

The Effect of Efficient Messaging and Input Variability on Neural-Agent Iterated Language Learning

Yuchen Lian^{1,2}, Arianna Bisazza³, Tessa Verhoef²

¹Faculty of Electronic and Information Engineering, Xi'an Jiaotong University

²Leiden Institute of Advanced Computer Science, Leiden University

³Center for Language and Cognition, University of Groningen

{y.lian, t.verhoef}@liacs.leidenuniv.nl

a.bisazza@rug.nl

Abstract

Natural languages display a trade-off among different strategies to convey syntactic structure, such as word order or inflection. This trade-off, however, has not appeared in recent simulations of iterated language learning with neural network agents (Chaabouni et al., 2019b). We re-evaluate this result in light of three factors that play an important role in comparable experiments from the Language Evolution field: (i) speaker bias towards efficient messaging, (ii) non systematic input languages, and (iii) learning bottleneck. Our simulations show that neural agents mainly strive to maintain the utterance type distribution observed during learning, instead of developing a more efficient or systematic language.

1 Introduction

The world's languages show immense variety, but linguistic patterns also show important universal tendencies (Greenberg, 1963). It has been argued that these common design features are shaped by human cognitive constraints and pressures during communication and transmission (Kirby et al., 2014). A well-known example of these tendencies is the trade-off between case marking and word order as redundant strategies to encode the role of sentence constituents (Sinnemäki, 2008; Futrell et al., 2015): flexible order typically correlates with the presence of case marking (e.g. in Russian), while fixed order is often observed in languages with little or no case marking (e.g. English).

Researchers interested in language universals and their origins have extensively used agent-based modeling techniques to study the impact of social processes on the emergence of linguistic structures (de Boer, 2006). Besides the horizontal transmission that is often modeled in the referential game setup, the process of iterated learning, where signals are transmitted vertically from generation to generation, has been identified to shape language (Kirby, 2001; Kirby et al., 2014).

Recently, the advent of deep learning based NLP has triggered a renewed interest in agent-based simulations of language emergence. Most existing studies simulate the emergence of language by letting neural network agents play referential games and studying the signals they use (Kottur et al., 2017; Havrylov and Titov, 2017; Lazaridou et al., 2018; Chaabouni et al., 2019a; Dagan et al., 2021). By contrast, Chaabouni et al. (2019b) expose their agents to a pre-defined language, which is then learned and reproduced iteratively by a chain of agents. They analyze how specific properties of the initial languages affect learnability, and further investigate how agent biases affect the evolution across generations. Among others, they studied whether neural agents tend to avoid redundant coding strategies as natural languages do. However, the case-marking/word-order trade-off did not clearly appear in their iterated learning experiments.

In this work, we re-evaluate this finding in light of three factors that play an important role in comparable experiments from the Language Evolution field: (i) speaker bias towards efficient messaging (i Cancho and Solé, 2003), (ii) unpredictable variation in the initial languages (Smith and Wonnacott, 2010; Fedzechkina et al., 2017), (iii) exposure to a limited set of example utterances, known as 'learning bottleneck' (Kirby et al., 2014). We follow the iterated learning setup of Chaabouni et al. (2019b) where neural agents are trained to communicate about trajectories in a simple gridworld, exchanging instructions in miniature languages (Fig. 1).¹

2 Miniature Languages

Word order and case marking are two different mechanisms to convey sentence constituent roles, both widely attested among world languages. Both

¹The original implementation is taken from <https://github.com/facebookresearch/brica>. Our revised code and data are available at https://github.com/Yuchen-Lian/neural_agent_trade-off.

Type	Utterance
Fix+Marker	<i>m1 up 2 m2 left 3 m3 down 1</i>
Fix	<i>up 2 left 3 down 1</i>
Free-order	<i>m1 up 2 m3 down 1 m2 left 3</i>
+Marker	<i>m2 left 3 m1 up 2 m3 down 1</i>
	<i>m3 down 1 m1 up 2 m2 left 3</i>
	...

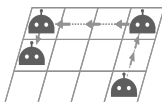


Figure 1: Utterances corresponding to ‘UP UP LEFT LEFT LEFT DOWN’, in three basic languages.

have been shown to serve as equally valuable cues to learn grammatical roles during language learning by children (Slobin and Bever, 1982), and by simple recurrent neural networks (Lupyan and Christiansen, 2002). To model these mechanisms we use simple artificial languages based on Chaabouni et al. (2019b). The meaning space is composed of trajectories defined by random combinations of four oriented actions $\{\text{LEFT}, \text{RIGHT}, \text{UP}, \text{DOWN}\}$. Each utterance or sentence (S) consists of several phrases (P), which in turn are composed of a command (C) and a quantifier (Q):

$$S \rightarrow P_i P_j P_k \dots \quad (1)$$

$$P_i | P_j | P_k | \dots \rightarrow C Q \quad (2)$$

$$C \rightarrow (\text{left} | \text{right} | \text{up} | \text{down}) \quad (3)$$

$$Q \rightarrow (1 | 2 | 3) \quad (4)$$

where *left*, *right*, *up*, *down*, 1, 2, 3 are spoken words which are atomic elements of the language. We consider three basic language types: Fixed-order with marker (redundant), Fixed-order without marker (non-redundant), and Free-order with marker (non-redundant). See examples in Fig. 1.

