contextual) Bandit, and reinforcement learning; and (c) evidence for the emergence of interesting linguistic properties in the RL agent protocol that are distinct from the other two.

Introduction

We are interested in the challenging problem of learning effective communication protocols for collaborative multi-agent settings in which limited bandwidth communication channels can be exploited by agents for task performance, but where no protocols are provided to the agents in advance. This topic, sometimes called *language emergence*, has attracted interest in multiple fields over several decades (we briefly review some of this work below), including recent progress in the application of neural networks (NNs). We empirically study language emergence in a two-agent, *speaker-listener* NN-based architecture, where the speaker observes the task goal and takes *communication actions* that sequentially emit symbols, and the listener (never seeing the goal) receives the utterance and acts on the task environment. We investigate the degree to which the agents converge on communication protocols that have interesting linguistic structure, and that can successfully lead the listener to (at least partially) achieve the goal. We compare the speaker-listener architecture to a *baseline* architecture where a single-agent both sees the goal and acts on the environment.

Our work makes three novel contributions to language-emergence research. First, we introduce a collaborative blocks-world construction task involving communication of discrete symbol sequences that is challenging, systematically structured, and affords interesting analyses of the emergent communication. Second, we demonstrate that, when representing speaker and listener policies as recurrent neural networks, how they are trained affects how well their emergent protocols support generalization to unseen configuration sizes (defined by the number of blocks in the configurations). Specifically, we find that using Bandit for training supports greater flexibility in construction to improve performance over using supervised learning (SL), and that when using reinforcement learning (RL) more structured communication emerges to overcome the harder problem induced by delayed rewards. Third, we demonstrate the emergence of interesting linguistic properties that distinguish the RL-trained agent from the other two. Specifically, we provide evidence for the emergence of subsequences (N-grams) of symbols with a power-law frequency distribution similar to that found in natural human languages, and that are important in carrying meaning in way that is consistent with some degree of compositionality.

Related Work

Research on language emergence spans many fields. Linguistics and cognitive science are particularly interested in the origins of properties of natural human language such as hierarchical structure (Nowak, Plotkin, and Jansen 2000; Nowak, Komarova, and Niyogi 2001; Steels 2003; Kirby, Griffiths, and Smith 2014). The role of evolutionary processes and pressures in shaping human language remains a controversial topic (Berwick and Chomsky 2015), but a point of agreement among most cognitive scientists and linguists is that language to some degree is shaped by the structure of the human cognitive architecture (Bratman et al. 2010). A potential beneficial outcome of computational explorations such as our work is to provide tools for cognitive science to explore the implications of agent architecture for emergent linguistic capacities.

Recent work on learning communication among cooperative agents has proposed deep learning techniques for end-to-end learning of communication protocols. In Sukhbaatar et al. (2016), a population of homogeneous agents based on

NNs learned effective continuous communication through end-to-end backpropagation in several simulated multi-agent tasks. Foerster et al. (2016) proposed a framework with an end-to-end differentiable communication channel that allowed agents to communicate with one-bit messages for solving riddle games. In contrast to our paper, both of these papers did not evaluate the generalization ability of the learned communication protocols to unseen tasks. More recently, Mordatch and Abbeel (2017) grounded discrete communication learning in a multi-agent navigation task with goals given to the agents as disentangled features, and each agent was trained with the auxiliary task of predicting the goals of other agents. Like us, they used the Gumble-softmax technique for end-to-end communication learning, but unlike us, their agents are homogeneous in that they all have the same roles and they share all their parameters. They also emphasize and evaluate different kinds of generalization than in our experiments (specifically, they generalize to unseen numbers of agents and the presence of distractors).

In another line of work, the communication is learned in the context of two-player referential games wherein one agent has to identify the objects (usually images) referred to by the other agent through a learned language. For example, Lazaridou et al. (2016) allowed for communication with a single discrete symbol, while Havrylov and Titov (2017) allowed for variable-length discrete communication. Choi et al. (2018) and Lazaridou et al. (2018) instead took synthetic images as the referential objects, each containing an item of a particular color-shape specification. In these studies the emergent languages were able to generalize to unseen images. Our work explores generalization in settings that require the listener to perform a sequential task that places greater demands on the communication from the speaker to make rich task-relevant discriminations.

The Blocks-World Construction Task

Grammar for generating target configurations. A target configuration is generated by the probabilistic grammar in Table 1 by expanding the rule for a CP ("configuration phrase") with (x, y) initialized to $(0, 0)$ (lower left corner of a 10×10 grid). There are small, medium, and large blocks of sizes $1 \times 1, 2 \times 2$, and 3×3 squares. The grammar creates a variable number of stacks, from left to right, bounded by the width of the environment. Stacks have at most one large block, and a variable number of medium and small blocks, bounded by the environment's height. Larger blocks never go on smaller ones, and blocks cannot cantilever. Two towers of small blocks can appear over a large block. Figure 1 shows sample target configurations to provide a sense of the range of possible configurations. See Figure 2 for a tree sampled from the grammar, and its corresponding target image.

