Compositionality and Capacity in Emergent Languages

Abhinav Gupta*
MILA
abhinavg@nyu.edu

Cinjon Resnick*
New York University
cinjon@nyu.edu

Jakob Foerster
Facebook AI Research
jnf@fb.com

Andrew M. Dai
Google AI
adai@google.com

Kyunghyun Cho
New York University
Facebook AI Research
kyunghyun.cho@nyu.edu

Abstract

Recent works have discussed the extent to which emergent languages can exhibit properties of natural languages particularly learning compositionality. In this paper, we investigate the learning biases that affect the efficacy and compositionality in multi-agent communication in addition to the communicative bandwidth. Our foremost contribution is to explore how the capacity of a neural network impacts its ability to learn a compositional language. We additionally introduce a set of evaluation metrics with which we analyze the learned languages. Our hypothesis is that there should be a specific range of model capacity and channel bandwidth that induces compositional structure in the resulting language and consequently encourages systematic generalization. While we empirically see evidence for the bottom of this range, we curiously do not find evidence for the top part of the range and believe that this is an open question for the community.

1 Introduction

Compositional language learning in the context of multi agent emergent communication has been extensively studied (Foerster et al., 2016; Lazaridou et al., 2017; Baroni, 2020). These works have found that while most emergent languages do not tend to be compositional, they can be guided towards this attribute through artificial task-specific constraints (Harding Graesser et al., 2019; Lee et al., 2018; Słowik et al., 2020).

In this paper, we focus on how a neural network, specifically a generative one, can learn a compositional language. Moreover, we ask how this can occur without task-specific constraints. To accomplish this, we first define what is a language and what we mean by compositionality. In tandem, we introduce *precision* and *recall*, two metrics that help us measure how well a generative model at large has learned a grammar from a finite set of training instances. We then use a variational autoencoder with a discrete sequence bottleneck to investigate how well the model learns a compositional language, in addition to what affects that learning. This allows us to derive *residual entropy*, a third metric that reliably measures compositionality in our particular environment. We use this metric to cross-validate precision and recall.

Our paper is most similar to Kottur et al. (2017), which showed that compositional language arose only when certain constraints on the agents are satisfied. While the constraints they examined were either making their models memoryless or having a minimal vocabulary in the language, we hypothesized about the importance for agents to have small capacity relative to the number of concepts to which they are exposed. Each of Verhoef et al. (2016); Kirby et al. (2015); Zaslavsky et al. (2018) examine the trade-off between expression and compression in both emergent and natural languages, in addition to how that trade-off affects the learners. We differ in that we target a specific aspect of the agent (capacity) and ask how that aspect biases the learning.

2 Compositional Language and Learning

We consider the problem of learning an underlying language $L^\star$ from a finite set of training strings randomly drawn from it: $D = \{s|s \sim G^\star\}$ where $G^\star$ is the minimal length generator associated with $L^\star$. We assume $|D| \ll |L^\star|$ and our goal is to use D to learn a language L that approximates $L^\star$ as well as possible. We know that there exists an equivalent generator G for L, and so our problem becomes estimating a generator from this finite set rather than reconstructing an entire set of strings belonging to the original language L^*. We cast the problem of estimating a generator G as density modeling, in which case the goal is to estimate a distribution $p(s)$. Sampling from $p(s)$ is equivalent

*These two authors contributed equally.

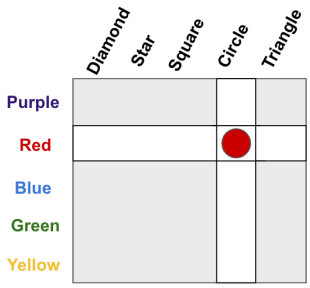


Figure 1: The grid above shows five shapes and five colors. Agents with a non-compositional language can use this shared map to communicate "Red Circle" with only $\lceil \log_2 5^2 \rceil = 5$ bits. If they instead used a compositional language, it would require $\lceil \log_2 5 \rceil = 3$ bits for each concept for a total of 6 bits to convey the string. On the other hand, the agent needs 25 memory slots to store the concepts in the former case but only 10 slots in the compositional case. This trade-off exemplifies the motivation for our investigation because it suggests that a key driver of compositionality in language is the capacity of an agent relative to the total number of objects in its environment.

to generating a string from the generator G.

