Entropy Minimization In Emergent Languages

Eugene Kharitonov [1] **Rahma Chaabouni** [1 2] **Diane Bouchacourt** [1] **Marco Baroni** [1 3]

Abstract

There is growing interest in studying the languages that emerge when neural agents are jointly trained to solve tasks requiring communication through a discrete channel. We investigate here the information-theoretic complexity of such languages, focusing on the basic two-agent, one-exchange setup. We find that, under common training procedures, the emergent languages are subject to an entropy minimization pressure that has also been detected in human language, whereby the mutual information between the communicating agent's inputs and the messages is minimized, within the range afforded by the need for successful communication. That is, emergent languages are (nearly) as simple as the task they are developed for allow them to be. This pressure is amplified as we increase communication channel discreteness. Further, we observe that stronger discrete-channel-driven entropy minimization leads to representations with increased robustness to overfitting and adversarial attacks. We conclude by discussing the implications of our findings for the study of natural and artificial communication systems.

1. Introduction

There has recently been much interest in the analysis of the communication systems arising when deep network agents that interact to accomplish a goal are allowed to exchange language-like discrete messages (Lazaridou et al., 2016; Havrylov & Titov, 2017; Choi et al., 2018; Lazaridou et al., 2018; Li & Bowling, 2019; Chaabouni et al., 2020). Understanding the emergent protocol is important if we want to eventually develop agents capable of interacting with each other and with us through language (Mikolov et al., 2016; Chevalier-Boisvert et al., 2019). The pursuit might also provide comparative evidence about how core properties of human language have evolved (Kirby, 2002; Hurford, 2014; Harding Graesser et al., 2019). While earlier studies reported ways in which deep agent protocols radically depart from human language (Kottur et al., 2017; Bouchacourt & Baroni, 2018; Chaabouni et al., 2019; Lowe et al., 2019), we show here that emergent communication shares an important property of the latter, namely a tendency towards *entropy minimization*.

Converging evidence shows that efficiency pressures are at work in language and other biological communication systems (Ferrer i Cancho et al., 2013; Gibson et al., 2019). One particular aspect of communicative efficiency, robustly observed across many semantic domains, is the tendency to minimize lexicon entropy, to the extent allowed by the counteracting need for accuracy (Zaslavsky et al., 2018; 2019). For example, while most languages distinguish grandmothers from grandfathers, few have separate words for mother- and father-side grandmothers, as the latter distinction makes communication only slightly more accurate at the cost of an increase in lexicon complexity (Kemp & Regier, 2012). We show here, in two separate games designed to precisely measure such property, that the protocol evolved by interacting deep agents is subject to the same complexity minimization pressure.

Entropy minimization in natural language has been connected to the Information Bottleneck principle (Tishby et al., 1999). In turn, complexity reduction due to the Information Bottleneck provides a beneficial regularization effect on learned representations (Fischer, 2019; Alemi et al., 2016; Achille & Soatto, 2018a;b). It is difficult to experimentally verify the presence of such effect in human language, but we can look for it in our computational simulations. We confirm that, when relaxing channel discreteness, the entropy minimization property no longer holds, and the system becomes less robust against overfitting and adversarial noise. This in turn raises intriguing questions about the origin of discreteness in human language, that we return to in the conclusion.

[1]Facebook AI Research, Paris, France [2]Cognitive Machine Learning (ENS - EHESS - PSL - CNRS - INRIA) [3]Catalan Institute for Research and Advanced Studies, Barcelona, Spain. Correspondence to: Eugene Kharitonov <kharitonov@fb.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

2. General framework

We establish our results in the context of signaling games (Lewis, 1969), as introduced to the current language emergence literature by Lazaridou et al. (2016) and adopted in several later studies (Havrylov & Titov, 2017; Bouchacourt & Baroni, 2018; Lazaridou et al., 2018). There are two agents, Sender and Receiver, provided with individual inputs at the beginning of each episode. Sender sends a single message to Receiver, and Receiver has to perform an action based on its own input and the received message. Importantly, there is no direct supervision on the message protocol. We consider agents that are deterministic functions of their inputs (after training).

As an example, consider the task of communicating a n-bit number, sampled uniformly at random from $0...2^n - 1$. The full number is shown to Sender, and its k ($0 \leq k \leq n$) least-significant bits are also revealed to Receiver. Receiver has to output the full number, based on the message from Sender and its own input. Would Sender transmit the entire number through its message? In this case, the protocol would be "complex," encoding n bits. Alternatively, Sender could only encode the bits that Receiver does not know, and let Receiver fill in the rest by itself. This emergent protocol would be "simple," encoding only strictly necessary information. We find experimentally that, once the agents are successfully trained to jointly solve the task, the emergent protocol *minimizes the entropy of the messages* or, equivalently in our setup, *the mutual information between Sender's input and messages*. In other words, the agents consistently approximate the simplest successful protocol (in the current example, the one transmitting $\approx n - k$ bits).

We can connect the entropies of Sender and Receiver inputs $\boldsymbol{i}_s$ and $\boldsymbol{i}_r$, messages $\boldsymbol{m}$, Receiver's output (the chosen action) $\boldsymbol{o}$, and ground-truth outputs $\boldsymbol{l}$ by standard inequalities (Cover & Thomas, 2012).[1] Denoting Sender's computation as a function $S : S(\boldsymbol{i}_s) = \boldsymbol{m}$, and Receiver as function $R : R(\boldsymbol{m}, \boldsymbol{i}_r) = \boldsymbol{o}$, we obtain:

$$\begin{aligned} H(\boldsymbol{i}_s) \geq H(S(\boldsymbol{i}_s)) = H(\boldsymbol{m}) \geq H(\boldsymbol{m}|\boldsymbol{i}_r) \geq \\ \geq H(R(\boldsymbol{m}, \boldsymbol{i}_r)|\boldsymbol{i}_r) = H(\boldsymbol{o}|\boldsymbol{i}_r) \approx H(\boldsymbol{l}|\boldsymbol{i}_r), \end{aligned} \quad (1)$$

where the last relation stems from the fact that after successful training $\boldsymbol{o} \approx \boldsymbol{l}$. Note that, since agents are deterministic after training, $H(\boldsymbol{m}) = I(\boldsymbol{i}_s; \boldsymbol{m})$. We can then use these quantities interchangeably.

