Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog

Satwik Kottur[1] and **José M.F. Moura**[1] and **Stefan Lee**[2,3] and **Dhruv Batra**[3,4]
[1]Carnegie Mellon University, [2]Virginia Tech, [3]Georgia Tech, [4]Facebook AI Research

Abstract

A number of recent works have proposed techniques for end-to-end learning of communication protocols among cooperative multi-agent populations, and have simultaneously found the *emergence of grounded human-interpretable language* in the protocols developed by the agents, learned without any human supervision!

In this paper, using a *Task & Talk* reference game between two agents as a testbed, we present a sequence of 'negative' results culminating in a 'positive' one – showing that while most agent-invented languages are effective (*i.e.* achieve near-perfect task rewards), they are decidedly *not* interpretable or compositional. In essence, we find that natural language does not emerge 'naturally', despite the semblance of ease of natural-language-emergence that one may gather from recent literature. We discuss how it is possible to coax the invented languages to become more and more human-like and compositional by increasing restrictions on how two agents may communicate.

1 Introduction

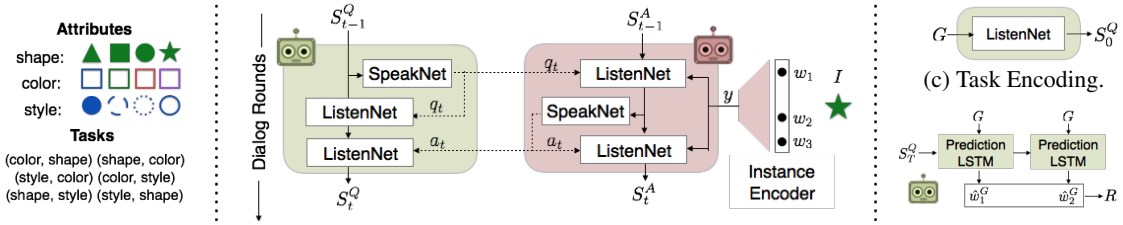
One fundamental goal of artificial intelligence (AI) is the development of goal-driven dialog agents – specifically, agents that can perceive their environment (through vision, audition, or other sensors), and communicate with humans or other agents in natural language towards a goal.

While historically such agents have been based on *slot filling* (Lemon et al., 2006), the dominant paradigm today is neural dialog models (Bordes and Weston, 2016; Weston, 2016; Serban et al., 2016a,b) trained on large quantities of data. Perhaps somewhat counterintuitively, this current paradigm treats dialog as a static supervised learning problem, rather than as the interactive agent learning problem that it naturally is. Specifically, a typical pipeline is to collect a large dataset of human-human dialog (Lowe et al., 2015; Das et al., 2017a; de Vries et al., 2017; Mostafazadeh et al., 2017), inject a machine in the middle of a dialog from the dataset, and supervise it to mimic the human response. While this teaches the agent correlations between symbols, it does not convey the functional meaning of language, grounding (mapping physical concepts to words), compositionality (combining knowledge of simpler concepts to describe richer concepts), or aspects of planning (why are we having this conversation?).

An alternative paradigm that has a long history (Winograd, 1971; Kirby et al., 2014) and is witnessing a recent resurgence (Wang et al., 2016; Foerster et al., 2016; Sukhbaatar et al., 2016; Jorge et al., 2016; Lazaridou et al., 2017; Havrylov and Titov, 2017; Mordatch and Abbeel, 2017; Das et al., 2017b) – is situated language learning. A number of recent works have proposed reinforcement learning techniques for learning the communication protocols of agents situated in virtual environments in a completely end-to-end manner – from perceptual input (*e.g.* pixels) to communication (discrete symbols without any prespecified meanings) to action (*e.g.* signaling in reference games or navigating in an environment) – and have simultaneously found the *emergence of grounded human-interpretable (often compositional) language* among agents, without any human supervision or pretraining, simply to succeed at the task.