Observations. Observations of the target configuration and of the current work space are as raw (artificially generated) pixel images of the sort seen in Figure 1.

The non-terminal expanded based at (x, y)		Probability
$CP \rightarrow$	$LP(x, y)$	0.4
	$LP(x, y)\ \ CP(x + 5, y)$	0.6
$LP \rightarrow$	left $L(x, y)$	0.5
	right $L(x + 2, y)$	0.5
$MP \rightarrow$	left $M(x, y)$	0.25
	right $M(x + 1, y)$	0.25
	$S(x, y)\ \ S(x + 2, y)$	0.5
$SP \rightarrow$	left $S(x, y)$	0.5
	right $S(x + 1, y)$	0.5
$L \rightarrow$	LargeBlock(x, y)	0.01
	LargeBlock(x, y) $MP(x, y + 3)$	0.495
	$MP(x, y)$	0.495
$M \rightarrow$	MedBlock(x, y)	0.1
	MedBlock(x, y) $M(x, y + 2)$	0.45
	MedBlock(x, y) $SP(x, y + 2)$	0.45
$S \rightarrow$	SmallBlock(x, y)	0.1
	SmallBlock(x, y) $S(x, y + 1)$	0.9

Table 1: The probabilistic context-free grammar for target configuration generation. LargeBlock, MedBlock and SmallBlock are terminals; the rest are non-terminals.

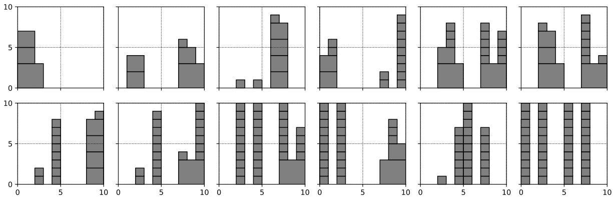


Figure 1: Sample target images generated from grammar in Table 1. Note the varying number of columns and heights.

Utterances and Construction Actions. In the two-agent setting, the speaker generates a fixed-length utterance by sequentially generating symbols from the elements of a fixed vocabulary V of primitive discrete symbols ("a", "b",...). **Note** that the symbols in V have no predefined meaning; their meaning gets implicitly grounded/defined during learning based on what actions the learner takes after hearing the utterances. A sample utterance is shown immediately above the grid in Figure 2b.

The learner's action set $\mathcal{A}$ contains (30) actions to add either a small, medium, or large block whose left side is aligned with a particular column x in the workspace, where $x \in [0, 10)$ is an integer. The y-coordinate of the added block is determined by simple gravity-like rules: in essence the block drops vertically until it hits another block below it or to a y of zero if there is no block below it. The listener also has a special (31^{st}) action *Terminate* $\in \mathcal{A}$ to end the episode. The episode also ends when the listener adds a block that is not in the target configuration. A sample action-sequence that places the blocks labeled 1–4, in that order, is shown at the top of Figure 2b.

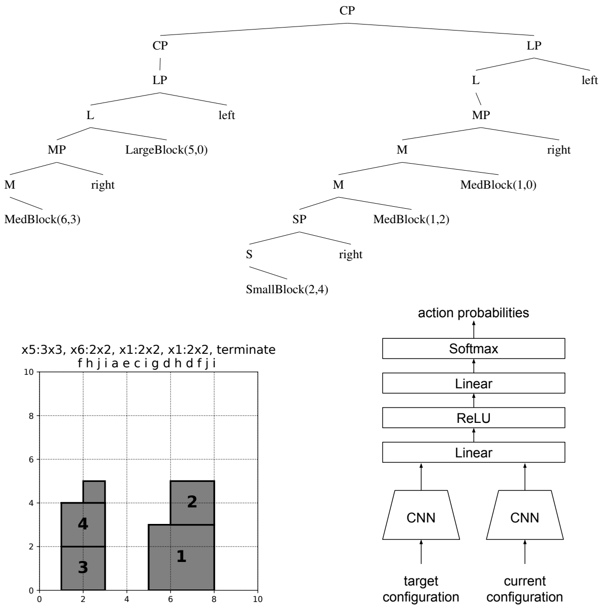


Figure 2: (a:top) Configuration tree generated from the grammar. (b:bottom-left) Target configuration rendered from the tree in (a). (c:bottom-right) The single agent architecture.

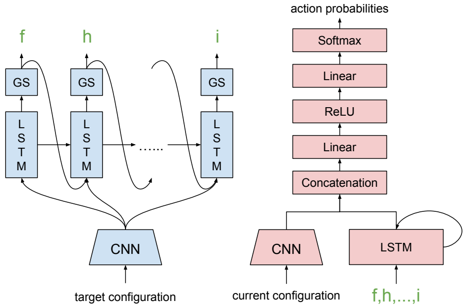


Figure 3: The speaker-listener architecture. The speaker's modules are in blue and the listener's modules are in red.