Evaluation metrics When the language was learned perfectly, any string sampled from the learned distribution $p(s)$ must belong to $L^\star$. Also, any string in $L^\star$ must be assigned a non-zero probability under $p(s)$. Otherwise, the set of strings generated from this generator, implicitly defined via $p(s)$, is not identical to the original language $L^\star$. This observation leads to two metrics for evaluating the quality of the estimated language with the distribution $p(s)$, *precision* and *recall*:

$$\text{Precision}(L^\star, p) = \frac{1}{|L^\star|} \sum_{s \in L} \mathbb{I}(s \in L^\star) \quad (1)$$

$$\text{Recall}(L^\star, p) = \sum_{s \in L^\star} \log p(s) \quad (2)$$

where $\mathbb{I}(x)$ is the indicator function. These metrics are designed to be fit for any compositional structure rather than one-off evaluation approaches.

Our setup We simplify and assume that each of the characters in the string $s \in L^\star$ correspond to underlying concepts. While the inputs are ordered according to the sequential concepts, our model encodes them using a bag of words (BoW) representation.

The speaker f_θ is parameterized using a recurrent policy which receives the sequence of concatenated one-hot input tokens of s and converts each of them to an embedding. It then runs an LSTM non-autoregressively for l timesteps taking the flattened representation of the input embeddings as its input and linearly projecting each result to a probability distribution over $\{0, 1\}$. This results in a sequential Bernoulli distribution over l latent variables: $f_\theta(z|s) = \prod_{t=1}^{l} p(z_t|s;\theta)$. From this distribution, we can sample a latent string $z = (z_1, \ldots, z_l)$.

The listener g_ϕ receives z and uses a BoW representation to encode them into its own embedding space. Taking the flattened representation of these embeddings as input, we run an LSTM for $|\mathcal{N}|$ time steps, each time outputting a probability distribution over the full alphabet Σ: $g_\phi(s|z) = \prod_{j=1}^{|\mathcal{N}|} p(s_j|z;\phi)$.

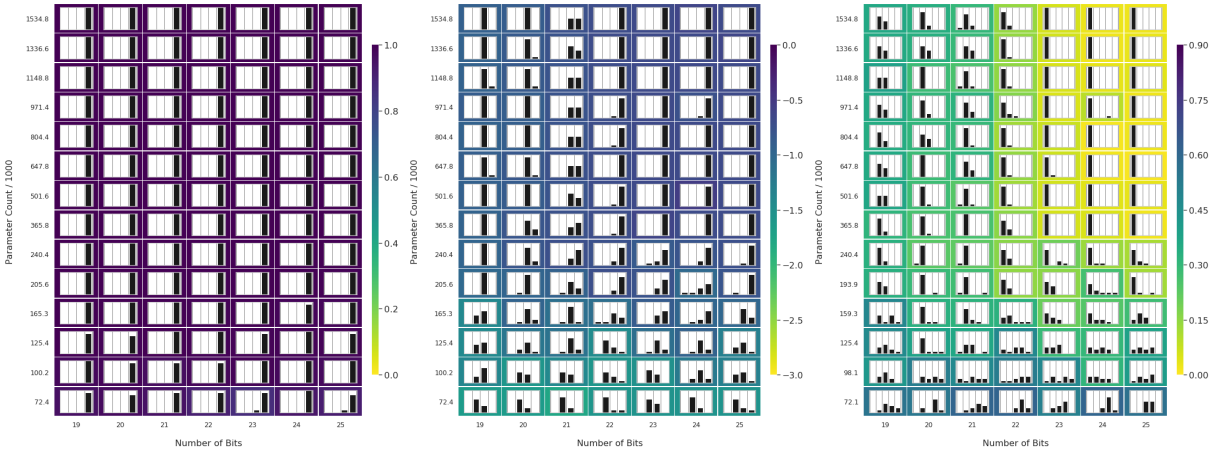
To train the whole system end-to-end (Sukhbaatar et al., 2016; Mordatch and Abbeel, 2018) via backpropogation, we apply a continuous approximation to z_t that depends on a learned temperature parameter τ. We use the 'straight-through' version of Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to convert the continuous distribution to a discrete distribution for each z_t. The final sequence of one hot vectors encoding z is our *message*, which is passed to the listener g_ϕ.

The prior p_λ encodes the *message* z using a BoW representation. It gives the probability of z according to the prior (binary) distribution for each z_t and is defined as: $p_\lambda(z) = \prod_{t=1}^{l} p(z_t|\lambda)$.

This can be used both to compute the prior probability of a latent string and also to efficiently sample from p_λ using ancestral sampling. Penalizing the KL divergence between the speaker's distribution and the prior distribution encourages the emergent protocol to use latent strings that are as diverse as possible.

Hypotheses on compositionality Under this framework for language learning, we can make the following observations. If the length of the latent sequence $l < \log_2 |L^\star|$, it is impossible for the model to avoid the failure case because there will be $|L^\star| - 2^l$ strings in $L^\star$ that cannot be generated from the trained model. Consequently, recall cannot be maximized. However, this may be difficult to check using the sample-based estimate as the chance of sampling $s \in L^\star \setminus \int g_\phi(s|z)p_\lambda(z)\mathrm{d}z$ decreases proportionally to the size of $L^\star$. This is especially true when the gap $|L^\star| - 2^l$ is narrow.

