Miss Tools and Mr Fruit: Emergent communication in agents learning about object affordances: Supplementary material

Diane Bouchacourt[1] and Marco Baroni[1,2]
[1]Facebook A.I. Research
[2]ICREA
{dianeb,mbaroni}@fb.com

1 Data and utility computation

This section provides additional details on the dataset we use and the utility function we employ to compute the utilities between fruits and tools. Note that we refer to fruits for conciseness, but some vegetables, such as *carrot* and *potato*, are included.

There are 11 fruits features: *is crunchy, has skin, has peel, is small, has rough skin, has a pit, has milk, has a shell, has hair, is prickly, has seeds* and 15 tools features: *has a handle, is sharp, has a blade, has a head, is small, has a sheath, has prongs, is loud, is serrated, has handles, has blades, has a round end, is adorned with feathers, is heavy, has jaws*. Note that, when we sample instances of each category as explained in Section 2 of the main paper, features are sampled independently. We filter out, however, nonsensical combinations. For example, the features *has prongs*, *has a blade* and *has blades* are treated as pairwise mutually exclusive.

In order to compute the utility for a pair ($tool$, $fruit$), we use three mapping matrices. The mapping matrix $M_T \in \mathbb{R}^{15\times6}$ (Table A1) maps from the space of tool features to a space of more general functional features: *(cut, spear, lift, break, peel, pit remover)*, and similarly $M_F \in \mathbb{R}^{11\times6}$ (Table A2) maps from the space of fruits features to a space of functional features: *(hard, pit, shell, pick, peel, empty inside)*. Finally, the matrix $M \in \mathbb{R}^{6\times6}$ (Table A3) maps the two abstract functional spaces of features together. For example, if an axe sample is described by the vector $t_a \in \mathbb{R}^{1\times15}$ and a nectarine sample is the vector $f_n \in \mathbb{R}^{1\times11}$, the utility is computed as $U(t_a, f_n) = (f_n M_F)M'(t_a M_T)'$ where $'$ denotes transpose. We always add a value of 0.01 to avoid zero utilities. Therefore we can compute the utility of any combination of (possibly new) fruits and tools, as long as it can be described in the corresponding functional representational space. Note that in our case we have the same number of abstract functional features for fruits and tools (6), but they need not be the same. In other words, M need not be a square matrix.

Given the values in the mapping matrices, 5 of the tools features have no impact on the utility computation since they do not affect the scores of the functional tool features (they have only zeros in the mapping matrix M_T). These are: *has a handle, is sharp, has a sheath, is loud, has handles, is adorned with feathers*. Such features only represent realistic aspects of objects and act as noise.

2 Implementation details

2.1 Training and architecture hyperparameters

We update the parameters with RMSProp (Tieleman and Hinton, 2012) with a learning rate of 0.001 and the rest of the parameters left to their Pytorch default value. We use a scalar reward baseline b to reduce variance, learned with Mean Square Error such that $1 + b$ matches the average reward. We clip all gradients at 0.1. For the Message encoder and decoder modules, we embed input and output symbols with dimensionality 50 and then use a RNN with 100 hidden dimensions. The Fruit embedder linear transformation is of output size 100, the Tool embedder is of size 50. The Body module is of size 100. We train the agents with batches of 128 games for a total of 1 million batches. We validate on 12 batches of 100 games, for a total of 1200 validation games, and similarly for testing.

2.2 Test procedure details

The computation of the ME values involves random sampling in steps 1 and 2 of Algorithm 1 so

Tools Feature	Cut	Spear	Lift	Break	Peel	Pit Remover
has a handle	0	0	0	0	0	0
is sharp	0	0	0	0	0	0
has a blade	1	0.5	0	0	1	0
has a head	0	0	0	1	0	0
is small	0	0	0	0	0	0.25
has a sheath	0	0	0	0	0	0
has prongs	0.5	1	0.25	0	0.25	0
is loud	0	0	0	0	0	0
is serrated	0.5	0	0	0	0	0
has handles	0	0	0	0	0	0
has blades	1	0.5	0	0	0.5	0
has a round end	0.25	0	1	0	0	1
is adorned with feathers	0	0	0	0	0	0
is heavy	0	0	0	0.5	0	0
has jaws	0	0	1	0	0	0.5

Table A1: M_T. Rows are dataset tool features, columns are functional tool features.

Fruits Feature	Hard	Pit	Shell	Pick	Peel	Empty inside
is crunchy	1	0	0	0	0	0
has skin	0	0	0	0	1	0
has peel	0	0	0	0	1	0
is small	0	0	0	1	0	0
has rough skin	0	0	0.5	0	0	0
has a pit	0	1	0	0	0	0
has milk	0	0	0	0	0	1
has a shell	0	0	1	0	0	0
has hair	0	0	0	0	0.5	0
is prickly	0	0	0	0	0.5	0
has seeds	0	0	0	0	0	1

Table A2: M_F. Rows are dataset fruit features, columns are functional fruit features.