Our empirical measurements indicate that the entropy of the messages $\boldsymbol{m}$ in the emergent protocol tends to approach the lower bound: $H(\boldsymbol{m}) \to H(\boldsymbol{l}|\boldsymbol{i}_r)$, even if the upper bound $H(\boldsymbol{i}_s)$ is far. that Receiver needs is reduced without changing other parameters, the emergent protocol becomes simpler (lower entropy). In other words, the emergent protocol adapts to minimize the information that passes through it.

Code for our experiments is publicly available at github.com/facebookresearch/EGG/ as a part of the EGG framework (Kharitonov et al., 2019).

[1] We also use the fact that that $H(x) \geq H(g(x))$ for any discrete r.v. x and function g.

3. Methodology

3.1. Games

We study two signaling games. In Guess Number, the agents are trained to recover an integer-representing vector with uniform Bernoulli-distributed components. This simple setup gives us full control over the amount of information needed to solve the task. The second game, Image Classification, employs more naturalistic data, as the agents are jointly trained to classify pairs of MNIST digits (LeCun et al., 1998).

Guess Number We draw an 8-bit integer $0 \leq z \leq 255$ uniformly at random, by sampling its 8 bits independently from the uniform Bernoulli distribution. All bits are revealed to Sender as an 8-dimensional binary vector $\boldsymbol{i}_s$. The last k bits are revealed to Receiver ($0 \leq k \leq 8$) as its input $\boldsymbol{i}_r$. Sender outputs a single-symbol message $\boldsymbol{m}$ to Receiver. In turn, Receiver outputs a vector $\boldsymbol{o}$ that recovers all the bits of z and should be equal to $\boldsymbol{i}_s$.

In this game, Sender has a linear layer that maps the input vector $\boldsymbol{i}_s$ to a hidden representation of size 10, followed by a leaky ReLU activation. Next is a linear layer followed by a softmax over the vocabulary. Receiver linearly maps both its input $\boldsymbol{i}_r$ and the message to 10-dimensional vectors, concatenates them, applies a fully connected layer with output size 20, followed by a leaky ReLU. Finally, another linear layer and a sigmoid nonlinearity are applied. When training with REINFORCE and the Stochastic Computation graph approach (see Sec. 3.2), we increase the hidden layer sizes threefold, as this leads to a more robust convergence.

Image Classification In this game, the agents are jointly trained to classify 28x56 images of two MNIST digits, stacked side-by-side (more details in Supplementary). Unlike Guess Number, Receiver has no side input. Instead, we control the informational complexity of Receiver's task by controlling the size of its output space, i.e., the number of labels we assign to the images. To do so, we group all two-digit sequences 00..99 into $N_l \in \{2, 4, 10, 20, 25, 50, 100\}$ equally-sized classes.

In Sender, input images are embedded by a LeNet-1 instance (LeCun et al., 1990) into 400-dimensional vectors. These embedded vectors are passed to a fully connected layer, followed by a softmax selecting a vocabulary symbol. Receiver embeds the received messages into 400-dimensional vectors, passed to a fully connected layer with a softmax activation returning the class probabilities.

We report hyperparameter grids in Supplementary. In the following experiments, we fix vocabulary to 1024 symbols (experiments with other vocabulary sizes, multi-symbol messages, and larger architectures are reported in Supplementary). No parts of the agents are pre-trained or shared. The loss being optimized depends on the chosen gradient estimation method (see Sec. 3.2). We denote it $\mathcal{L}(\boldsymbol{o}, \boldsymbol{l})$, and it is a function of Receiver's output $\boldsymbol{o}$ and the ground-truth output $\boldsymbol{l}$. When training in Guess Number with REINFORCE, we use a 0/1 loss: the agents get zero loss only when all bits of z are correctly recovered. When training with Gumbel-Softmax relaxation or the Stochastic Computation Graph approach, we use binary cross-entropy (Guess Number) and negative log-likelihood (Image Classification).

3.2. Training with discrete channel

Training to communicate with discrete messages is non-trivial, as we cannot back-propagate through the messages. Current language emergence work mostly uses Gumbel-Softmax relaxation (e.g., Havrylov & Titov, 2017) or REINFORCE (e.g., Lazaridou et al., 2016) to get gradient estimates. We also explore the Stochastic Computation Graph optimization approach. We plug the obtained gradient estimates into Adam (Kingma & Ba, 2014).

Gumbel-Softmax relaxation Samples from the Gumbel-Softmax distribution (a) are reperameterizable, hence allow gradient-based training, and (b) approximate samples from the corresponding Categorical distribution (Maddison et al., 2016; Jang et al., 2016). To get a sample that approximates an n-dimensional Categorical distribution with probabilities p_i, we draw n i.i.d. samples g_i from Gumbel(0,1) and use them to calculate a vector $\boldsymbol{y}$ with components:

$$y_i = \frac{exp\left[(g_i + \log\ p_i)/\tau\right]}{\sum_j exp\left[(g_j + \log\ p_j)/\tau\right]}, \tag{2}$$

where τ is the temperature hyperparameter. As τ tends to 0, the samples $\boldsymbol{y}$ get closer to one-hot samples; as $\tau \to +\infty$, the components y_i become uniform. During training, we use these relaxed samples as messages from Sender, making the entire Sender/Receiver setup differentiable.