In this short paper, we study the following question – what are the conditions that lead to the emergence of human-interpretable or compositional grounded language? Our key finding is that

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2962–2967
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics



(a) World. (b) Q-BOT (left) and A-BOT (right) policy networks. (d) Prediction.

Figure 1: *Task & Talk*: The testbed for our study is cooperative 2-player game, *Task & Talk*, grounded in a synthetic world of objects with 4 *shapes* × 4 *colors* × 4 *styles*. The two agents, Q-BOT and A-BOT, are modeled as neural networks, and their policies are learned via REINFORCE. We find that the languages invented by the two agents are typically *not* 'natural'.

natural language *does not emerge 'naturally'* in multi-agent dialog, despite independently reported successful demonstrations in recent literature.

Specifically, in a sequence of 'negative' results culminating in a 'positive' one, we find that while agents always successfully invent communication protocols and languages to achieve their goals with near-perfect accuracies, the invented languages are decidedly *not* compositional, interpretable, or 'natural'; and that it is possible to coax the invented languages to become more and more human-like and compositional by increasing restrictions on how two agents may communicate.

Related work and novelty. The starting point for our investigation is the recent work of Das et al. (2017b), who proposed a cooperative reference game between two agents, where communication is necessary to accomplish the goal due to an information asymmetry. Our key contribution over Das et al. (2017b) is an exhaustive study of the conditions that must be present before compositional grounded language emerges, and subtle but important differences in execution – tabular Q-Learning (which does not scale) *vs*. REINFORCE (which does), and generalization to novel environments (not studied in prior work). In the spirit of Abbeel et al. (2017), we hope our findings shed more light into the interpretability of languages invented in cooperative multi-agent settings, place recent work in appropriate context, and inform fruitful directions for future work.

2 The *Task & Talk* Game

Our testbed is a reference game (*Task & Talk*) between two agents, Q-BOT and A-BOT. The game is grounded in a synthetic world of objects comprised of three attributes – *color*, *style*, and *shape* – each with four possible values for a total of $4 \times 4 \times 4 = 64$ objects. Fig. 1a shows some example instances from this set.

Task & Talk plays out over multiple rounds of dialog. At the start, A-BOT is given an instance (*e.g. (green, dotted, square)*) unseen by Q-BOT, and Q-BOT is assigned a task G (unknown to A-BOT) consisting of two attributes for Q-BOT to discover from A-BOT (*e.g. (color, style)*). For two rounds, Q-BOT and A-BOT exchange utterances from finite vocabularies V_Q and V_A, with Q-BOT speaking first. The game culminates in Q-BOT guessing a pair of attribute values (*e.g. (green, dotted)*) and both agents are rewarded identically based on the accuracy of this prediction.

Note that the Task & Talk game setting involves an informational asymmetry between the agents – A-BOT sees the object while Q-BOT does not; similarly Q-BOT knows the task while A-BOT does not. Thus, a two-way communication is necessary for success. Without this asymmetry, A-BOT could simply convey the target attributes from the task without Q-BOT having to speak. Such a setting has been widely studies in economics and game theory as the classic Lewis Signaling (LS) game (Lewis, 2008). By necessitating *dialog* between agents, we are able ground both V_A and V_Q in our final setting (Sec. 4.3).

3 Modeling Q-BOT and A-BOT

We formalize Q-BOT and A-BOT as agents operating in a partially observable world and optimize their policies using deep reinforcement learning.

States and Actions. Each agent observes its own input (task G for Q-BOT and object instance I for A-BOT) and the output of the other agent as a stochastic environment. At the beginning of round t, Q-BOT observes state $s_Q^t = [G, q_1, a_1, \ldots, q_{t-1}, a_{t-1}]$ and acts by uttering some token q_t from its vocabulary V_Q. Similarly, A-BOT observes the history and this new utterance as state $s_A^t = [I, q_1, a_1, \ldots, q_{t-1}, a_{t-1}, q_t]$ and emits a response a_t from V_A. At the last round, Q-BOT takes a final action by predicting a pair of attribute values $\hat{w}^G = (\hat{w}_1^G, \hat{w}_2^G)$ to solve the task.