	Hard	Pit	Shell	Pick	Peel	Empty inside
Cut	1	0	0.5	0	0.5	0
Spear	0	0	0	1	0	0
Lift	0	0	0	0.5	0	1
Break	0.5	0	1	0	0	0
Peel	0	0	0	0	1	0
Pit Remover	0	1	0	0	0	0

Table A3: M. Rows are functional tool features, columns are functional fruit features.

we test using 20 testing seeds. For each successful training seeds, we compute the average ME over the test seeds, and report the mean and standard error of the mean (SEM) of the average ME. Given two trained agents A and B, there are $C = 4$ possibles configurations at test time:

1. A is Fruit Player/position 1 and B is Tool Player/position 2
2. A is Fruit Player/position 2 and B is Tool Player/position 1
3. A is Tool Player/position 1 and B is Fruit Player/position 2
4. A is Tool Player/position 2 and B is Fruit Player/position 1

We balance the number of test games in each configuration: we use 3 batches of 100 test games in each configuration, resulting in 12 batches for a total of 1200 test games. The ME value in each configuration c is the average ME over the number of batches in this configuration (3 in our case). We then average the ME in each configuration over the four possible configurations to obtain $ME^{1\to 2}$, $ME^{F\to T}$ and their reverse.

3 Message effect metric

3.1 Causal graph and assumptions

Figure A1 shows the causal graph we consider when computing $\text{ME}_t^{A\to B}$. We write all variables that should be considered at this turn t. Conditioning on c_t^A, s_{t-1}^B, i^B blocks any backdoor paths when we compute the causal influence of m_t^A on c_{t+1}^B, m_{t+1}^B (Pearl et al., 2016). Moreover, the path between m_t^A and c_{t+1}^B, m_{t+1}^B through i^A is blocked at the collider node s_{t+2}^A. Therefore, we ensure there is no confounder when we compute the influence of m_t^A on c_{t+1}^B, m_{t+1}^B. As in Jaques et al. (2018), we have knowledge of the inputs to the model and the distributions of the variables we consider. Therefore we do not need to perform abduction to update probabilities of unobserved exogenous variables that may alter the causal relations in our model (Pearl et al., 2016).

We denote agent B's choice and message pair at turn $t+1$ as $z_{t+1}^B = (c_{t+1}^B, m_{t+1}^B)$. We explain in Section 3 of the main paper that we compare (i) the conditional distribution $p(z_{t+1}^B|m_t^A)$ and (ii) the marginal distribution $p(z_{t+1}^B)$ which does not take m_t^A into account. We intervene on

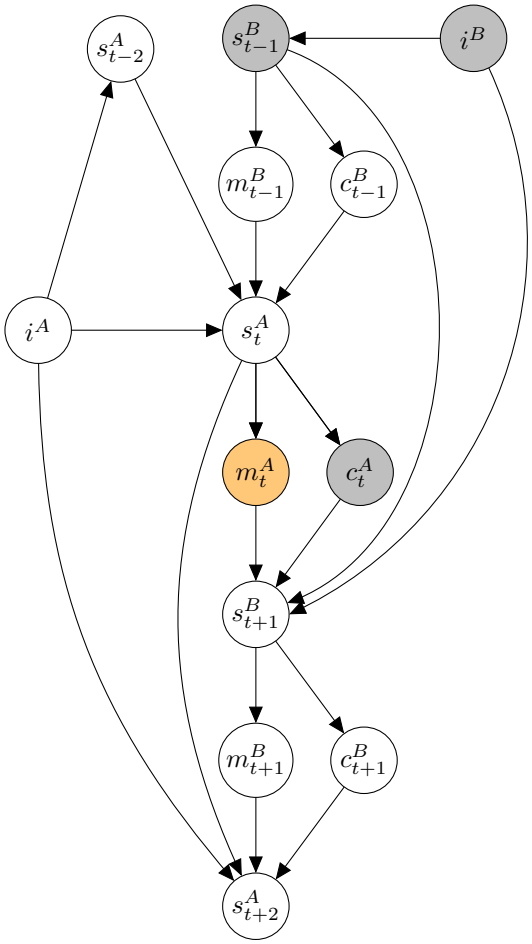

Figure A1: Causal graph considered when we compute $\text{ME}_t^{A\to B}$. The orange node m_t^A is the variable we intervene on. Shaded nodes represent the variables we condition on.

