Towards Graph Representation Learning in Emergent Communication

Agnieszka Słowik[*][1] **Abhinav Gupta**[*][2]
William L. Hamilton[23] **Mateja Jamnik**[1] **Sean B. Holden**[1]
[1] University of Cambridge [2] MILA [3] McGill University
agnieszka.slowik@cl.cam.ac.uk, abhinavg@nyu.edu

arXiv:2001.09063v2 [cs.LG] 4 Feb 2020

Abstract

Recent findings in neuroscience suggest that the human brain represents information in a geometric structure (for instance, through conceptual spaces). In order to communicate, we flatten the complex representation of entities and their attributes into a single word or a sentence. In this paper we use graph convolutional networks to support the evolution of language and cooperation in multi-agent systems. Motivated by an image-based referential game, we propose a graph referential game with varying degrees of complexity, and we provide strong baseline models that exhibit desirable properties in terms of language emergence and cooperation. We show that the emerged communication protocol is robust, that the agents uncover the true factors of variation in the game, and that they learn to generalize beyond the samples encountered during training.

Introduction

The ability to represent complex concepts and the relationships between them in the manner of a mental graph was found to be one of the key factors behind knowledge generalization and prolonged learning (Bellmund et al. 2018). Through communication, humans flatten the non-Euclidean representation of ideas into a sequence of words. Advances in graph representation learning (Kipf and Welling 2017) and sequence decoding (Sutskever, Vinyals, and Le 2014; Cho et al. 2014) provide the means for simulating this *graph linearization* process in the multi-agent setting.

One of the approaches to learning to communicate in a multi-agent system is through emergent communication (Sukhbaatar, Szlam, and Fergus 2016; Lazaridou, Peysakhovich, and Baroni 2017; Foerster et al. 2016). In contrast to training a dialogue system in a supervised way, emergent communication supports development of a grounded, goal-oriented and compositional language (Mordatch and Abbeel 2018; Lazaridou et al. 2018; Resnick* et al. 2019). The agents develop a language from scratch in order to solve a task in an end-to-end virtual environment, and this allows an extensive study of the communication protocols. Such studies contribute to the long-standing quest
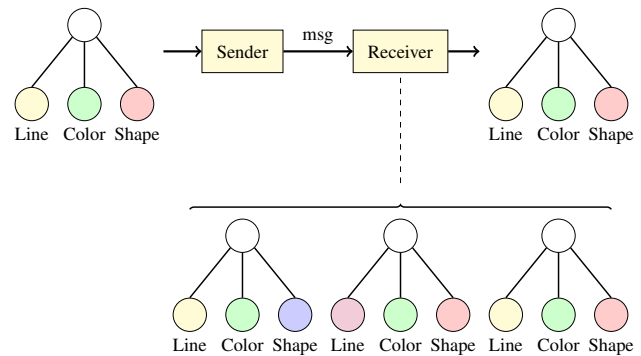
---

[*]Equal Contribution

Figure 1: Intuition behind the graph referential game. The colored graph nodes correspond to independent properties (line, color, shape). The receiver identifies the target graph among distractors based on the message transmitted by the sender.

to develop multi-agent systems that support a biologically-inspired evolution of cooperation through language (Brennan and Clark 1996; Smith, Brighton, and Kirby 2003).

Existing environments for emergent communication use sequences of one-hot vectors or images as the input data (Lazaridou et al. 2018; Evtimova et al. 2018; Bouchacourt and Baroni 2018). The degree of structure in the training samples was found to affect the evolution of compositional understanding (Smith, Kirby, and Brighton 2003; Kirby 2014; Raviv and Arnon 2018), with more structured prelinguistic representations leading to a more compositional language (Lazaridou et al. 2018). Compositionality is one of the most desired properties in a communication protocol because it allows the agents to understand an (in principle) infinite number of complex structures through a finite vocabulary.

Motivated by the properties of graph representations—namely, their explicit structural bias (Hamilton, Ying, and Leskovec 2017), wider array of applications than when using sequences and images, and ability to encode rich compositional properties—we introduce graph representation learning to emergent communication.