Agent Architectures and Learning

Single Agent Baseline Architecture. Shown in Figure 2c, the single agent gets both the target and current configuration images as input. The two images are processed by shared multiple convolutional layers (i.e., the 2 CNNs in the figure use the same weights). The concatenated learned feature encodings of the two configurations are fed into a ReLU layer and then into a softmax layer that produce a distribution over construction (and terminate) action probabilities. There are no utterances.

Speaker-Listener Architecture. Shown in Figure 3, the speaker gets the target image as input and outputs a fixed-length sequence of symbols, while the listener gets the full utterance of the speaker as input as well as the current workspace image, and outputs block placement / terminate actions. Note that during training the speaker and listener NNs are trained end to end as if they were a single NN, i.e., gradients flow from the listener to the speaker; it is during evaluation that they act as separate agents. Both the speaker and the listener use convolutional layers with shared weights to extract high-level features from the observed configurations presented as images, and separate LSTM networks for speaking and listening. The speaker's LSTM recursively takes the encoding of the target configuration and of the previous symbol to produce a distribution over symbols to emit at the next step. A fixed-length sequence of symbols is generated in this way using a Gumbel-softmax (Jang, Gu, and Poole 2016) trick (this allows gradients to pass through from listener to speaker that simply sampling from the softmax over symbols does not allow). The listener receives the entire discrete symbol sequence produced after Gumbel-softmax, and processes it through its LSTM to create an encoding of the speaker's full sentence. At each time step of acting on the workspace this sentence representation is concatenated with the CNN's encoding of the current configuration and fed through a ReLU layer into a final softmax layer to produce a distribution over construction (and terminate) actions.

Supervised, Bandit, and Reinforcement Learning

We evaluated three training algorithms on both the baseline and the speaker-listener agents. Note that there are exponentially many action sequences because of the partial order over actions that yield the same target image. When using supervised learning (SL), a *canonical* correct sequence of construction actions is used to do the training, where the action sequence builds stacks left to right, and each stack from the bottom up (corresponding to the blocks' order in a depth first traversal of the configuration's parse tree). When using (contextual) Bandit, the agents are given a reward of $+1$ for every construction action that is consistent with the target configuration and a -1 for an action that is inconsistent with the target configuration. When using RL the reward is the same as for Bandit but is accumulated and only made available at the end of the episode to the agents. In both Bandit and RL training an episode ends when the agent chooses an action that is inconsistent with the target image.

SL uses the following cross-entropy loss:

$$\mathcal{L}_{\text{SL}} = \mathbb{E}_{o^s \sim \mathcal{O}^s, o_0^l, a_0^*, o_1^l, a_1^*, \dots, a_{T-1}^*} \left[-\sum_t \log \pi_t^l(a_t^*) \right] \quad (1)$$

where o^s is a target configuration uniformly sampled from the set of possible target images $\mathcal{O}^s$, $\{a_t^*\}_{t=0}^{T-1}$ is the supervised action sequence for o^s, o_t^l is the configuration after taking the first t actions, and π_t^l is the distribution of the listener's actions conditioned on o_t^l and the speaker's utterance.

The Bandit loss function $\mathcal{L}_{\text{BL}} = \mathcal{L}_{\text{Bandit}} - \lambda \mathcal{L}_{\text{entropy}}$ is defined via

$$\mathcal{L}_{\text{Bandit}} = \mathbb{E}_{o^s \sim \mathcal{O}^s, o_0^l, a_0, o_1^l, a_1, \cdots} \left[-\sum_t r_t \log \pi_t^l(a_t) \right] \quad (2)$$

$$\mathcal{L}_{\text{entropy}}^l = \mathbb{E}_{o^s \sim \mathcal{O}^s, o_0^l, a_0, o_1^l, a_1, \cdots} \left[\sum_t H(\pi_t^l) \right] \quad (3)$$

where H is entropy, $\lambda \geq 0$ is the entropy regularization coefficient, and $r_t \in \{\pm 1\}$ is the reward of the Bandit. Here o_t^l is the configuration after taking the first t actions, and the listener's action a_t is sampled from π_t^l.

For RL training, we delay the reward signals until the very end of an episode. Formally, the RL loss function is $\mathcal{L}_{\text{RL}} = \mathcal{L}_{\text{REINFORCE}} - \lambda \mathcal{L}_{\text{entropy}}$, and $\mathcal{L}_{\text{REINFORCE}}$ is defined via

$$\mathcal{L}_{\text{REINFORCE}} = \tag{4}$$
$$\mathbb{E}_{o^s \sim \mathcal{O}^s, o_0^l, a_0, o_1^l, a_1, \cdots}\left[-\sum_t \log \pi_t^l(a_t)(R_t - b)\right]$$

where $R_t = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}^{\text{RL}}$ is the cumulative reward from time step t with discount factor $\gamma = 0.99$, and b is the baseline of REINFORCE which is set to be the average episodic reward of the previous epoch. Again o_t^l is the configuration after taking the first t actions, and the listener's action a_t is sampled from π_t^l.