REINFORCE by Williams (1992) is a standard reinforcement learning algorithm. In our setup, it estimates the gradient of the expectation of the loss $\mathcal{L}(\boldsymbol{o}, \boldsymbol{l})$ w.r.t. the parameter vector $\boldsymbol{\theta}$ as follows:

$$\mathbb{E}_{\boldsymbol{i}_s,\boldsymbol{i}_r}\mathbb{E}_{\boldsymbol{m}\sim S(\boldsymbol{i}_s),\boldsymbol{o}\sim R(\boldsymbol{m},\boldsymbol{i}_r)}\left[(\mathcal{L}(\boldsymbol{o};\boldsymbol{l}) - b)\nabla_{\boldsymbol{\theta}} \log P_{\boldsymbol{\theta}}(\boldsymbol{m},\boldsymbol{o})\right] \tag{3}$$

The expectations are estimated by sampling $\boldsymbol{m}$ from Sender and, after that, sampling $\boldsymbol{o}$ from Receiver. We use the running mean baseline b (Greensmith et al., 2004; Williams, 1992) as a control variate. We adopt the common trick to add an entropy regularization term (Williams & Peng, 1991; Mnih et al., 2016) that favors higher entropy. We impose entropy regularization on the outputs of the agents with coefficients λ_s (Sender) and λ_r (Receiver).

Stochastic Computation Graph (SCG) In our setup, the gradient estimate approach of Schulman et al. (2015) reduces to computing the gradient of the surrogate function:

$$\mathbb{E}_{\boldsymbol{i}_s,\boldsymbol{i}_r}\mathbb{E}_{\boldsymbol{m}\sim S(\boldsymbol{i}_s)}\left[\mathcal{L}(\boldsymbol{o};\boldsymbol{l}) + sg\left(\mathcal{L}(\boldsymbol{o};\boldsymbol{l}) - b\right)\log P_{\boldsymbol{\theta}}(\boldsymbol{m})\right], \tag{4}$$

where sg denotes *stop-gradient* operation. We do not sample Receiver actions: Its parameter gradients are obtained with standard backpropagation (first term in Eq. 4). Sender's messages are sampled, and its gradient is calculated akin to REINFORCE (second term in Eq. 4). Again, we apply entropy-favoring regularization on Sender's output (with coefficient λ_s) and use the mean baseline.

Role of entropy regularization As we mentioned above, when training with REINFORCE and SCG, we include a (standard) entropy regularization term in the loss which explicitly *maximizes* entropy of Sender's output. Clearly, this term is at odds with the entropy *minimization* effect we observe. In our experiments, we found that high values of λ_s (the parameter controlling Sender's entropy regularization) prevent communication success; on the other hand, a small non-zero λ_s is crucial for successful training. In Sec. 4 we investigate the effect of λ_s on entropy minimization.[2]

3.3. Experimental protocol

In Guess Number, we use all 2^8 possible inputs for training, early stopping and analysis. In Image Classification, we train on random image pairs from the MNIST training data, and use image pairs from the MNIST held-out set for validation. We select the runs that achieved a high level of performance (training accuracy above 0.99 for Guess Number and validation accuracy above 0.98 for Image Classification), thus studying typical agent behavior *provided they succeeded at the game*.

At test time, we select the Sender's message symbol greedily, hence the messages are discrete and Sender represents a (deterministic) function S of its input $\boldsymbol{i}_s$, $\boldsymbol{m} = S(\boldsymbol{i})$. Calculating the entropy $H(\boldsymbol{m})$ of the distribution of discrete messages $\boldsymbol{m}$ is straightforward. In Guess Number, we enumerate all 256 possible values of z as inputs, obtain messages and calculate entropy $H(\boldsymbol{m})$. For Image Classification, we sample image pairs from the held-out set.

The upper bound on $H(\boldsymbol{m})$ is as follow: $H_{max} = 8$ bits (bounded by $H(\boldsymbol{i}_s)$) in Guess Number, and $H_{max} = 10$

[2] The parameter λ_r, that controls Receiver's entropy regularization, does not influence the observed effect.

bits (bounded by vocabulary size) in Image Classification. Its lower bound is equal to $H_{min} = H(\boldsymbol{l}|\boldsymbol{i}_r) = 8 - k$ bits for Guess number. In Image Classification, communication can only succeed if $H(\boldsymbol{m})$ is not less than $H(\boldsymbol{l})$, i.e., $H_{min} = H(\boldsymbol{l}) = \log_2 N_l$, with N_l the number of equally-sized classes we split the images into.

4. Experiments

4.1. Entropy minimization

Guess Number In Figure 1, the horizontal axes span the number of bits of z that Receiver lacks, $8 - k$. The vertical axis reports the information content of the protocol, measured by messages entropy $H(\boldsymbol{m})$. Each integer on the horizontal axis corresponds to a game configuration, and for each such configuration we aggregate multiple (successful) runs with different hyperparameters and random seeds. H_{min} indicates the minimal amount of bits Sender has to send in a particular configuration for the task to be solvable. The upper bound (not shown) is $H_{max} = 8$ bits. Across hyperparameters and random seeds, trainings with Gumbel-Softmax and SCG have success rate above 50%. With REINFORCE success rate is approximately 20%.