Our contributions are as follows:

- We propose a graph referential game with varying degrees of complexity, based on the number of distractors, vocabulary size, properties, and the corresponding types present in the graph.

- We provide baselines using Graph Convolutional Networks (Kipf and Welling 2017) and empirically show that these models generalize to graphs outside of the training set.

- We analyze the resulting communication protocol and find that the agents are able to make use of the available symbols in an efficient way.

- We show that the communication channel is robust in terms of permutation invariance and that it promotes co-operation between the agents.

Environment

Multi Agent Reinforcement Learning

We define the game with agents deployed in a Multi Agent Reinforcement Learning (MARL) framework. We consider multi-agent Markov games (Littman 1994). A Markov game for N agents is a partially observable Markov Decision Process (MDP) defined by: a set of states S describing the state of the world and the possible joint configuration of all the agents, a set of observations $O^1, \ldots, O^N$ for each agent, a set of actions for each agent $A^1, \ldots, A^N$, a transition function $T : S \times A^1 \ldots A^N \to S$ determining a distribution over the next states, and a reward for each agent i, which is a function of the state and the agent's action $r^i : S \times A^i \to R$. Agents choose their actions according to a stochastic policy $\pi_{\theta^i} : O^i \times A^i \to [0, 1]$, where θ^i are the parameters of the policy. Each agent i aims to maximize its own total expected return $R^i = \sum_{t=0}^{T} \gamma^t r_t^i$, where γ is a discount factor and T is the time horizon.

In this paper we refer to the Lewis Signalling Game (Lewis 1969), which is extensively used in multi-agent communication. This referential game (Lazaridou et al. 2018) involves two agents: the sender and the receiver. The agents communicate with each other in order to learn to distinguish a given *target* among a set of *distractors* (other samples from the target distribution). The sender only 'sees' the target and produces a message. The message is received by the receiver along with a set of distractors and the target. The agents are rewarded with a reward $r \in R$ if they correctly identify the target. This framework can be considered as a cooperative partially-observable Markov game. The target and the distractors represent permutations of combinatorial properties represented as symbolic vectors, images, graphs or any efficient data structure able to encode composed entities.

Graph Referential Game

In our game, we represent the set of properties as unique nodes in a graph. Unlike sequential encoding, this representation allows us to build a hierarchy of concepts, where parent nodes are composed of basic properties encoded in their children. Although this can be achieved using sequences as

well, we hypothesize that using graph representations improves the compositional understanding of the agents.

We use p properties of t types, which amounts to t^p unique combinations. As shown in Figure 1, each object is represented using a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ corresponds to the set of all nodes representing unique properties, and a 'central' node such that $|\mathcal{V}| = p + 1$. The set $\mathcal{E}$ comprises undirected edges, which connect two nodes with a relation. In our simple game, $\mathcal{E}$ consists of the edges between the central node and its children, that represent individual properties, such that $|\mathcal{E}| = p$. All of the nodes except the central node are represented using node features. The node features consist of a concatenation of the property encoding and the type encoding (represented as one-hot vectors). The central node is encoded as an empty node and no edge features are used.

For the purpose of providing a graph baseline, we use one level of concepts. This can be easily extended to a deeper hierarchy of concepts using the aforementioned graph representations. The target and distractors are randomly sampled from the set of all possible graphs without replacement. Each *sample* in the game dataset $\mathcal{D}$ consists of a target graph d^* and the set of K distractors. We obtain a collection of these samples and create the train, validation and test splits (60%/20%/20%).

The sender f_θ and the receiver g_ϕ are parameterized using graph convolutional networks (GCNs) (Kipf and Welling 2017). The sender takes the target graph d^* as input and produces a softmax distribution over the vocabulary V, where V refers to the finite set of all distinct messages generated by the sender. In this work the message always comprises one word, i.e. a single symbol from V. Similar to (Sukhbaatar, Szlam, and Fergus 2016; Mordatch and Abbeel 2018), we use the 'straight through' version of Gumbel-Softmax (Jang, Gu, and Poole 2017; Maddison, Mnih, and Teh 2017) during training to make the message discrete. At test time, we take the argmax over this distribution.

