The spontaneous emergence of discrete and compositional messages

(1) Thoughts on the title? (2) It's hidden on submission, but we need to decide author order as well.

Anonymous ACL submission

Abstract

To be written.

1 Introduction

In a signalling game, artificial agents learn to communicate to achieve a common goal: a sender sees some piece of information and produces a message, this message is then sent to a receiver that must take some action (**??**). If the action is coherent with the sender's initial piece of information, the whole communication stream —the choice of the message and its interpretation— is reinforced. For instance, in a referential game, sender and receiver see a set of objects, and the sender knows which of these the receiver must pick; the sender thus sends a message to the receiver, who must interpret it to pick up the right object (**????**).

This setting has been used to study the factors influencing the emergence of various fundamental properties of natural language, such as *compositionality* (**??????**). In this paper, we add focus on two other so-called 'design features' of natural language (**?**): *discreteness* (i.e. words form clusters in acoustic space), and *displacement* (i.e. efficient communication can occur about objects and facts beyond the immediate context of the conversation).

From an implementation point of view, we follow the recent literature which has shown that a signalling game is essentially an autoencoder setting, with the encoder playing the role of the sender, and the decoder the role of the receiver. In this literature, however, the discreteness of the communication protocol is assumed, since the autoencoders use a (normally sequential and) discrete latent space (**???**).

Our main contribution is a generalization of the current implementation of signalling games as autoencoders, called Function Games. Our implementation covers a broader variety of signalling games, and it crucially incorporates the possibility of displacement and no *a priori* assumption of discreteness. In this setting we propose new analyses for the degree of emergent discreteness, compositionality and displacement. Our main result is that under appropriate conditions, discreteness emerges spontaneously, that is, if the latent space is thought about as a continuous acoustic space, then trained messages form coherent clusters, just like regular words do.

In addition to contributing to our understanding of the emergence of communication protocols with features like natural language, these results have a technical significance: by using a fundamentally continuous communication protocol, with discreteness emerging, we can train end-to-end using standard backpropagation techniques, instead of reinforcement learning algorithms like REINFORCE and its refinements (**???**), which often have high variance and are difficult to use in practice.

2 Related Work

A related line of work attempts to avoid the difficulties of reinforcement learning—which introduces stochastic nodes into a computation graph—by reparameterization and/or non-stochastic estimators (**??**). In the emergent communication case, where the stochastic nodes are discrete (e.g. sampling a message from a sender distribution), the Gumbel-Softmax estimator has become increasingly popular (**??**).

This work enables standard back-propagation to be used for training by optimizing approximations to the true reinforcement learning signal. By contrast, by taking the auto-encoder analogy seriously, we do not approximate the discrete RL learning signal, but rather ask under what conditions discreteness will emerge. Probably can be worded better.

3 Function Games

We here introduce a general communication game setting, which we call Function Games. Our games contain three basic components: (i) a set of contexts C, (ii) a set of actions A, (iii) a family of functions F, from contexts to actions. One play of a Function Game game runs as follows:

1. Nature chooses $f \in F$ and a context $c \in C$.
2. Sender sees the context c and f.
3. Sender sends a message m to Receiver.
4. Receiver sees *a possibly different* context c' and the message m and chooses an action a'.
5. Both are 'rewarded' iff $a' = f(c')$.

Abstractly, the function f represents some piece of knowledge available directly only to Sender, and which determines what action is appropriate in any given context. Two concrete interpretations will help illustrate the variety of communication protocols and goals that this framework encompasses.

Generalized referential games. A reference game is one in which Sender tries to get Receiver to pick the correct object out of a given set (**?????**). Here, contexts are sets of objects (i.e. an $m \times n$ matrix, with m objects represented by n features). Normally (though we will drop this assumption later), $c' = \texttt{shuffled}(c)$: Sender and Receiver see the same objects, but in a different arrangement. Actions are the objects, and the functions $f \in F$ are *choice functions*: $f(c) \in c$ for every context c.

Belief update games. Contexts can represent the (possibly different) belief states of the agents. 'Actions' can represent updated belief states ($A = C$), the different functions in F then representing how to update an agent's beliefs in the light of learning a particular piece of information (passed directly to Sender, and only through the message to Receiver).

What should we cite here? Something from dynamic semantics?