Experiments: Agent Performance

Our experiments have two primary aims. The first aim is to demonstrate and understand the generalization abilities of the speaker-listener agents by examining their interpolation and extrapolation (these terms are defined formally below) performance. We contrast the performances obtained via Bandit, SL, and RL training, and use the single-agent as a useful baseline to provide a kind of upper-bound on expected performance of the speaker-listener agents. A second aim is to understand the emergent communication protocols and how their linguistic properties relate to agent performance. We describe the structure of the experiments and report the performance measures in this section, and report on the analyses of the communication protocols in the following section.

Experiment Structure

Data generation. We created data sets for our experiments by first sampling configurations from the probabilistic grammar and populating bins corresponding to configuration size (number of blocks) with unique configurations. More specifically, configurations were repeatedly sampled from the grammar up to a maximum of 10,000 unique configurations per bin and until the growth in unique configurations became very slow. Bins corresponding to configuration sizes of 8–21 blocks had the maximum of 10,000 unique configurations; bins in the 5–31 size range had 1,000 or more. The maximum configuration size is 40.

Training sets and interpolation-extrapolation testing sets. For each experiment we choose an *interpolation-extrapolation boundary* $B \in \{15, 25\}$. All configurations with number of blocks N satisfying $N \leq B$, except for configuration sizes of multiples of 5 are used for training (i.e., we trained only on configurations $N \leq B, N \bmod 5 \neq 0$). All other configurations are used for testing. Specifically, configurations with $N < B, N \bmod 5 = 0$ are used for interpolation testing, and $N > B$ for extrapolation.

Construction action and utterance symbol choice during testing. Unless otherwise noted explicitly, during testing, the speaker selects the next symbol with the maximum probability (breaking ties randomly) to utter and the listener selects the next action with the maximum probability (breaking any ties randomly) to take. The vocabulary size is set to $|V| = 10$ symbols, the utterance length is set to $L = 15$ symbols.

Results

Overall performance of interpolation and extrapolation. We use two measures to assess interpolation and extrapolation performance of all the agents. *Full completion* measures the empirical probability of completing the target configuration and terminating once it is constructed. *Partial completion* measures the proportion of the total number of blocks in the target configuration constructed before the first incorrect action (i.e., this degree is defined as n/N, where n is the number of blocks correctly added by the agent before the first, if any, mistake). The second measure is more forgiving, but it is informative of how much the agents have learned.

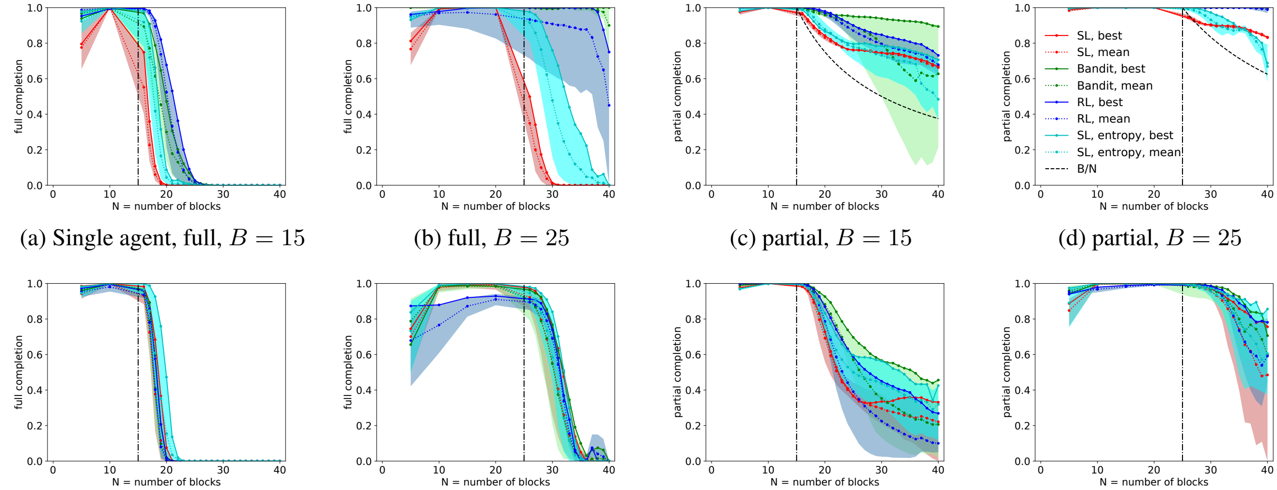
Figures 4a, 4b, 4e, and 4f show the full completion on testing configurations as a function of target image size N for interpolation-extrapolation boundary $B \in \{15, 25\}$. Figures 4c, 4d, 4g, and 4h show the partial completion. The graphs summarize the results of five independent runs for each algorithm at each boundary B; the shaded region is the performance range of the five runs, the top solid line is the best run, and the dotted line is the mean.