The receiver takes two inputs: the discretized message m sent by the sender along with the set of distractors K and the target d^*. It then outputs a softmax distribution over the $|K| + 1$ embeddings representing each graph. We formally define this as follows:

$$m(d^*) = \texttt{Gumbel-Softmax}(f_\theta(d^*))$$
$$o(m, \{K, d^*\}) = g_\phi(m, \{K, d^*\})$$

For reference, the graph convolution network is defined as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)})$$

where $H^{(l)}$ refers to the lth layer in the network, σ is the non-linearity, W corresponds to the weight matrix of the lth layer, and $\tilde{D}$ and $\tilde{A}$ represent the normalized degree matrix and adjacency matrix of a graph, respectively. In order to compute the graph embedding, we experimented with the standard graph pooling methods: mean, sum and max functions. We found no significant difference in performance, and thus use mean pooling throughout the experiments presented in this paper. We used the Deep Graph Library (Wang et al. 2019) when using graphs and EGG (Kharitonov et al.
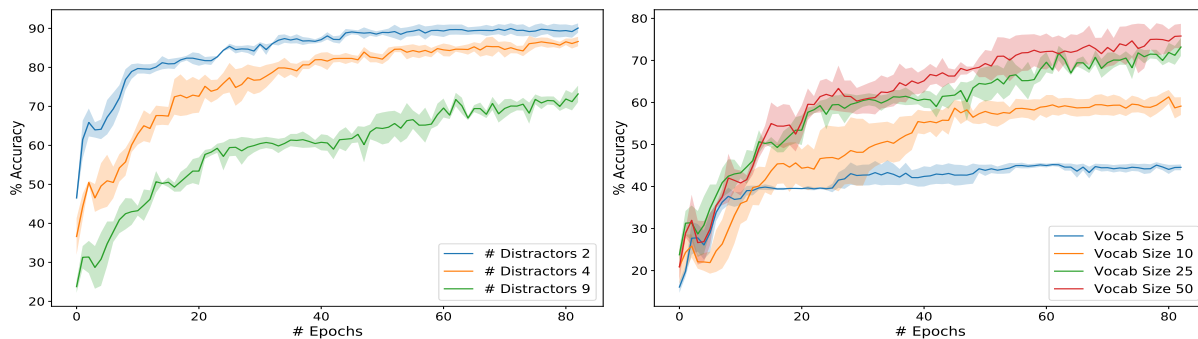
Figure 2: Learning curves showing the performance of the agents on the test set. Left: We vary the number of distractors and fix the vocabulary size (25). Right: We vary the vocabulary size for a fixed number of distractors (4).

Vocab Size	2 distractors	4 distractors	9 distractors
5	$5.0 \pm 0.0 \ (100 \pm 0)$	$5.0 \pm 0.0 \ (100 \pm 0)$	$5.0 \pm 0.0 \ (100 \pm 0)$
10	$9.33 \pm 0.47 \ (93.33 \pm 4.71)$	$9.33 \pm 0.47 \ (93.33 \pm 4.71)$	$9.0 \pm 0.81 \ (90 \pm 8.16)$
25	$12.66 \pm 0.47 \ (50.66 \pm 1.88)$	$14.66 \pm 0.47 \ (58.66 \pm 1.88)$	$13.33 \pm 0.47 \ (53.33 \pm 1.88)$
50	$14.0 \pm 1.41 \ (28.0 \pm 2.83)$	$16.33 \pm 0.94 \ (32.66 \pm 1.89)$	$16.66 \pm 0.47 \ (33.33 \pm 0.94)$

Table 1: The expected number of symbols the system used per number of distractors. In the parentheses we include the percentage of symbols used for the given vocabulary size. All the values are averaged across three different random seeds and standard errors are shown.

2019) for building the framework while the whole codebase was written using PyTorch (Paszke et al. 2019).