Consider first the configurations where Receiver's input is insufficient to answer correctly (at least one binary digit hidden, $k \leq 7$). From Figure 1a, we observe that the transmitted information is strictly monotonically increasing with the number of binary digits hidden from Receiver. Thus, even if Sender sees the very same input in all configurations, a more nuanced protocol is only developed when it is necessary. Moreover, the entropy $H(\boldsymbol{m})$ (equivalently: the transmitted information) stays close to the lower bound. This entropy minimization property holds for all the considered training approaches across all configurations.

Consider next the configuration where Receiver is getting the whole integer z as its input ($k = 8$, the leftmost configuration in Figure 1, corresponding to 0 on x axis). Based on the observations above, one would expect that the protocol would approach zero entropy in this case (as no information needs to be transmitted). However, the measurements indicate that the protocol is encoding considerably more information. It turns out that this information is entirely ignored by Receiver. To demonstrate this, we fed all possible distinct inputs to Sender, retrieved the corresponding messages, and *shuffled* them to destroy any information about the inputs they might carry. The shuffled messages were then passed to Receiver alongside its own (un-shuffled) inputs. The overall performance was not affected by this manipulation, confirming the hypothesis that Receiver ignores the messages. We conclude that in this case there is no entropy minimization pressure on Sender simply because there is no communication. The full experiment is in Supplementary.

We further consider the effect of various hyperparameters. In Figure 1b, we split the results obtained with Gumbel-Softmax by relaxation temperature. As discussed in Sec. 3.2, lower temperatures more closely approximate discrete communication, hence providing a convenient control of the level of discreteness imposed during training (recall that at test time we enforce full discreteness by selecting the symbol greedily). The figure shows that lower temperatures consistently lead to lower $H(\boldsymbol{m})$. This implies that, as we increase the "level of discreteness" at training, we get stronger entropy minimization pressure.

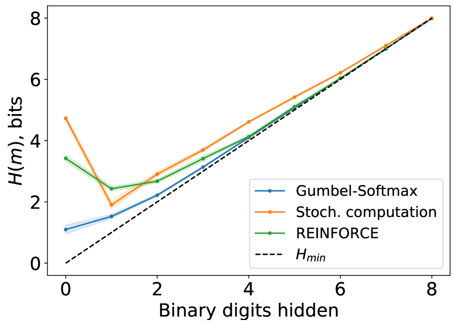
In Figures 1c & 1d, we report $H(\boldsymbol{m})$ when training with Stochastic Graph Optimization and REINFORCE across degrees of entropy regularization. We report curves corresponding to λ_s values which converged in more than three configurations. With REINFORCE, we see a weak tendency for a higher λ_s to trigger a higher entropy in the protocol. However, message entropy stays generally close to the lower bound even in presence of strong exploration, which favors higher entropy in Sender's output distribution.

Image Classification As the models are more complex, we only had consistent success when training with Gumbel-Softmax (success rate is approximately 80%). In Figure 2a we aggregate all successful runs. The information encoded by the protocol grows as Receiver's output requires more information. However, in all configurations, the transmitted information stays well below the 10-bit upper bound and tends to be close to H_{min}. A natural interpretation is that Sender prefers to take charge of image classification and directly pass information about the output label, rather than sending along a presumably more information-heavy description of the input. In Figure 2b, we split the runs by temperature. Again, we see that lower temperatures consistently lead to stronger entropy minimization pressures.

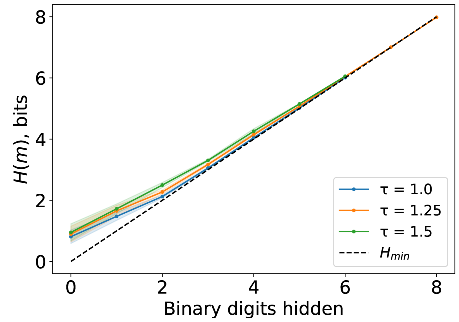
Summarizing, *when communicating through a discrete channel, there is consistent pressure for the emergent protocol to encode as little information as necessary*. This holds across games, training methods and hyperparameters. When training with Gumbel-Softmax, temperature controls the strength of this pressure, confirming the relation between entropy minimization and discreteness.

4.2. Evolution of message entropy during training

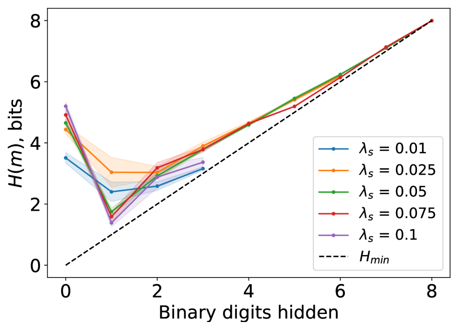
To gain further insights into the minimization trend, we studied the evolution of message entropy during training. We observed that the initial entropy of Sender can be both higher and lower than the minimum entropy H_{min} required for solving the task. Further, we measured how the entropy of the messages changes after each training epoch by applying the same procedure as above, i.e., feeding the



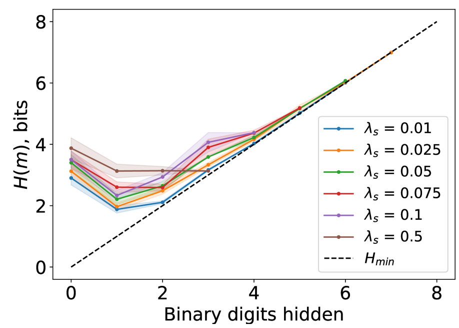
(a) All three training approaches.



(b) Training with Gumbel-Softmax relaxation.



(c) Training with Stochastic Computation Graph.