Cooperative Reward. Both Q-BOT and A-BOT are rewarded identically based on the accuracy of

Q-BOT's prediction $\hat{w}^G$, receiving a positive reward of $R{=}1$ if the prediction matches ground truth and a negative reward of $R{=}{-}10$ otherwise. We arrive at these values empirically.

Policy Networks. We model Q-BOT and A-BOT as operating under stochastic policies $\pi_Q(q_t|s_t^Q;\theta_Q)$ and $\pi_A(a_t|s_t^A;\theta_A)$ respectively, which we instantiate as LSTM-based models. We use lower case characters (*e.g.* s_t^Q) to denote the strings (*e.g.* Q-BOT's token at round t), and upper case S_t^Q to denote the corresponding vector as encoded by the model.

As shown in Fig. 1, Q-BOT is modeled with three modules – speaking, listening, and prediction. The task G is received as a 6-dimensional one-hot encoding over the space of possible tasks and embedded via the listener LSTM. At each round t, the *speaker network* models the probability of output utterances $q_t \in V_Q$ based on the state S_{t-1}^Q. This is modeled as a fully-connected layer followed by a softmax that transforms S_{t-1}^Q to a distribution over V_Q. After receiving the reply a_t from A-BOT, the *listener LSTM* updates the state by processing both tokens of the dialog exchange. In the final round, the *prediction LSTM* is unrolled twice to produce Q-BOT's prediction based on the final state S_T^Q and the task G. As before, task G is fed in one-hot to the prediction LSTM for two time steps, resulting in a pair of outputs used as the prediction $\hat{w}_G$.

Analogously, A-BOT is modeled as a combination of a *speaker network*, a *listener LSTM*, and an *instance encoder*. Like in Q-BOT, the speaker network models the probability of utterances $a_t \in V_A$ given the state S_t^A and the listener LSTM updates the state S_t^A based on dialog exchanges. The instance encoder embeds each one-hot attribute vector via a linear layer and concatenates all three encodings to obtain a unified instance representation.

Learning Policies with REINFORCE. We train these models using the popular REINFORCE (Williams, 1992) policy gradient algorithm. Note that while the game is fully-cooperative, we do not assume full observability of one agent by another, opting instead to treat one agent as part of the unknown stochastic environment when updating the other. During training, we sample 1000 two round dialog episodes per batch and update policy parameters with Adam (Kingma and Ba, 2015) based on these approximate gradients. Our code is publicly available[1].

[1] github.com/batra-mlp-lab/lang-emerge

4 The Road to Compositionality

This section details our key observation – that while the agents always successfully invent a language to solve the game with near-perfect accuracies, the invented languages are decidedly *not* compositional, interpretable, or 'natural' (*e.g.* A-BOT ignoring Q-BOT's utterances and simply encoding every object with a unique symbol if the vocabulary is sufficiently large). In our setting, the language being compositional simply amounts to the ability of the agents to communicate the compositional atoms of a task (*e.g.* *shape* or *color*) and an instance (*e.g.* *square* or *blue*) independently.

Through this section, we present a series of settings that get progressively more restrictive to coax the agents towards adopting a compositional language, providing analysis of the learned languages developed along the way. Table 1 summarizes results for all settings. In all experiments, optimal policies (achieving near-perfect rewards) were found. For each setting, we provide qualitative analysis of the learned languages and report their ability to generalize to unseen instances. We use 80% of the object-instances for training and the remaining 20% to evaluate these learned policies. Further, greedy argmax policies are used at evaluation time.

4.1 Overcomplete Vocabularies

We begin with the simplest setting where both A-BOT and Q-BOT are given arbitrarily large vocabularies. We find that when $|V_A|$ is greater than the number of instances (64), the learned policy simply has A-BOT ignore what Q-BOT asks and instead convey the instance using a single symbol, *e.g.* token 1≡*(red, square, filled)*. Notice that this means no 'dialog' is necessary and amounts to each agent having a codebook that maps symbols to object instances.