Results & Analysis

We present the results for the graphs of $p = 3$ properties and $t = 4$ types. In Figure 2, we confirm that an increase in the number of distractors leads to a higher complexity of the game (measured by the decrease in the performance as the complexity grows). The baseline accuracy for a game with 2, 4 and 9 distractors is 33%, 20% and 10%, respectively, which corresponds to choosing the target at random. A naive way to solve this game would be to learn unique symbols for each unique graph. For our game, this means that the agents would require at least $4^3 = 64$ different symbols in the vocabulary. However, we observe that the agents are able to solve the game with a fraction of this vocabulary. We posit that this is due to the agents learning some compositional properties that are encoded in the graph data structure and are essential to solving the game.

In the right image of Figure 2, we observe that there exists a lower bound on the size of the vocabulary with which the agents are able to achieve high accuracy on the task. This hypothesis is also supported in Table 1, which shows the symbol usage across different sizes of the vocabulary. Figure 2 (Left) shows that the models achieve the accuracy of 90% for 2 distractors with less than 25 symbols. This implies that the sender and the receiver learnt to refer to more than one graph using the same symbol. We hypothesize that graphs mapped to one symbol share a similar structure. Another observation is that the variance of the test accuracies was found to be lower for the higher vocabulary size. We attribute this to the stability in training when there is no pressure in the message

channel and higher variance of gradients due to approximate backpropagation (Gumbel-Softmax).

In Figure 3, we analyze the robustness of the communication protocols learnt by the sender/receiver. For each symbol i in the vocabulary, we collected the set of correct test samples D_i where the sender used the corresponding symbol to represent the target graph. Each subplot in Figure 3 represents the distribution of D_i over the whole vocabulary size $|V| = 10$. We observe that in all cases the symbol sent by the sender (referred to by the title in each subplot and the position of the bar) is the one that makes the receiver correctly identify the target graph. We think that it shows that the receiver actually cooperates with the sender and uses the message to identify the target among distractors.

We also analyzed the behavior of the receiver when presented with a shuffled set of graphs. In each experiment, the position of the target is permuted across all of the possible $|K| + 1$ positions. We observe that the receiver is still able to correctly identify the target based on the message sent from the sender. We thus posit that the agents learnt an order-invariant representation of the graphs and not some positional information about the ordering of the graphs.

Future Work

We present results of our work in progress. We are currently designing the set of distractors in a way that allows us to study systematic generalization in our game. One can design the game samples such that the target and the distractors differ in k types, where $k = 1, \ldots, p$ and p is the number of properties. In order to study compositionality in the emerged language, we will extend the sender with a sequence decoder and produce messages of multiple symbols.
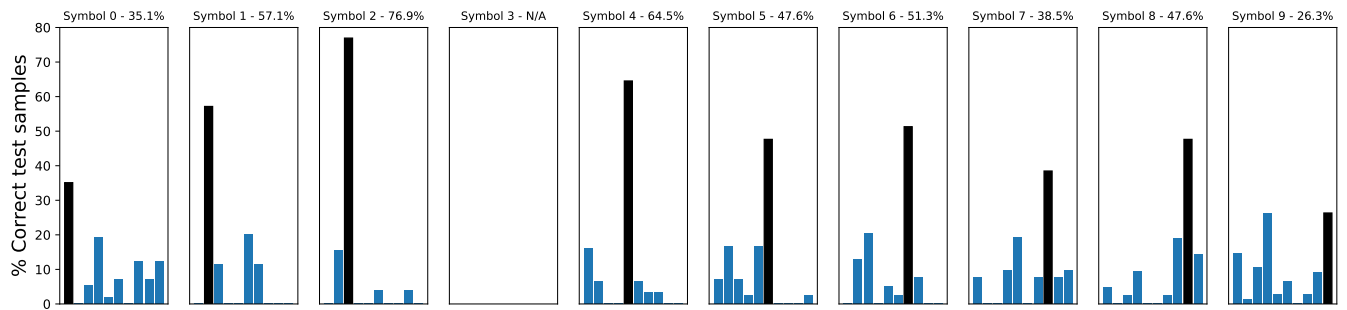
Figure 3: Robustness of the communication protocol learnt by the sender/receiver. We show that for a symbol different than the one sent by the sender (black), the receiver does not correctly identify the target. For more details, refer to Section . The above chart is for vocab size 10 and number of distractors 4. Note that there were no test samples found where the sender used the symbol 3.