For the baseline agent (Figures 4a-4d), Bandit and RL training are comparable and both significantly outperform SL. Indeed, the extrapolation of both the best Bandit and the best RL trained baseline agent is excellent for the $B = 25$ setting, which itself is an interesting result (previous success on extrapolation in blocks world has required use of relational or other structured representations (Irodova and Sloan 2005)). For the speaker-listener agents (Figures 4e-4h), the 3 training algorithms are roughly comparable at full completion, but Bandit is better at partial completion (especially evident in Figure 4g). Below we highlight a few empirical conclusions.

Unsurprising advantage of Single Agent. For extrapolation, the single agent performs overall better than the speaker-listener, and this performance gap is especially large when the configuration size becomes large. This suggests that in this task, it is difficult for the agents to learn to generalize communication about very large numbers of blocks.

Unless explicitly stated otherwise all of the discussion that follows is for the speaker-listener agents.

Explaining the Bandit extrapolation advantage over SL. For extrapolation, the Bandit partial completion performance is better overall than SL. We tried a version of SL with entropy regularization with the same coefficient as in Bandit and RL, and while it improves performance a little bit, it is still worse than Bandit. All of the discussion about



(a) Single agent, full, $B = 15$ (b) full, $B = 25$ (c) partial, $B = 15$ (d) partial, $B = 25$

(e) Speaker-listener, full, $B = 15$ (f) full, $B = 25$ (g) partial, $B = 15$ (h) partial, $B = 25$

Figure 4: Full and partial completion on test configurations; X-axis is target configuration size. Points left of the vertical boundary line are interpolation tests; points right are extrapolation. *Top graphs*: Single agent architecture. *Bottom graphs*: Speaker-listener architecture. *Left graphs*: Full completion (probability). *Right graphs*: Partial completion (proportion). Shaded region is performance range of five runs, top solid line is best run, and dotted line is mean. Shared legend in upper right graph.

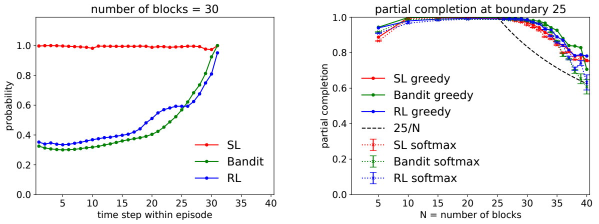


Figure 5: *Left:* Mean of maximum softmax probabilities for listener's actions during $N = 30$, $B = 25$ episodes. X-axis is time step within episode. *Right:* Partial completion from greedy sampling (solid) and softmax probabilities (dashed).

SL that follows in the rest of the paper is for training without entropy regularization. In the partial completion graphs (Figures 4c, 4d, 4g, 4h), we also plot the curve of B/N for $N > B$. If partial completion exceeds B/N for a $N > B$, the agents have learned to communicate about more blocks than they saw during training, instead of constructing only those blocks shared with the training configurations and ignoring the rest. The best performing Bandit run dominates B/N for $B \in \{15, 25\}$, while SL has runs dominating B/N only for $B = 25$. We conjecture that Bandit learning is better at extrapolation than SL because it can exploit the partial ordering over construction actions for a given configuration, while SL cannot. To confirm this conjecture we examined the softmax probabilities according to which the listener selects actions, and assessed performance when greedy action selection is replaced with sampling from the softmax. Figure 5 (left) shows the maximum probability of the listener's softmax-action over time in an episode, averaged over target configurations with $N = 30$ blocks. (The results correspond to the learning runs at boundary $B = 25$ reported in Figure 4h above.) For the SL listener, the maximum softmax probability is almost always near one, while for the Bandit listener, it begins below half and then increases over time in an episode, indicating that substantial probability mass is spread over multiple actions. Figure 5 (right) shows partial completion when the listener selects actions greedily and when it selects actions by sampling from the softmax. For the SL listener, there is little difference between the two selection strategies because the softmax is very sharp. For the Bandit listener, sampling from the softmax results in only modest reduction in partial completion, indicating that the Bandit agents have learned to communicate such that the listener is able to identify more than one correct action for a given time point of the episode. Of course, RL agents have the same flexibility as the Bandit agents in terms of the ability to learn multiple correct actions and unsurprisingly the RL curves in Figure 5 are similar to the Bandit curves. However, RL agents do have a harder problem because of the delayed reward, and this leads to their lower performance.