(d) Training with REINFORCE.

Figure 1. Guess Number: entropy of the messages $\boldsymbol{m}$. Shaded regions represent one standard error of the mean (SEM).

entire dataset to Sender and selecting the message symbol greedily. When message entropy starts higher than H_{min}, it falls close to it during the training. Similarly, when it starts lower than H_{min}, it increases during training. This experiment is reported in Supplementary. Thus, information minimization is not simply due to the difficulty of discovering a higher-entropy protocol during learning, but also due to the complexity of maintaining mutual coordination between the agents.

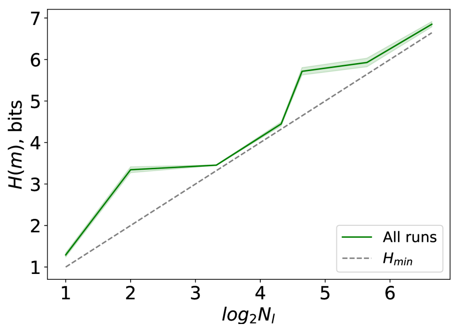
4.3. Representation discreteness and robustness

The entropy minimization effect indicates that a discrete representation will only store as much information as necessary to solve the task. This emergent behavior resembles the Information Bottleneck principle (Tishby et al., 1999; Achille & Soatto, 2018a). The fact that lower training-time temperatures in Gumbel-Softmax optimization correlate with both higher discreteness and a tighter bottleneck (see Sec. 3.3) makes us further conjecture that discreteness is causally connected to the emergent bottleneck. The Information Bottleneck principle has also been claimed to govern entropy minimization in natural language (Zaslavsky et al., 2018; 2019). Bottleneck effects in neural agents and natural language might be due to the same cause, namely communication discreteness.

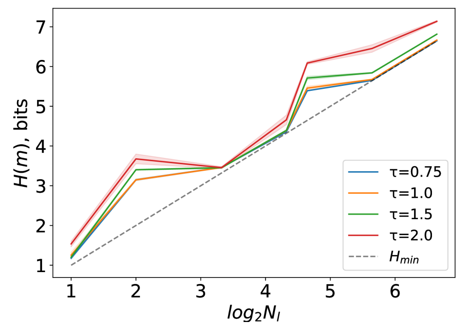
Further, we hypothesize that the emergent discrete bottleneck might have useful properties, since existing (continuous) architectures that explicitly impose a bottleneck pressure are more robust to overfitting (Fischer, 2019) and adversarial attacks (Alemi et al., 2016; Fischer, 2019). We test whether similar regularization properties also emerge in our computational simulations (without any explicit pressure imposed through the cost function), and whether they are correlated with communication channel discreteness. If this connection exists, it also suggests that discreteness might be “beneficial” to human languages for the same reasons.

4.3.1. Robustness to over-fitting

To assess our hypotheses, we consider the Image Classification game ($N_l = 10$) in presence of randomly-shuffled training labels (the test set is untouched) (Zhang et al., 2016). This task allows us to explore whether the discrete communication bottleneck is associated to robustness to overfitting, and whether the latter depends on discreteness level (controlled by the temperature τ of Gumbel-Softmax). We use the same architecture as above. The agents are trained with



(a) Successful runs pooled together.



(b) Successful runs grouped by temperature.

Figure 2. Image Classification: entropy of the messages $\boldsymbol{m}$ in function of log number of target classes, N_l. Shaded regions mark SEM.

Gumbel-Softmax relaxation; at test-time the communication is fully discrete.

We also consider two baseline architectures without the discrete channel. In Linear, the fully connected output layer of Sender is directly connected to the linear embedding input of Receiver. Softmax (SM) places a softmax activation (with temperature) after Sender's output layer and passes the result to Receiver.

We vary temperature and proportion of training examples with shuffled labels. We use temperatures $\tau = 1.0$ and $\tau = 10.0$ (the agents reach a test accuracy of 0.98 when trained with these temperatures on the original training set). SM with $\tau = 1.0$ and $\tau = 10.0$ behave similarly, hence we only report SM with $\tau = 1.0$.

Figure 3a shows *training* accuracy when all labels are shuffled. Linear and SM fit the random labels almost perfectly within the first 150 epochs. With $\tau = 10.0$, GS achieves 0.8 accuracy within 200 epochs. When GS with $\tau = 1.0$ is considered, the agents only start to improve over random guessing after 150 epochs, and accuracy is well below 0.2 after 200 epochs. As expected, test set performance is at chance level (Figure 3b). In the next experiment, we shuffle labels for a randomly selected half of the training instances. Train and test accuracies are shown in Figures 3c and 3d, respectively. All models initially fit the true-label examples (train accuracy ≈ 0.5, test accuracy ≈ 0.97). With more training, the baselines and GS with $\tau = 10.0$ start (over)fitting the random labels, too: train accuracy grows, while test accuracy falls. In contrast, GS with $\tau = 1.0$ does not fit random labels, and its test accuracy stays high. Note that SM patterns with Linear and high-temperature GS, showing that the training-time discretization noise in GS is instrumental for robustness to over-fitting.

We interpret the results as follows. To fully exploit their joint capacity for "successful" over-fitting, the agents need to coordinate label memorization. This requires passing large amounts of information through the channel. With a low temperature (more closely approximating a discrete channel), this is hard, due to a stronger entropy minimization pressure. To test the hypothesis, we run an experiment where all labels are shuffled and a layer of size 400x400 is either added to Sender (just before the channel) or to Receiver (just after the channel). We predict that, with higher τ (less discrete, less entropy minimization pressure), the training curves will be close, as the extra capacity can be used for memorization equally easy in both cases. With lower τ (more discrete, more pressure), the accuracy curves will be more distant, as the extra capacity can only be successfully exploited for memorization when placed *before* the channel. Figures 3e & 3f bear out the prediction.