It would be interesting to see if the sender parameterized with a Graph2Seq (Xu et al. 2019) network is able to generate sentences close to natural language in terms of compositionality. Another possible direction can be to use a deeper hierarchy of concepts in the graph representations. We expect that hierarchical concepts will show the advantage of using graph representations over sequences and images in multi-agent communication.

Conclusion

We proposed a new referential game defined on graphs. We showed that agents using simple graph neural networks generalized to new combinations of familiar concepts and types. We found that the agents made an efficient use of the vocabulary, learnt an order-invariant representation of the target graph, and solved the graph games with a varying number of distractors through communication and cooperation.

References

Bellmund, J. L. S.; Gärdenfors, P.; Moser, E. I.; and Doeller, C. F. 2018. Navigating cognition: Spatial codes for human thinking. *Science* 362(6415).

Bouchacourt, D., and Baroni, M. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 981–985. Brussels, Belgium: Association for Computational Linguistics.

Brennan, S. E., and Clark, H. H. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology. Learning, Memory, and Cognition* 22(6):1482–1493.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics.

Evtimova, K.; Drozdov, A.; Kiela, D.; and Cho, K. 2018. Emergent communication in a multi-modal, multi-step refer-

ential game. In *International Conference on Learning Representations*.

Foerster, J.; Assael, I. A.; de Freitas, N.; and Whiteson, S. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. 2137–2145.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. 1024–1034.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Kharitonov, E.; Chaabouni, R.; Bouchacourt, D.; and Baroni, M. 2019. EGG: a toolkit for research on emergence of lanGuage in games. 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): System Demonstrations*, 55–60. Hong Kong, China: Association for Computational Linguistics.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kirby, S. 2014. Iterated learning and the evolution of language. *Current Opinion in Neurobiology* 7.

Lazaridou, A.; Hermann, K. M.; Tuyls, K.; and Clark, S. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*.

Lazaridou, A.; Peysakhovich, A.; and Baroni, M. 2017. Multi-Agent Cooperation and the Emergence of (Natural) Language. In *International Conference on Learning Representations*.

Lewis, D. 1969. *Convention: A philosophical study*. Harvard University Press.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, volume 157, 157–163.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*.

Mordatch, I., and Abbeel, P. 2018. Emergence of grounded compositional language in multi-agent populations. In *AAAI Conference on Artificial Intelligence*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc. 8024–8035.

Raviv, L., and Arnon, I. 2018. Systematicity, but not compositionality: Examining the emergence of linguistic structure in children and adults using iterated learning. *Cognition* 181:160–173.

Resnick*, C.; Gupta*, A.; Foerster, J. N.; Dai, A. M.; and Cho, K. 2019. Capacity, bandwidth, and compositionality in emergent language learning. *ArXiv* abs/1910.11424.

Smith, K.; Brighton, H.; and Kirby, S. 2003. Complex Systems In Language Evolution: The Cultural Emergence Of Compositional Structure. *Advances in Complex Systems (ACS)* 6(04):537–558.

Smith, K.; Kirby, S.; and Brighton, H. 2003. Iterated learning: A framework for the emergence of language. *Artif. Life* 9(4):371–386.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning multiagent communication with backpropagation. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. 2244–2252.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 3104–3112.

Wang, M.; Yu, L.; Zheng, D.; Gan, Q.; Gai, Y.; Ye, Z.; Li, M.; Zhou, J.; Huang, Q.; Ma, C.; Huang, Z.; Guo, Q.; Zhang, H.; Lin, H.; Zhao, J.; Li, J.; Smola, A. J.; and Zhang, Z. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Xu, K.; Wu, L.; Wang, Z.; Feng, Y.; Witbrock, M.; and Sheinin, V. 2019. Graph2seq: Graph to sequence learning with attention-based neural networks.