Evidence of learning the blocks-world domain structure. Because we have used a grammar to specify the structural regularities of the domain, it is possible to ask to what extent the agents have learned this structure. One way to do this is to ask whether the incorrect construction actions nevertheless are *grammatical* in the sense that they result in a configuration within the generative space of the grammar. Specifically, we analyzed the first incorrect action and found that in all runs for all agents, at least 80% of the incorrect actions were grammatical in this sense; in nearly all runs for block sizes of < 30, the proportion of ungrammatical errors was less than 5%. A random action baseline produces only

20% grammatical actions.

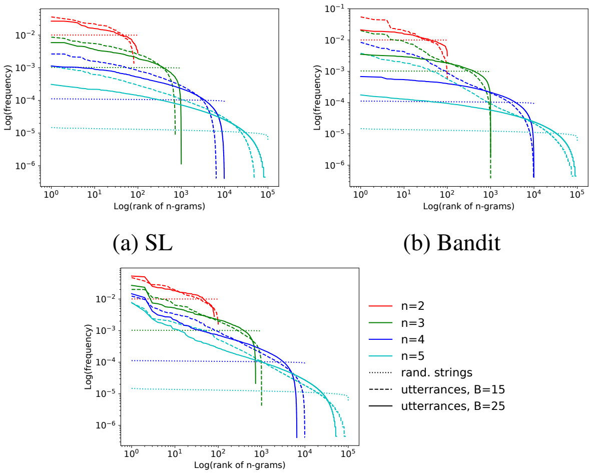
Analyses of the Communication Protocols

We now analyze the emergent communication protocols to gain insight into their linguistic properties and how these properties relate to performance. The analyses support four specific claims concerning interesting qualitative differences in the RL agents' protocol as compared to the other two.

N-gram distributions for the three protocols. The first claim is that the RL agents rely on the reuse of frequent subexpressions to a much greater degree than the other agents. A simple distributional analysis of N-grams in the utterance corpora from the three protocols reveals this difference. Figure 6 plots (log) frequency against (log) rank for all the of N-grams (up to 5-grams) produced by the RL, Bandit, and SL speakers. We plot the distributions in this manner to assess a possible correspondence with a Zipfian distribution, which is expressed as an approximately log-linear relationship between frequency and rank, a robust statistical regularity of linguistic forms in human languages (including words and even larger collocations) (Ellis 2002). The dashed lines show what is expected from a distribution of random strings. All agents are reusing subexpressions at much higher frequency than chance, and the approximate log-linear relationship holds up to very infrequent forms. But for the RL agents, the top-ranked N-grams are one to two orders of magnitude more frequent than the corresponding rank for the SL agents for both boundary 15 and 25 and for the Bandit agents in the boundary 25 case. In the boundary 15 case, the top-ranked N-gram for the RL agents are as frequent as, if not more than, the corresponding rank for the Bandit agents. Thus, RL utterances are much more likely to be composed of longer subexpressions that are reused in other utterances.

Action sequences induced by novel utterances composed of frequent N-grams. The second claim is that these frequent subexpressions and their compositions are more important for conveying meaning for the RL agents. We test this claim by examining the action sequences induced by giving the listeners novel utterances composed from a vocabulary of frequent N-grams. The listeners take utterances as input and take actions until the first action that results in an ungrammatical configuration; we record the number of grammatical (in the sense of producing a grammatical blocks configuration) actions. Note that this is not a partial completion measure because there is no target configuration, but longer mean grammatical action sequences for a set of utterances indicate that the utterances are able to convey a larger portion of the space of grammatical configurations.

The analysis starts with the K most frequent N-grams, for $n = 2, 3, 4$, and 5, to form a vocabulary of $4K$ "words"; we explore $K = 10, 20, 30, 40, 50$. We compare the mean grammatical action sequence length across three utterance types: utterances composed of random strings, utterances composed of a *single* N-gram followed by a random string of symbols, and utterances composed entirely of a concatenation of frequent N-grams. An increase in the grammatical action sequence length across these three types would provide evidence that N-grams and their composition are important conveyors of meaning.



(a) SL

(b) Bandit

(c) RL

Figure 6: (Log) frequency of N-gram subsequences against (log) frequency rank for N-grams of different sizes (dashed for B=15, solid for B=25), with the expected distribution of random strings (dotted).

Table 2: Mean number of grammatical actions induced by length 15 utterances composed of top 20 ranked N-grams, one N-gram with random string suffixes, and random strings. Cells are in the format of $B = 15$ / $B = 25$. All standard errors are less than 0.01.