4.3.2. Robustness to adversarial examples

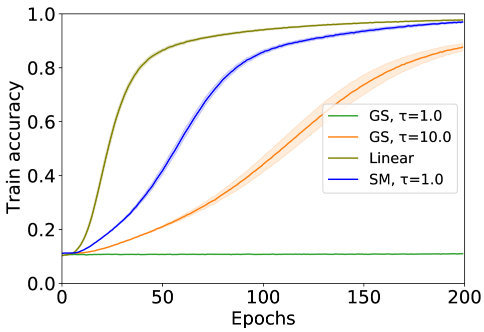
We study next robustness of agents equipped with a relaxed discrete channel against adversarial attacks. We use the same architectures as in the preceding experiment.

We train agents with different random seeds and implement white-box attacks on the trained models, varying temperature τ and the allowed perturbation norm, ϵ. We use the standard *Fast Gradient Sign Method* of (Goodfellow et al., 2014). The original image $\boldsymbol{i}_s$ is perturbed to $\boldsymbol{i}_s^*$ along the direction that maximizes the loss of Receiver's output $\boldsymbol{o} = R(S(\boldsymbol{i}_s))$ w.r.t. the ground-truth class $\boldsymbol{l}$:

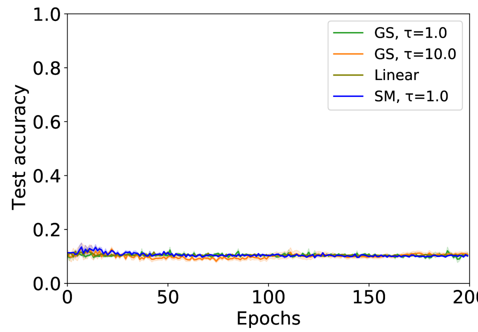
$$\boldsymbol{i}_s^* = clip\left[\boldsymbol{i}_s + \epsilon \cdot sign\left[\nabla_{\boldsymbol{i}_s} \mathcal{L}(\boldsymbol{o}, \boldsymbol{l})\right], 0, 1\right], \qquad (5)$$

where ϵ controls the L_∞ norm of the perturbation. Under an attack with a fixed ϵ, a more robust method will have a higher accuracy. To avoid numerical stability issues akin to those reported by (Carlini & Wagner, 2016), all computations are done in 64-bit floats.

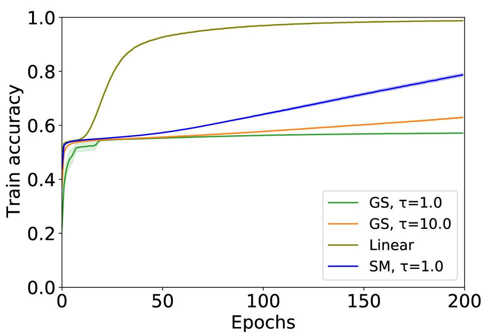
We experiment with two approaches of getting gradients for



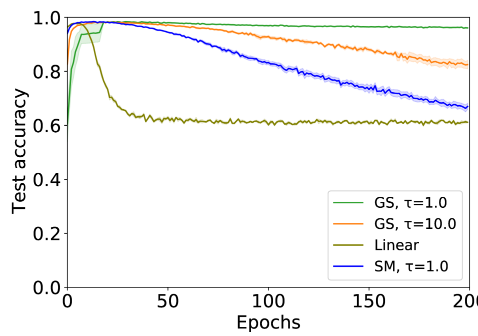
(a) All train labels are shuffled.



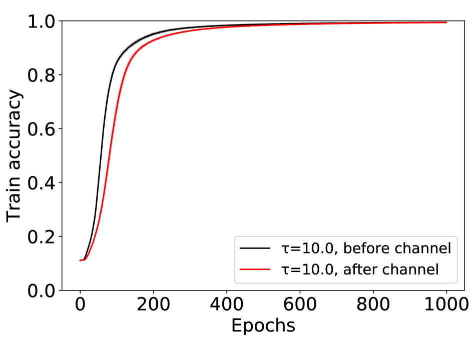
(b) All train labels are shuffled.



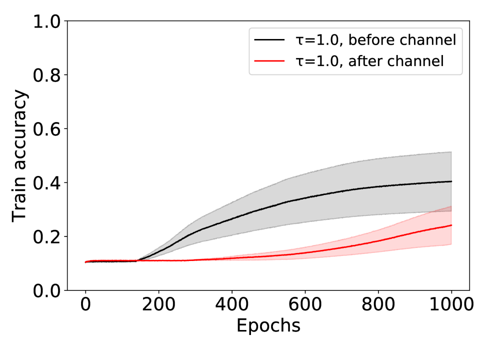
(c) Half of train labels are shuffled.



(d) Half of train labels are shuffled.