m_t^A, and draw counterfactual messages not from agent A but from another distribution over m_t^A, the *intervention distribution*. We define $\tilde{p}(z_{t+1}^B)$, the marginal computed with counterfactual messages $m_t'^A$, as:

$$\tilde{p}(z_{t+1}^B) = \sum_{m_t^A} p(z_{t+1}^B|m_t'^A)\tilde{p}(m_t'^A). \quad (1)$$

where $\tilde{p}(m_t'^A)$ is the *intervention distribution*. In our experiments we take a uniform intervention distribution. Importantly, $\tilde{p}(m_t'^A)$ is different from the *observational distribution* distribution $p(m_t^A|s_t^A)$ that agent A actually defines over the messages. Contrarily to Bottou et al. (2013) by feeding the counterfactuals messages to agent B, we access $p(z_{t+1}^B|m_t'^A)$ and need not estimate it from empirical data.

3.2 Difference with Mutual Information

Jaques et al. (2018) train agents to have impact on

other agents by maximizing the causal influence of their actions. They show their definition of influence to relate to the Mutual Information (MI) between influencing and influenced agents. Lowe et al. (2019) also define a causal influence metric based on MI. MI computation requires counterfactuals drawn from the influencing agent distribution, and not from an intervention one. In our setting, this means drawing counterfactuals from agent A's distribution $p(m_t^A|s_t^A)$, and not $\tilde{p}(m_t'^A)$, in step 2 of Algorithm 1 (main paper).

There is an issue with employing MI and drawing counterfactuals from the influencing agent's distribution, e.g., $p(m_t^A|s_t^A)$, that is particularly pressing if message distributions are very skewed (as it is the case with our agents below). Consider a simple setting where agents A and B utter a message that has two possible values u and v. A is the influencing agent and B is the influencee. Suppose that the dynamics are such that A almost always says u, and B always replies with the message it received. Most of the exchanges we would sample would be: "A says u, B replies u". The MI estimate would then be very low, and one might erroneously conclude that B is not influenced by A. Indeed when distributions are very peaky as in this example, it would require very large samples to witness rare events, such as "'A says v, B replies v". Lowe et al. (2019) ensure that all possible messages from the influencing agent are considered. This is computationally expensive when the message space is large. Moreover, the resulting MI can still be small as each message's contribution is weighted by its probability under the influencing agent. By using a uniform intervention distribution, we ensure that B in the current example would receive v and therefore reply v in half the exchanges, easily detecting the influence of A on B.

4 Additional results: performance and pragmatics

Table A4 reports a more detailed view of the ME in Table 1 of the main paper. $1T/2F$ denotes games where the Tool Player is in first position and the Fruit Player is in second position, and $1F/2T$ denotes games where the Fruit Player in first position and Tool Player in second. This table shows that in the no-memory, with-communication setting (top right quadrant), the difference between the influence of the Fruit Player on the Tool player and its reverse is greater when the Fruit Player is in position 2. On in-domain data, the Fruit Player has a stronger influence on the Tool Player only when in position 2. This explains the effect $\text{ME}^{2\to1} > \text{ME}^{1\to2}$ we mention in Section 4.1 in the main paper. We also observe that in the no-memory, no-communication setting (top left quadrant), when the Tool Player is in position 1, $ME^{1\to2}$ $1T/2F \approx 0$. This relates to the artifact we describe in the main paper: in that case the Tool Player stops the game at $t = 0$, leaving no room for the Fruit Player to be influenced.

5 Details on the semantics classifier

5.1 Classifier training and hyperparameters

Our classifier consists of an Embedding table of size 50 which maps the agents' discrete utterances to a continuous space, then uses a RNN with a hidden size of 100 to map the entire embedded conversation into a hidden state. The hidden state is then fed to a linear classifier that predicts a score for each class, and the number of classes depends on the prediction task (e.g. 31 classes when the task is to predict the fruit). We consider the setting with communication and with memory. From successful test in-domain conversations, we create train/validation/test partitions for the classifier. We ensure that each fruit is in the train set, and each tool in either of the two positions. We use 20 different seeds for initializing the classifier dataset partitioning into train/validation/test. For each successful training seed, we compute the average accuracy over these 20 test initialization seeds, and report the classifier accuracy mean and standard error of the mean (SEM) over the successful training seeds.

The agents were trained with symmetrical roles and random starting agent, but we generate conversations with fixed roles and positions, so that all conversations follow the same pattern (for example: agent A always starts and agent A is always the Fruit Player).