4 Experiment

Because we are interested in the simultaneous emergence both of discrete signals and of compositional messages, we use a Function Game called the Extremity Game designed to incentivize and test rich compositionality (**?**). In this game, it is natural to think of the n dimensions of the objects as gradable properties, e.g., size and darkness, so that a 2D object is determined by a given size and shade of gray. For the functions, we set $F = \{\arg\min_i, \arg\max_i : 0 \leq i < n\}$ and, hopefully, the emerging language should contain messages like '(pick the) MOST + BIG', 'LEAST + DARK', etc.

cite my forthcoming philosophy of science paper instead? pre-print is only on my website, so URL would be sort of de-anonymizing

4.1 Model

Our model resembles an encoder-decoder architecture, with the Sender encoding the context/target pair into a message, and the Receiver decoding the message (together with its context c') into an action. Both the encoder and decoder are multilayer perceptrons with two hidden layers of size 64 and rectified linear activation (ReLU) (**??**). A smaller, intermediate layer without an activation function bridges the encoder and decoder and represents the transformation of the input information to messages. Figure 1 depicts this architecture.

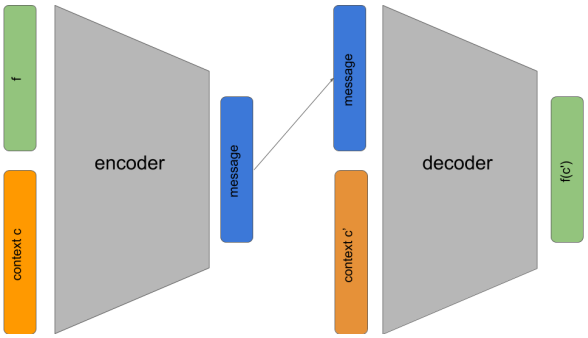


Figure 1: Model architecture caption Do we really need this? It isn't super informative and might use too much space. I can also do something in TikZ if we think it's important and that this one is ugly.

4.2 Game Parameters

We manipulated the following parameters:

- Context identity. In the *shared* setting, Receiver sees a shuffled version of Sender's context ($c' = \texttt{shuffled}(c)$). In the *non-shared* setting, Receiver's context c' is entirely distinct from Sender's. This forces displacement and may incentivize compositional messages, since Sender cannot rely on the raw properties of the target object in communication.
- Context strictness. In *strict* contexts, there is a one-to-one (and onto) correspondence between F and $A = C$ A.[1] In *non-strict* contexts,

[1] These are the contexts used in the original Extremity Game from (**?**). Drop if we need space? Or incorporate in text with 3 words, like 'As in the original Extremity Game from (**?**)'?

an object may be the $\arg\max$ or $\arg\min$ of several dimensions, or of no dimension.

In all experiments, the latent space (message) dimension was always 2,[2] and objects had 5 dimensions. Strict contexts therefore contained 10 objects, while non-strict contexts could contain 5, 10, or 15 objects.

4.3 Training Details

By using a continuous latent space, the entire model, including the communication channel, is differentiable and so can be trained end-to-end using backpropagation to compute gradients. We can drop the previous sentence if we need space. We used the Adam optimizer (**?**) with learning rate 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The model was trained for 5,000 epochs by feeding the network with mini-batches of 64 contexts concatenated with one-hot function selectors. The network's loss is taken as the MSE between the target object $f(c')$ and the object generated by the Receiver. For each setting of the above parameters, we ran 20 trials with different random seeds.

Code and data will be made available once the paper can be de-anonymized.

5 Results

We measure the communicative success of the network by calculating the accuracy of recovering the correct object from c'. The Receiver's prediction is considered correct if its output is closest to $f(c')$ than to all other objects in c'. The recovery accuracy of the different settings is reported in Table 2. While the network handles well the $c \neq c'$ setting (*unshared context*), the model struggles with non-strict contexts. Note that although accuracy is not 100%, it is still well above chance, since e.g. for a context of 10 objects random guessing will yield an expected accuracy of 10%. Table 1 shows the mean loss and recovery accuracy for a network before and after training. The randomly-initialized network's accuracy is at the expected chance level for a 10-object setting.

5.1 Discrete signals

The model's ability to discretize the communication is measured by inspecting the information generated by the intermediate layer. Figure 2 depicts

[2]The model also performs well with messages of size 1, not reported here. Using messages of size 2 makes it easier to inspect the latent space using 2-D visualization.

	Trained	**Untrained**
Accuracy	$63.78\% \pm 0.02$	$9.92\% \pm 0.01$
Loss (MSE)	0.04 ± 0.00	0.35 ± 0.03

Table 1: Mean object recovery accuracy and prediction loss before and after training, for objects of size 5 in a shared, strict context setting (10 objects per context).