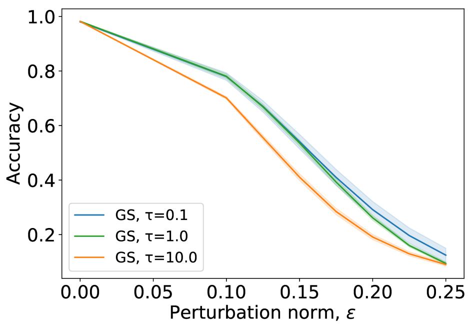


(e) All labels shuffled; Additional layer before channel vs. after channel

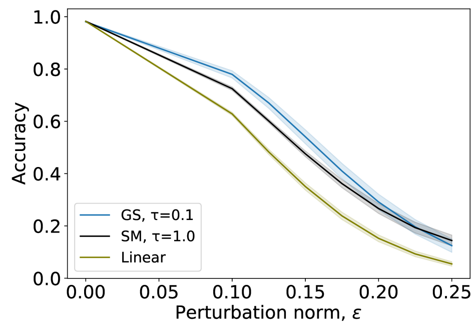


(f) All labels shuffled; Additional layer before channel vs. after channel

Figure 3. Learning in presence of random labels. *GS* (*SM*) denotes models trained with Gumbel-Softmax (Softmax) channel. *Linear* are models with the channel removed.



(a) Robustness vs. temp. τ.



(b) Comparison to the baselines.

Figure 4. Robustness to adversarial examples: higher accuracy given fixed ϵ implies more robustness.

the attack. Under the first approach, the gradient $\nabla_{\boldsymbol{i}_s}\mathcal{L}(\boldsymbol{o}, \boldsymbol{l})$ is estimated using the standard Gumbel-Softmax relaxation. It is possible, however, that the randomization that Gumbel-Softmax uses internally reduces the usefulness of gradients used for the attack. Hence we also experiment with a setup that is easier for an adversary: after training (and during the attack), we replace the Gumbel-Softmax by a softmax non-linearity with the same temperature. We found that performance in these two setups is virtually the same, indicating that the obtained robustness results are independent from the randomization in the channel. Rather, they are due to emergence of well-separated "categories" during training.

As in the preceding experiment, SM behaves similarly with different temperatures (we experimented with $\tau \in \{0.1, 1.0, 10.0\}$): we only report results with $\tau = 1.0$. Figure 4a shows that, as temperature decreases, the accuracy drop also decreases. The highest robustness is achieved with $\tau = 0.1$. Comparison with the baselines (Figure 4b) confirms that relaxed discrete training with $\tau = 0.1$ improves robustness.

In sum, increased channel discreteness makes it harder to transmit large amounts of information, and leads to increased robustness against over-fitting and adversarial examples. Discreteness brings about a bottleneck that has beneficial properties, which might ultimately provide a motivation for why an emergent communication system should evolve towards discreteness.

5. Related Work

We briefly reviewed studies of emergent deep agent communication and entropy minimization in human language in the introduction. We are not aware of earlier work that looks for this property in emergent communication, although Evtimova et al. (2018) used information theory to study protocol development during learning, and, closer to us, Kågebäck et al. (2018) studied the effect of explicitly adding a complexity minimization term to the cost function of an emergent color-naming system.

Discrete representations are explored in many places (e.g., van den Oord et al., 2017; Jang et al., 2016; Rolfe, 2016). However, these works focus on ways to learn good discrete representations, rather than analyzing the properties of representations that are independently emerging on the side. Furthermore, our study extends to agents communicating with variable-length messages, produced and consumed by GRU (Cho et al., 2014) and Transformer (Vaswani et al., 2017) cells (see Supplementary). The sequential setup is specific to language, clearly distinguished from the settings studied in generic sparse-representation work.

Other studies, inspired by the Information Bottleneck principle, control the complexity of neural representations by regulating their information content (Strouse & Schwab, 2017; Fischer, 2019; Alemi et al., 2016; Achille & Soatto, 2018a;b). While they externally impose the bottleneck, we observe that the latter is an intrinsic feature of learning to communicate through a discrete channel.

6. Discussion

Entropy minimization is pervasive in human language, where it constitutes a specific facet of the more general pressure towards communication efficiency. We found that the same property consistently characterizes the protocol emerging in simulations where two neural networks learn to solve a task jointly through a discrete communication code.

In a comparative perspective, we hypothesize that entropy minimization is a general property of discrete communication, independent of specific biological constraints humans are subject to. In particular, our analysis tentatively establishes a link between this property and the inherent difficulty of encoding information in discrete form (cf. the effect of adding a layer before or after the communication bottleneck in the over-fitting experiment).

Exploring entropy minimization in computational simulations provides a flexibility we lack when studying humans. For example, we uncovered here initial evidence that the communication bottleneck is acting as a good regularizer, making the joint agent system more robust to noise and adversarial examples. This leads to an intriguing conjecture on the origin of language. Its discrete nature is often traced back to the fact that it allows us to produce an infinite number of expressions by combining a finite set of primitives (e.g., Berwick & Chomsky, 2016). However, it is far from clear that the need to communicate an infinite number of concepts could have provided the initial pressure to develop a discrete code. More probably, *once such code independently emerged*, it laid the conditions to develop an infinitely expressive language (Bickerton, 2014; Collier et al., 2014). Our work suggests that, because of its inherent regularizing effect, discrete coding is advantageous already when communication is about a limited number of concepts, providing an alternative explanation for its origin.

In the future, we would like to study more continuous semantic domains, such as color maps, where perfect accuracy is not easily attainable, nor desirable. Will the networks find an accuracy/complexity trade-off similar to those attested in human languages? Will other core language properties claimed to be related to this trade-off, such as Zipfian frequency distributions (Ferrer i Cancho & Díaz-Guilera, 2007), concurrently emerge? We would also like to compare the performance of human subjects equipped with novel continuous vs. discrete communication protocols, adopting the methods of experimental semiotics (Galantucci, 2009). We expect discrete protocols to be more general and robust.