	SL	Bandit	RL
N-gram full	8.06 / 6.04	10.11 / 7.24	7.60 / 7.14
One N-gram	7.28 / 6.70	6.26 / 6.58	3.80 / 4.27
Rand. strings	6.29 / 6.34	4.33 / 5.47	3.15 / 2.75

Table 2 summarizes the results for utterances composed of the top $K = 20$ N-grams ($K = 20$ produced the longest grammatical sequences for all three agents). There is a clear increase for the RL agents across the three utterance types, suggesting that the frequent subexpressions are indeed meaningful at multiple positions in the utterance string and when they are composed. (This is not evidence that the subexpressions have a meaning that is independent of position or context; showing this would require more detailed analysis but does not bear on the present claim.) The protocols of the SL and Bandit agents have this property in the boundary 15 case, but in the boundary 25 case in which more of the larger configurations are used for training, this property disappears for the SL, and the Bandit has it to a lesser degree.

The relationship between utterance similarity and configuration similarity. Our third claim is that the RL protocol bears one simple signature of compositionality. The finding that the compositions of N-grams are more likely to

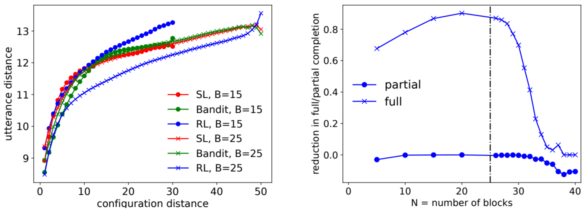


Figure 7: *Left:* Mean utterance similarity measured by edit distance as a function of configurations similarity measured by edit distance between the canonical action sequences of sampled pairs. All standard errors are less than 0.02. *Right:* Reduction in full and partial completion for the specific case of RL B=25 one-symbol prefix truncation.

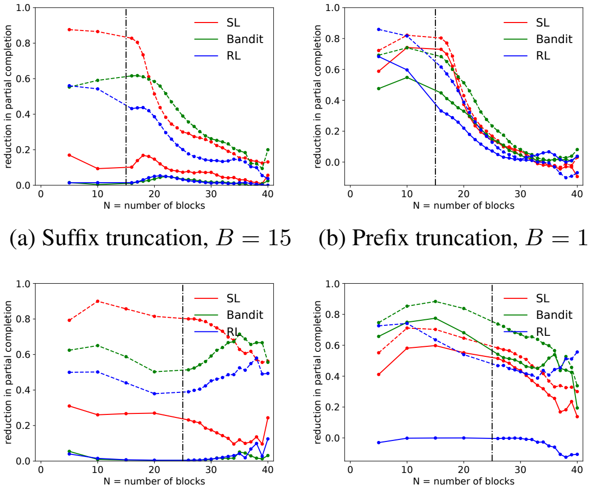
carry meaning than random strings or single N-grams for RL (and to a lesser degree, Bandit) agents suggests that the emergent protocol has some degree of compositional structure: the meaning of an expression is composed of the meanings of subparts of the expression. (Compositionality is thought to be a hallmark of human language, though even in human language, forms such as idioms violate compositionality.)

One implication of compositionality is that meaning similarity and utterance similarity will covary, and we test for this relationship as follows. We sample pairs of target configurations of 5–15 blocks for boundary 15 and 5–25 blocks for boundary 25, for which the full completion of the agents is near one. We then give these configurations as input to the speaker to generate corresponding pairs of utterances, and measure the mean edit distance between pairwise utterances as a function of the edit distance between the canonical action sequences of the corresponding pairwise configurations.

Figure 7(left) summarizes the results. For all agents, it is clear that utterance similarity and configuration similarity covary. From boundary 15 to 25, the agents are trained with more of the larger configurations, and the RL utterance pairs become more similar (that is, the edit distance is lower). The SL and Bandit utterance pairs do not have this property.

Prefix and suffix meanings: Robustness to truncation. Our final claim is that the RL protocol is more robust to prefix and suffix truncation, consistent with another signature of compositionality: parts of expressions are meaningful in isolation, and convey part of the meaning of the whole. More specifically, we probe here to what degree prefixes and suffixes of utterances are meaningful to the listener.

Figure 8(left) shows the *reduction* in partial completion relative to no truncation when the utterances are truncated at $l < L = 15$ and the listener only takes as input the first l symbols before taking any action. Similarly Figure 8(right) shows the reduction that happens when the utterance communicated to the listener drops the first symbol or first two symbols. The performance of all agents is more sensitive to prefix truncation; dropping just the first two symbols degrades performance considerably—but the RL agents are the most robust to prefix truncation, followed by Bandit. Both RL and Bandit agents are more robust to suffix truncation than SL, suggesting that these agents have learned to use prefixes to encode earlier parts of the correct action sequence.



(a) Suffix truncation, $B = 15$ (b) Prefix truncation, $B = 15$

(c) Suffix truncation, $B = 25$ (d) Prefix truncation, $B = 25$

Figure 8: Effect of truncated utterances. The curves are from the best SL/Bandit/RL, and show the *reduction* in partial completion relative to no-truncation (higher values correspond to greater reduction and hence worse performance). *Left:* For suffix truncation, the utterances are truncated at 14 (solid), or 10 (dashed) *Right:* For prefix truncation, we drop the first one (solid), or two (dashed) symbols.