Perhaps as expected, the generalization of this language to unseen instances is quite poor (success rate: 25.6%). The adopted strategy of mapping instances to token pairs fails for test instances containing novel combinations of attributes for which the agents lack an agreed-upon code from training.

It seems clear that like in human communication (Nowak et al., 2000), a limited vocabulary that cannot possibly encode the richness of the world seems to be necessary for non-trivial dialog to emerge. We explore such a setting next.

Setting	Vocab.		Memory		Gen.(%)	
	V_Q	V_A	A	Q	Both	One
Overcomplete (§4.1)	64	64	✓	✓	25.6	79.5
Attr-Value (§4.2)	3	12	✓	✓	38.5	88.4
NoMem-Min (§4.3)	3	4	✗	✓	**74.4**	**94.9**

Table 1: Overview of settings we explore to analyze the language learnt by two agents in a cooperative game, Task & Talk. Last two columns measure generalization in terms of prediction accuracy of **both** or at least **one** of the attribute pair, on a held-out test set containing unseen instances.

4.2 Attribute & Value Vocabulary

Since our world has 3 attributes (*shape*/*color*/*style*) and $4+4+4=12$ possible settings of their states, one may believe that the intuitive choice of $|V_Q|=3$ and $|V_A|=12$ will be enough to circumvent the 'cheating' enumeration strategy from the previous experiment. Surprisingly, we find that the new language learned in this setting is not only decidedly non-compositional but also very difficult to interpret! We present two salient observations.

We observe that Q-BOT uses only the first round to convey the task to A-BOT by encoding tasks in an order-independent fashion *e.g.* the *(style,color)* and *(color,style)* tasks are both expressed as the utterance *Z* in the first round. Consequentially, multiple rounds of dialog are rended unnecssary and the second round is inconsistent across instances even for the same task.

Given the task from Q-BOT in the first round, A-BOT only needs to identify one of the 4×4=16 attribute pairs for a given task. Rather than ground its symbols into individual states, A-BOT follows a 'set partitioning' strategy, *i.e.* A-BOT identifies a pair of attributes with a unique combinations of round 1 and 2 utterances (*i.e.* the round 2 utterance has no meaning independent from round 1). Thus, symbols are reused across tasks to describe different attributes (*i.e.* symbols do not have individual consistent groundings). This 'set partitioning' strategy is consistent with known results from game theory on Nash equilibria in 'cheap talk' games (Crawford and Sobel, 1982).

This strategy has improved generalization to unseen instances because it is able to communicate the task; however, it fails on unseen attribute value combinations because it is not compositional.

4.3 Memoryless A-BOT, Minimal Vocabulary

The key problem with the previous setting is that A-BOT's utterances *mean different things* based on the round of dialog ($a_1=1$ is different from $a_2=1$). Essentially, the communication protocol is overparameterized and we must limit it further. First, we limit A-BOT's vocabulary to $|V_A|$=4 to reduce the number of 'synonyms' the agents learn. Second, we *remove* A-BOT*'s memory* by reseting the state vector S^A at each time step, which eliminates its ability to enumerate all attribute pairs.

These restrictions result in a learned language that grounds *individual symbols* into attributes and their states. For example, Q-BOT learns that $Y \rightarrow$ shape, $X \rightarrow$ color, and $Z \rightarrow$ style. Q-BOT does not however learn to always utter these symbols in the same order as the task, *e.g.* asking for *shape* first for both $(color, shape)$ and $(shape, color)$. Notice that this is perfectly valid as Q-BOT can later re-arrange the attributes in the task desired order. Similarly, A-BOT learns mappings to attribute values for each attribute query that remain consistent regardless of round (*i.e.* when asked for $color$, 1 always means $blue$).

This is similar to learned languages reported in recent works and is most closely related to Das et al. (2017b), who solve this problem by taking away Q-BOT's state rather than A-BOT's memory. Their approach can be interpreted as Q-BOT 'forgetting' the task after interacting with A-BOT. However, this behavior of Q-BOT to remember the task only during dialog but not while predicting is somewhat unnatural compared to our setting.