5.2 Inverted-roles experiment

Table A5 shows the results of the inverted-roles experiment: e.g., we train the classifier on conversations where A is Fruit Player and B is Tool Player, and test on conversations about the same inputs, but where the roles are inverted, that is, B is Fruit Player and A is Tool Player. The performance drops compared to testing on conversations

		No communication		With communication	
	Metric	In	Transfer	In	Transfer
No memory	Av. perf. (%)	84.83 ± 0.09	84.0 ± 0.11	96.9 ± 0.32	94.5 ± 0.37
	$ME^{F \to T}$	$0.133^* \pm 0.01$	$0.14^* \pm 0.01$	$5.0^* \pm 0.39$	$5.0^* \pm 0.36$
	$ME^{T \to F}$	0.05 ± 0.02	0.030 ± 0.01	3.9 ± 0.38	3.3 ± 0.30
	$ME^{1 \to 2}$	0.066 ± 0.00	0.067 ± 0.01	3.9 ± 0.29	3.7 ± 0.26
	$ME^{2 \to 1}$	$0.12^* \pm 0.02$	$0.10^* \pm 0.01$	$5.0^* \pm 0.38$	$4.7^* \pm 0.33$
	$ME^{1 \to 2}$ $1T/2F$	0.000001 ± 0.00	0.000001 ± 0.00	3.7 ± 0.46	3.0 ± 0.34
	$ME^{2 \to 1}$ $1T/2F$	$0.13^* \pm 0.01$	$0.15^* \pm 0.01$	$5.8^* \pm 0.50$	$5.7^* \pm 0.47$
	$ME^{1 \to 2}$ $1F/2T$	0.133 ± 0.01	$0.13^* \pm 0.01$	4.2 ± 0.34	$4.4^* \pm 0.34$
	$ME^{2 \to 1}$ $1F/2T$	0.10 ± 0.03	0.06 ± 0.02	4.2 ± 0.39	3.6 ± 0.29
With memory	Av. perf. (%)	88.5 ± 0.11	87.7 ± 0.16	97.4 ± 0.12	95.3 ± 0.16
	$ME^{F \to T}$	$0.11^* \pm 0.01$	$0.13^* \pm 0.01$	$3.0^* \pm 0.29$	$2.8^* \pm 0.24$
	$ME^{T \to F}$	0.064 ± 0.01	0.071 ± 0.01	1.8 ± 0.22	1.8 ± 0.21
	$ME^{1 \to 2}$	0.085 ± 0.01	0.10 ± 0.01	2.4 ± 0.29	2.3 ± 0.22
	$ME^{2 \to 1}$	0.093 ± 0.01	0.103 ± 0.01	2.4 ± 0.22	2.4 ± 0.21
	$ME^{1 \to 2}$ $1T/2F$	0.063 ± 0.01	0.064 ± 0.01	1.8 ± 0.24	1.8 ± 0.23
	$ME^{2 \to 1}$ $1T/2F$	$0.12^* \pm 0.02$	$0.13^* \pm 0.02$	$2.9^* \pm 0.25$	$2.9^* \pm 0.25$
	$ME^{1 \to 2}$ $1F/2T$	$0.106^* \pm 0.01$	$0.13^* \pm 0.02$	$3.1^* \pm 0.35$	$2.8^* \pm 0.25$
	$ME^{2 \to 1}$ $1F/2T$	0.065 ± 0.01	0.077 ± 0.01	1.8 ± 0.21	1.9 ± 0.20

Table A4: Detailed ME values (compare to Table 1 in main paper). $1T/2F$ denotes games where the Tool Player is in first position and the Fruit Player is in second position, and $1F/2T$ denotes games where the Fruit Player in first position and Tool Player in second.

Utterances	Fruit	Tool 1	Tool 2
Both, A is F	42 ± 2.21	32 ± 2.04	27 ± 1.33
Both, B is F	44 ± 2.00	28 ± 1.58	28 ± 1.69
Both, Train A is F / Test B is F	6.8 ± 0.61	11 ± 1.16	8.8 ± 0.68
Both, Train B is F / Test A is F	5.9 ± 0.53	10 ± 1.05	8.8 ± 0.62
Stats A is F	6.4 ± 0.27	8.9 ± 0.42	8.2 ± 0.38
Stats B is F	6.4 ± 0.15	9.1 ± 0.61	9.0 ± 0.75

Table A5: Semantic classifier % accuracy in inverted-roles setup

where the roles are not inverted. For this experiment, we consider only the conversations that have at least one utterance from each agent (conversation length ≥ 2) in order to remove the potential confounding effect of conversation length.

References

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Çaglar Gülçehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas. 2018. Intrinsic social motivation via causal influence in multi-agent RL. *CoRR*, abs/1810.08647.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. 2019. Measuring emergent communication is tricky. In *Proceedings of AAMAS*, Montreal, Canada. In press.

Judea Pearl, Madelyn Glymour, and Nicholas Jewell. 2016. *Causal Inference in Statistics: A Primer*. John Wiley & Sons.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.