When $l \geq \log_2 |L^\star|$, there are three cases. The first is when there are not enough parameters θ to learn the underlying compositional grammar, in



(a) Precision (b) Recall (c) Entropy

Figure 2: Histograms showing precision, recall (defined in § 2), and entropy (defined in § 3) over the test set. We show results for bits 19 to 25 and parameter range $72k$ to $1534k$ (details in § 3). Each bit/parameter combination is trained for 10 seeds over $200k$ steps.

which case $L^\star$ cannot be learned. The second case is when the number of parameters $|\theta|$ is greater than that required to store all the training strings, i.e., $|\theta| = O(l|D|)$. Here, it is highly likely for the model to overfit as it can map each training string with a unique latent string without having to learn any of $L^\star$'s compositional structure. Lastly, when the number of parameters lies in between these two poles, we hypothesize that the model will capture the underlying compositional structure and exhibit systematic generalization (Bahdanau et al., 2019).

3 Experiments

Models and Learning The task is to communicate 6 concepts, each of which have 10 possible values with a total dataset size of 10^6. We train the proposed VAE We gradually decrease the number of LSTM units from the base model by a factor $\alpha \in (0, 1]$. This is how we control the number of parameters ($|\theta|$ and $|\phi|$). We obtain seven models from each of these by varying the length of the latent sequence l from $\{19, 20, 21, 22, 23, 24, 25\}$. These were chosen because we both wanted to show a range of bits and because we need at least 20 bits to cover the 10^6 strings in L^* ($\lceil \log_2 10^6 \rceil = 20$).

Evaluation: Residual Entropy Our setup allows us to design a metric by which we can check the compositionality of the learned language L by examining how the underlying concepts are described by a string. Let p be a sequence of partitions of $\{1, 2, \ldots, l\}$. We define the degree of compositionality as the ratio between the variability of each concept C_i and the variability explained by a latent subsequence $z[p_i]$ indexed by an associated partition p_i. More formally, the degree of compositionality given the partition sequence p is defined as a residual entropy

$$\text{re}(p, L, L^\star) = \frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} \mathcal{H}_L(C_i|z[p_i])/\mathcal{H}_{L^\star}(C_i)$$

where there are $|\mathcal{N}|$ concepts by the definition of our language. When each term inside the summation is close to zero, it implies that a subsequence $z[p_i]$ explains most of the variability of the specific concept C_i, and we consider this situation compositional. The residual entropy of a trained model is then the smallest $\text{re}(p)$ over all possible sequences of partitions $\mathcal{P}$ and spans from 0 (compositional) to 1 (non-compositional) where $\text{re}(L, L^\star) = \min_{p \in \mathcal{P}} \text{re}(p, L, L^\star)$.

3.1 Results

Fig. 3 shows the main findings of our research. In plot (a), we see the parameter counts at the threshold. Below these values, the model cannot solve the task but above these, it can solve it. Further, observe the curve delineated by the lower left corner of the shift from unsuccessful to successful models. This inverse relationship between bits and parameters shows that the more parameters in the model, the fewer bits it needs to solve the task. Note however that it could only solve the task with fewer bits if it was forming a non-compositional code, suggesting that higher parameter models are able to do so while lower parameter ones cannot.

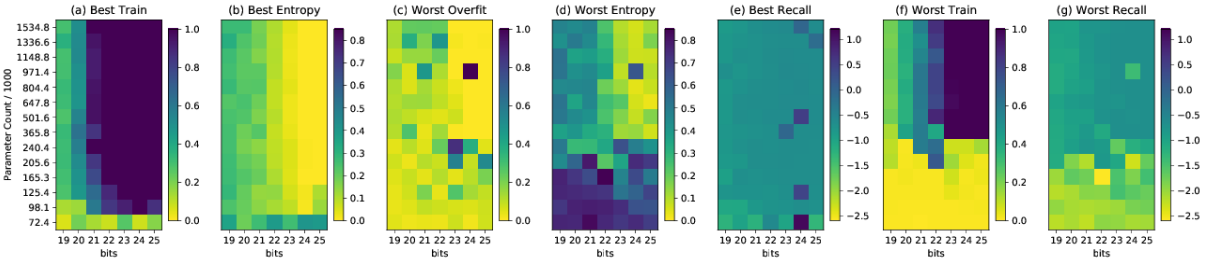


Figure 3: Main results showing best and worst performances of the proposed metrics over 10 seeds. See Section 3.1 for detailed analysis. Panels (a) and (f) show the accuracy of the training data, (b) and (d) show entropy, (e) and (g) show recall over the test data, and (c) plots the max difference in accuracy between training and test.