Our results have implications for the efforts to evolve agents interacting with each other and with humans through a discrete channel. First, because of entropy minimization, we should not agents to develop a richer protocol than the simplest one ensuring accurate communication. For example, Bouchacourt & Baroni (2018) found that agents trained to discriminate pairs of natural images depicting instances of about 500 high-level categories, such as cats and dogs, developed a lexicon that does not denote such categories, but low-level properties of the images themselves. This makes sense from an entropy-minimization perspective, as talking about the 500 high-level categories demands $\log_2 500$ bits of information, whereas many low-level strategies (e.g., discriminating average pixel intensity in the images) will only require transmitting a few bits. To have agents developing rich linguistic protocols, we must face them with varied challenges that truly demand them.

Second, the focus on a discrete protocol is typically motivated by the goal to develop machines eventually able to communicate with humans. Indeed, discrete messages are not required in multi-agent scenarios where no human in the loop is foreseen (Sukhbaatar et al., 2016). Our results suggest that, long before agents reach the level of complexity necessary to converse with humans, there are independent reasons to encourage discreteness, as it leads to simpler protocols and it provides a source of robustness in a noisy world. An exciting direction for future applied work will be to test the effectiveness of discrete communication as a general form of representation learning.

Acknowledgements The authors thank Emmanuel Dupoux for discussions and the anonymous reviewers for their feedback.

References

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE TPAMI*, 40(12):2897–2905, 2018a.

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018b.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Berwick, R. and Chomsky, N. *Why Only Us: Language and Evolution*. MIT Press, Cambridge, MA, 2016.

Bickerton, D. *More than Nature Needs: Language, Mind, and Evolution*. Harvard University Press, Cambridge, MA, 2014.

Bouchacourt, D. and Baroni, M. How agents see things: On visual representations in an emergent language game. In *EMNLP*, 2018.

Carlini, N. and Wagner, D. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

Chaabouni, R., Kharitonov, E., Dupoux, E., and Baroni, M. Anti-efficient encoding in emergent communication. In *NeurIPS*, 2019.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., and Baroni, M. Compositionality and generalization in emergent languages. In *ACL*, 2020.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. BabyAI: A platform to study the sample efficiency of grounded language learning. In *ICLR*, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

Collier, K., Bickel, B., van Schaik, C., Manser, M., and Townsend, S. Language evolution: Syntax before phonology? *Proceedings of the Royal Society B: Biological Sciences*, 281(1788):1–7, 2014.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, 2012.

Evtimova, K., Drozdov, A., Kiela, D., and Cho, K. Emergent communication in a multi-modal, multi-step referential game. In *ICLR*, 2018.

Ferrer i Cancho, R. and Díaz-Guilera, A. The global minima of the communicative energy of natural communication systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06009, 2007.

Ferrer i Cancho, R., Hernández-Fernández, A., Lusseau, D., Agoramoorthy, G., Hsu, M., and Semple, S. Compression as a universal principle of animal behavior. *Cognitive Science*, 37(8):1565–1578, 2013.

Fischer, I. The conditional entropy bottleneck, 2019. URL https://openreview.net/forum?id=rkVOXhAqY7.

Galantucci, B. Experimental semiotics: A new approach for studying communication as a form of joint action. *Topics in Cognitive Science*, 1(2):393–410, 2009.

Gibson, E., Piantadosi, R. F. S., Dautriche, I., Mahowald, K., Bergen, L., and Levy, R. How efficiency shapes human language. *Trends in Cognitive Science*, 2019. In press.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *JMLR*, 5(Nov):1471–1530, 2004.

Harding Graesser, L., Cho, K., and Kiela, D. Emergent linguistic phenomena in multi-agent communication games. In *EMNLP*, 2019.

Havrylov, S. and Titov, I. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *NIPS*, 2017.

Hurford, J. *The Origins of Language*. Oxford University Press, Oxford, UK, 2014.

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

Kågebäck, M., Dubhashi, D., and Sayeed, A. DeepColor: Reinforcement learning optimizes information efficiency and well-formedness in color name partitioning. In *Proceedings of CogSci*, pp. 1895–1900, Austin, TX, 2018.

Kemp, C. and Regier, T. Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054, 2012.

Kharitonov, E., Chaabouni, R., Bouchacourt, D., and Baroni, M. EGG: a toolkit for research on Emergence of lanGuage in Games. In *EMNLP: System Demonstrations*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirby, S. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

Kottur, S., Moura, J. M., Lee, S., and Batra, D. Natural language does not emerge "naturally" in multi-agent dialog. *arXiv preprint arXiv:1706.08502*, 2017.

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

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

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1990.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lewis, D. Convention harvard university press. *Cambridge, MA*, 1969.

Li, F. and Bowling, M. Ease-of-teaching and language structure from emergent communication. In *NeurIPS*. 2019.

Lowe, R., Foerster, J., Boureau, Y., Pineau, J., and Dauphin, Y. On the pitfalls of measuring emergent communication. In *Proceedings of AAMAS*, pp. 693–701, Montreal, Canada, 2019.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Mikolov, T., Joulin, A., and Baroni, M. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 29–61. Springer, 2016.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

Rolfe, J. T. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

Schulman, J., Heess, N., Weber, T., and Abbeel, P. Gradient estimation using stochastic computation graphs. In *NIPS*, 2015.

Strouse, D. and Schwab, D. J. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

Sukhbaatar, S., Szlam, A., and Fergus, R. Learning multiagent communication with backpropagation. In *NIPS*. 2016.

Tishby, N., Pereira, F., and Bialek, W. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*. University of Illinois Press, 1999.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *NIPS*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.

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

Williams, R. J. and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

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

Zaslavsky, N., Regier, T., Tishby, N., and Kemp, C. Semantic categories of artifacts and animals reflect efficient coding. In *Proceedings of CogSci*, pp. 1254–1260, Montreal, Canada, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.