Tab. 2 enumerates the learnt groundings for both the agents. Given this mapping, we can predict a plausible dialog between the agents for any unseen instance and task combination. Notice that this is possible only due to the compositionality in the emergent language between the two agents. For example, consider solving *(shape, color)* for an instance *(red, square, filled)* from Fig. 2(b). Q-BOT queries *Y (shape)* and *X (color)* across two rounds, and receives *2 (square)* and *4 (red)* as answers.

Intuitively, this consistently grounded and compositional language has the greatest ability to generalize among the settings we have explored, correctly answering 74.4% of the held out instances. We note that errors in this setting seem to largely be due to A-BOT giving an incorrect answers despite Q-BOT asking the correct questions to accomplish the task. A plausible reason could be the model approximation error stemming from the instance encoder as test instances are unseen and have novel attribute combinations.

Fig. 2(b) shows the dialog for the instance *(red, square, filled)* and task *(shape, color)*. Q-BOT queries *Y (shape)* and *(color)* across two rounds,

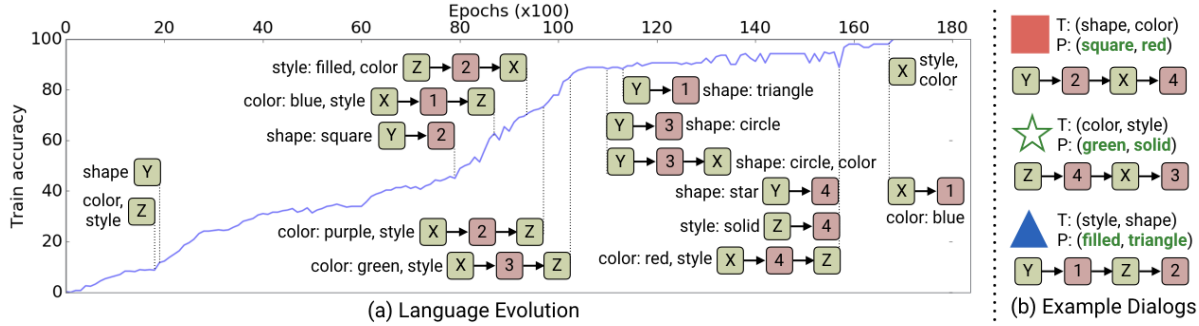


Figure 2: (a) Evolution of Language: timeline shows groundings learned by the agents during training, overlaid on the accuracy. Note that Q-BOT learns encodings for all tasks early (around epoch 20) except *(style, color)*. Improvement in accuracy is strongly correlated with groundings learnt. (b) Example dialogs for memoryless A-BOT, minimal vocabulary setting (§4.3.

	Attributes		
V_A	*color* X	*shape* Y	*style* Z
1	*blue*	*triangle*	*dotted*
2	*purple*	*square*	*filled*
3	*green*	*circle*	*dashed*
4	*red*	*start*	*solid*

(a) A-BOT

Task	q_1, q_2
(color, shape)	*Y, X*
(shape, color)	
(shape, style)	*Y, Z*
(style, shape)	
(color, style)	*Z, X*
(style, color)	*X, Z*

(b) Q-BOT

Table 2: Emergence of compositional grounding for language learnt by the agents. **A-BOT** (Tab. 2a) learns consistent mapping across rounds, depending on the query attribute. Token grounding for **Q-BOT** (Tab. 2b) depends on the task at hand. Though compositional, Q-BOT does not necessarily query attribute in the order of task, but instead re-arranges accordingly at prediction time as it contains memory.

and receives *2 (square)* and *4 (red)* as answers.