	Shared	**Non-shared**
Strict context	$63.78\% \pm 1.63$	$60.22\% \pm 1.56$
Non-strict, 5 objects	$49.37\% \pm 1.67$	$43.55\% \pm 1.69$
Non-strict, 10 objects	$33.06\% \pm 1.47$	$31.89\% \pm 1.63$
Non-strict, 15 objects	$27.58\% \pm 1.30$	$27.95\% \pm 1.24$

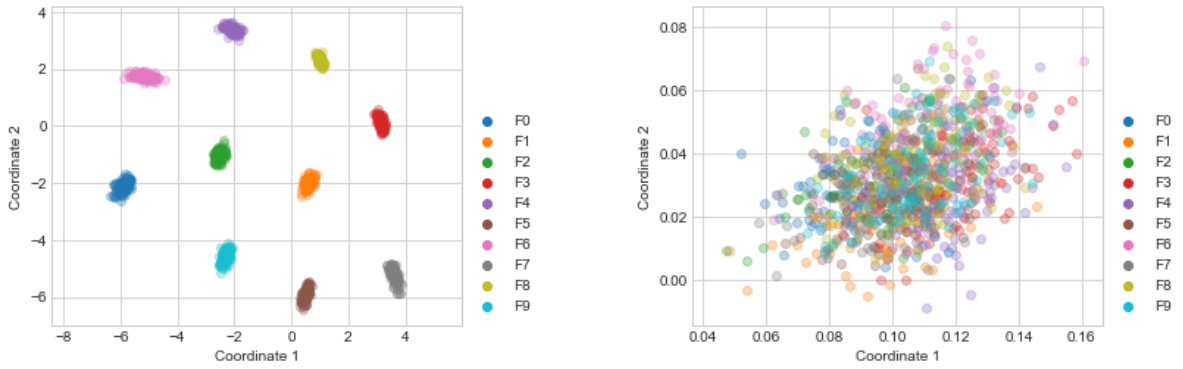
Table 2: Object recovery accuracy for the different model settings.

message vectors sampled from this layer. The same is depicted for an untrained network with randomized weights, where the messages are not yet clustered.

We make the discretization measure concrete by calculating the F1 score between the cluster labels for transmitted messages and the target functions for which they were generated. For this, an unsupervized clustering algorithm is first applied to the message vectors, giving an expected number of clusters (DBSCAN, **?**, with $\epsilon = 0.5$). The cluster labels are then matched with the respective function labels by taking the most common function in each cluster. If message clusters are well separated from one another, the labeling will have less to no confusion and an F1 score closer to 1. The F1 scores for the different model settings are given in Table 3. The model reached near-optimal clusterization measures for both shared and non-shared contexts, and for both strict and non-strict contexts.

Given the clusterization of the message space, we are able to sample unseen messages from each cluster, and test the Receiver's perception of 'artificial' messages. 10 messages are sampled from each cluster, and their average vector is fed to the Receiver. The output object accuracy for these unseen messages is given in Table 4. The model achieves recovery accuracy similar to when messages are generated using actual inputs. This can be paralleled with the phenomenon of Categorical Perception, which describes how continuous sig-

Commented out the actual figures because I can't compile without them.



(a) (b)

Figure 2: (a) Messages sampled from latent space of a trained network, for objects of size 5 and contexts of 10 objects. (b) Messages sampled from an untrained network. Colors represent the $f_i \in F$ input part of the Sender.

	Shared	**Non-shared**
Strict context	1.00 ± 0.00	0.90 ± 0.09
Non-strict, 5 objects	0.99 ± 0.02	0.54 ± 0.15
Non-strict, 10 objects	1.00 ± 0.00	0.99 ± 0.01
Non-strict, 15 objects	1.00 ± 0.00	1.00 ± 0.00

Table 3: Message clusterization F1 scores.

	Shared	**Non-shared**
Strict context	$63.39\% \pm 1.45$	$55.37\% \pm 3.43$
Non-strict, 5 objects	$46.94\% \pm 1.70$	$29.40\% \pm 5.59$
Non-strict, 10 objects	$32.63\% \pm 1.43$	$31.51\% \pm 1.62$
Non-strict, 15 objects	$28.24\% \pm 1.11$	$27.94\% \pm 1.20$

Table 4: Object prediction accuracy using average message from each function cluster.

nals, such as phonemes in an acoustic space, are perceived as stable and discrete, even when the signal is gradually shifted in the continuous space.

canonical reference for Categorical Perception?

5.2 Compositionality

We trained a model to predict various features from the message, to see what they encode. Results show that predicting min/max and param(f) are easy to do. But this leaves open the question: are these features systematically / compositionally encoded in the message?

First test: analogy; include table

Could be limitation of that method, so try: compositionality network. Include table here.

6 Discussion

7 Conclusion