Interestingly, for the RL agent in the boundary 25 case (see Figure 8d) partial completion when the first symbol is truncated is even higher than that with no truncation for some configuration sizes. However, as Figure 7(right) shows, there is significant reduction in full completion when the first symbol is truncated, verifying that information is spread across the full utterance.

We conclude by arguing why the differences between RL and other agents that we have claimed are suggestive of compositionality are unlikely to be due instead just to differences in how agents distribute information across the utterance. It is unclear how even a complex information distribution could explain how RL is more robust to both prefix and suffix truncation, where the N-gram composition analysis supports the plausible explanation that subexpressions carry meaning more systematically for RL protocols. A distribution that makes the RL utterance effectively smaller could explain some results, but is inconsistent with affix truncation analysis and with the edit distance results for dissimilar configurations. And the pattern in RL, but not SL, where the number of induced grammatical actions increases with the use of more frequent N-grams, are consistent with a compositionality explanation but much less so based on information distribution.

Conclusion

This work presented a new collaborative blocks construction domain for studying emergent communication that is challenging because it demands the speaking agent to express a rich set of task-relevant discriminations for the listening agent to succeed in the complex sequential decision-making task it faces. We demonstrated that it is possible to train speaker-agent recurrent neural nets with supervised, bandit, and reinforcement learning algorithms such that all three emergent communication protocols allow the agents to successfully communicate about blocks configurations of sizes unseen in the test set, including robust interpolation to sizes within the range of the test set, and modest extrapolation to larger sizes. We furthermore provided evidence that the Bandit training generalizes more robustly than SL training by exploiting the partial order over correct action sequences versus the canonical order forced upon SL.

Finally, we took steps to understand the nature of the emergent communication protocols, and how the differences among the three protocols might manifest in performance. Specifically, we provided evidence for the emergence of a Zipfian distribution of N-grams of symbols in all three protocols, but with a qualitatively greater use of frequent N-grams in the RL protocols. We furthermore showed that for the RL protocols the frequent N-grams are more important in conveying meaning, that similar meanings (configurations) yield more similar utterances, and that parts of complete utterances (prefixes and suffixes) more robustly carry parts of the whole meaning. These differences are possible signatures of a greater degree of compositionality in the RL protocols.

These qualitative protocol differences and improved robustness to truncation for the RL agents arise despite the fact that the RL agents performed somewhat worse overall on the interpolation and extrapolation measures than the SL and Bandit agents. The overall performance difference is not surprising given the significantly more challenging nature of the delayed reward RL training. But we can only conjecture that the more robust communication learned by the speaker trained via RL also results from having to overcome the same more challenging delayed RL feedback. Evaluating this conjecture, and applying unsupervised learning and other induction methods used in human language analysis, remains future work.

Acknowledgments We thank the anonymous reviewers for great suggestions. This work was supported in part by the Air Force Office of Scientific Research under grant FA9550-15-1-0039. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

Berwick, R. C., and Chomsky, N. 2015. *Why Only Us: Language and Evolution.* MIT press.

Bratman, J.; Shvartsman, M.; Lewis, R. L.; and Singh, S. 2010. A new approach to exploring language emergence as boundedly optimal control in the face of environmental and cognitive constraints. In *Proceedings of the 10th International Conference on Cognitive Modeling*, 7–12.

Choi, E.; Lazaridou, A.; and de Freitas, N. 2018. Compositional obverter communication learning from raw visual input. *arXiv preprint arXiv:1804.02341.*

Ellis, N. C. 2002. Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in Second Language Acquisition* 24(2):143–188.

Foerster, J.; Assael, I. A.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Havrylov, S., and Titov, I. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems*, 2146–2156.

Irodova, M., and Sloan, R. H. 2005. Reinforcement learning and function approximation. In *FLAIRS Conference*, 455–460.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144.*

Kirby, S.; Griffiths, T.; and Smith, K. 2014. Iterated learning and the evolution of language. *Current Opinion in Neurobiology* 28:108–114.

Lazaridou, A.; Hermann, K. M.; Tuyls, K.; and Clark, S. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984.*

Lazaridou, A.; Peysakhovich, A.; and Baroni, M. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182.*

Mordatch, I., and Abbeel, P. 2017. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908.*

Nowak, M. A.; Komarova, N. L.; and Niyogi, P. 2001. Evolution of universal grammar. *Science* 291(5501):114–118.

Nowak, M. A.; Plotkin, J. B.; and Jansen, V. A. 2000. The evolution of syntactic communication. *Nature* 404(6777):495.

Steels, L. 2003. Evolving grounded communication for robots. *Trends in Cognitive Sciences* 7(7):308–312.

Sukhbaatar, S.; Fergus, R.; et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2244–2252.