Observe further that all of our models above the minimum threshold (72,400) have the capacity to learn a compositional code. This is shown by the perfect training accuracy achieved by all of those models in plot (a) for 24 bits and by the perfect compositionality (zero entropy) in plot (b) for 24 bits. Together with the above, this validates that learning compositional codes requires less capacity than learning non-compositional codes. Plot (c) confirms our hypothesis that large models can memorize the entire dataset. The 24 bit model with 971,400 parameters achieves a train accuracy of 1.0 and a validation accuracy of 0.0. Cross-validating this with plots (d) and (g), we find that a member of the same parameter class is non-compositional and that there is one that achieves unusually low recall. We verified that these are all the same seed, which shows that the agents in this model are memorizing the dataset.

Plots (b) and (e) show that our compositionality metrics pass two sanity checks - high recall and perfect entropy can only be achieved with a channel that is sufficiently large (i.e. 24 bits) to allow for a compositional latent representation. Plot (f) shows that while the capacity does not affect the ability to learn a compositional language across the model range, it does change the *learnability*. Here we find that smaller models can fail to solve the task for any bandwidth, which coincides with literature suggesting a link between overparameterization and learnability (Li and Liang, 2018; Du et al., 2019). Finally, as expected, we find that no model learns to solve the task with < 20 bits, validating that the minimum required number of bits for learning a language of size $|L|$ is $\lceil \log(|L|) \rceil$. We also see that no model learns to solve it for 20 bits, which is likely due to optimization difficulties.

We first confirm the effectiveness of training by observing that almost all the models achieve perfect precision (Fig. 2 (a)), implying that $L \subseteq L^\star$, where L is the language learned by the model. This occurs even with our learning which encouraging the model to capture all training strings rather than to focus on only a few training strings. A natural follow-up question is how large is $L^\star \backslash L$. We measure this with recall in Fig. 2 (b), which shows a clear phase transition according to the model capacity when $l \geq 22$. This agrees with what we saw in Fig. 3 and is equivalent to saying $|L^\star \backslash L| \gg 0$ at a value that is close to our predicted boundary of $l = \lceil \log_2 10^6 \rceil = 20$. We attribute this gap to the difficulty in learning a perfectly-parameterized neural network.

These results clearly confirm the first part of our hypothesis - the latent sequence length must be at least as large as $\log |L^\star|$. They also confirm that there is a lowerbound on the number of parameters over which this model can successfully learn the underlying language. We have not been able to verify the upper bound in our experiments, which may require either a more (computationally) extensive set of experiments with even more parameters or a better theoretical understanding of the inherent biases behind learning with this architecture, such as from recent work on overparameterized models (Belkin et al., 2019; Nakkiran et al., 2020).

4 Conclusion

This paper opens the door for a vast amount of follow-up research. All our models were sufficiently large to represent the compositional structure of the language when given sufficient bandwidth. Furthermore, while large models did overfit, this was an exception rather than the rule. We hypothesize that this is due to the large number of examples in our language, which forces the model to generalize, but note that there are likely additional biases at play that warrant further investigation.

Acknowledgements

We would like to thank Marco Baroni and Angeliki Lazaridou for their comments on an earlier version of the paper. We would also like to thank the anonymous reviewers for giving insightful feedback in turn enhancing this work, particularly reviewer two for their thoroughness. Special thanks to Adam Roberts, Doug Eck, Mohammad Norouzi, and Jesse Engel.

References

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375:20190307.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2019. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2137–2145. Curran Associates, Inc.

Laura Harding Graesser, Kyunghyun Cho, and Douwe Kiela. 2019. Emergent linguistic phenomena in multi-agent communication games. In *EMNLP-IJCNLP*, pages 3691–3701, Hong Kong, China. Association for Computational Linguistics.

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

Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. 2015. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967. Association for Computational Linguistics.

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

Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. 2018. Emergent translation in multi-agent communication. In *International Conference on Learning Representations*.

Yuanzhi Li and Yingyu Liang. 2018. Learning over-parameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems 31*, pages 8157–8166. Curran Associates, Inc.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

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

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2020. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*.

Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher Pal. 2020. Exploring structural inductive biases in emergent communication. *arXiv*, 2002.01335.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning multiagent communication with backpropagation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *NeurIPS*, pages 2244–2252. Curran Associates, Inc.

Tessa Verhoef, Simon Kirby, and Bart de Boer. 2016. Iconicity and the emergence of combinatorial structure in language. *Cognitive Science*, 40(8):1969–1994.

Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942.