4.4 Evolution of Language Timeline

To gain further insight into the languages learned, we create a *language evolution* plot in Fig. 2. Specifically, at regular intervals during policy learning, we construct 'dialog trees'. A dialog tree enumerates all plausible dialogs between the two agents (q_1, a_1, q_2, a_2), as a tree. The root node is q_1, and at any node, we go deeper by choosing a branch based on the next utterance in the dialog. Since Task & Talk runs for two rounds, our dialog trees are 4 layers deep with $|V_A|^2|V_Q|^2$ leaves. Notice that a single input instance could potentially result in different dialogs depending on the task. Hence, we consider *(instance, task)* pairs and assign each to a leaf by traversing the tree according to the resulting dialog. At some point in the learning, the nodes become and stay 'pure' (the common trend among all *(instance, task)* at the node stays constant till the end of training), at which point we can say that the agents have learned this dialog subsequence.

Construction. After constructing dialog trees at regular intervals, we identify 'concepts' at each node/leaf using the dialog tree of the completely trained model, which achieves a perfect accuracy on train set. A concept is simply the common trend among all the *(instance, task)* tuples either assigned to a leaf or contained within the subtree with a node as root. Next, given a resultant concept for each of the node/leaf, we backtrack in time and check for the first occurrence when only tuples which satisfy the corresponding concept are assigned to that particular node/leaf. In other words, we compute the earliest time when a node/leaf is 'pure' with respect to its final learned concept. Finally, we plot these leaves/nodes and the associated concept with their backtracked time to get Fig. 2.

Observations. We highlight key observations below: (a) The agents ground most of the tasks initially at around epoch 20. Specifically, Q-BOT assigns *Y* to both *(shape, style), (style, shape), (shape,color)* and *(color, shape)*, while *(color, style)* is mapped to *Z*. Hence, Q-BOT learns its first token very early into the training procedure. (b) The only other task *(style, color)* is grounded towards the end (around epoch 170) using *X*, leading to an immediate convergence. (c) We see a strong correlation between improvement in performance and when agents learn a language grounding. In particular, there is an improvement from 40% to 80% within a span of 25 epochs where most of the grounding is achieved, as seen from Fig. 2.

5 Conclusion

In conclusion, we presented a sequence of 'negative' results culminating in a 'positive' one – showing that while most invented languages are effective (*i.e.* achieve near-perfect rewards), they are decidedly *not* interpretable or compositional. Our goal is simply to improve understanding and interpretability of invented languages in multi-agent dialog, place recent work in context, and inform fruitful directions for future work.

References

Pieter Abbeel, Igor Mordatch, Ryan Lowe, Jon Gauthier, and Jack Clark. 2017. Learning to Communicate. https://blog.openai.com/learning-to-communicate/.

Antoine Bordes and Jason Weston. 2016. Learning End-to-End Goal-Oriented Dialog. *arXiv preprint arXiv:1605.07683* .

Vincent Crawford and Joel Sobel. 1982. Strategic information transmission. *Econometrica* 50(6):1431–51.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017a. Visual Dialog. In *CVPR*.

Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. 2017b. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. *arXiv preprint arXiv:1703.06585* .

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*.

Jakob Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *NIPS*.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *ICLR Workshop*.

Emilio Jorge, Mikael Kågebäck, and Emil Gustavsson. 2016. Learning to play guess who? and inventing a grounded language as a consequence. In *NIPS workshop on deep reinforcement learning*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Simon Kirby, Tom Griffiths, and Kenny Smith. 2014. Iterated learning and the evolution of language. *Current opinion in neurobiology* 28:108–114.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*.

Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle. 2006. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system. In *EACL*.

David Lewis. 2008. *Convention: A philosophical study*. John Wiley & Sons.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *SIGDIAL*.

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

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P. Spithourakis, and Lucy Vanderwende. 2017. Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation. *arXiv preprint arXiv:1701.08251* .

Martin A. Nowak, Joshua B. Plotkin, and Vincent A. A. Jansen. 2000. The evolution of syntactic communication. *Nature* 404(6777):495–498.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016a. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016b. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. *arXiv preprint arXiv:1605.06069* .

Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *NIPS*. pages 2244–2252.

Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. *Association for Computational Linguistics (ACL)* .

Jason Weston. 2016. Dialog-based language learning. *arXiv preprint arXiv:1604.06045* .

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, DTIC Document.