Fixed-order vs. Free-order This concerns Rule 1: In a fixed-order language, the order of phrases strictly corresponds to the temporal order of instructions in the trajectory.² Free-order languages, instead, allow any permutation of phrases. For instance, there are in total six possible free-order utterances for a 3-phrase trajectory (Fig. 1).

Case Marking In a case-marking language, each phrase is preceded by a temporal marker indicating its role. Thus, Rule 2 changes to: $P_i \rightarrow m_i C Q$ with the marker m_i indicating that CQ is the i^{th} action segment. Note that a fully free-order language is unintelligible without markers.

²This is the ‘forward-iconic’ language of Chaabouni et al. (2019b). We do not consider other fixed orders in this work, as we are mostly interested in the contrast between redundant and non-redundant languages.

3 Neural-Agents Iterated Learning

We strictly follow the iterated learning setup of Chaabouni et al. (2019b) unless explicitly noted.

Agent architecture Agents are implemented as 1-layer attentional Seq2Seq (Sutskever et al., 2014; Bahdanau et al., 2015) LSTM (Hochreiter and Schmidhuber, 1997) networks. Each agent acts as both speaker (receiving trajectories and describing them with utterances) and listener (receiving utterances and trying to induce the corresponding trajectories). The vocabulary contains both actions and words; embeddings of the encoder input and decoder output are tied (Press and Wolf, 2017).

Individual and iterated learning Given trajectory-utterance pairs, agents are trained by teacher forcing (Salakhutdinov, 2014) in both listening and speaking mode, using the early-stopping and optimizer settings of Chaabouni et al. (2019b).³ Iterated learning (Kirby, 2001) is implemented by letting a trained adult agent teach a randomly initialized child agent, and repeating this process for a number of generations. At each generation, two steps are performed: (1) a trained adult agent receives a batch of trajectories and generates utterances by sampling from its own decoder outputs; (2) a randomly initialized child agent is trained on these agent-specific trajectory-utterance pairs. As an exception, the generation-0 agent is directly trained on the corpus generated by a given miniature grammar.

Evaluation In both speaking and listening mode, sequences are generated by greedy decoding and evaluated by sentence-level accuracy. Listener evaluation is standard as the true meaning of an utterance is unique. For speakers in the first generation, instead, we consider all acceptable utterances according to the grammar as candidate targets; for later generations, we take $k = i!$ utterances sampled from the parent’s speaking network as targets (i is the maximum number of phrases per trajectory). Validation for early stopping is performed similarly. These evaluation procedures allow a child agent’s language to deviate from the parent language, even while achieving perfect accuracy.

³To handle 1-to-N trajectory-to-utterance mappings in free-order languages, Chaabouni et al. (2019b) used a modified training loss for the Speaker direction. Empirically, we find that sampling multiple free-order utterances in the initial training corpus leads to very similar results, so we do not use the modified training loss. This allows us to support more complex languages without major changes to the training procedure.

For each experiment, we report speaking accuracy, listening accuracy, as well as average utterance length across generations. To get more insight into language changes, we plot the distribution of utterance types in the adult speaking agents across generations. Specifically, we count how often an utterance belongs to one of the basic language types (*fix*, *fix_marker*, *free*, *free_marker*), and how often markers are dropped for some of the phrases (*fix_drop*, *free_drop*). Utterances that do not fall into any of these categories are labeled as ‘*other*’.

For more technical details on the agents and evaluation, see Appendix A. Example utterances at various generations are shown in Appendix B.

4 Effect of Least-Effort Bias

A bias towards efficient messaging has been proposed as explaining factor for several tendencies observed in natural languages (i Cancho and Solé, 2003; Kanwal et al., 2017; Fedzechkina et al., 2017). Could the lack of such a bias in neural networks explain the survival of redundant languages? To verify this, we design a simple mechanism simulating an agent’s preference to minimize utterance length, based on Chaabouni et al. (2019b)’s framework: To teach the next generation, given a trajectory t , an adult agent generates n (possibly identical) utterances $\{\hat{u}\} = \{\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n\}$ by sampling from its trained network. Instead of modifying the training process, we exploit the diversity occurring in the sampled utterances and hard-code a shorter-sentence selection bias into this adult language generation. As shown in Algorithm 1, the sampling function is called n times to generate n samples. In turn, at each iteration, we ask the adult speaker to generate ℓ sentences and select each time the shortest one. Thus, we can control the bias strength by varying the number of generated samples (ℓ) in each of the n iterations. As ℓ increases, the chances of sampling a shorter sentence increase, resulting in a stronger pressure; $\ell = 1$ means no bias. We expect this least-effort bias will cause the redundant disambiguation mechanism to gradually disappear, and the fixed-order strategy to dominate as that always leads to shorter utterances.

Results Fig. 2 shows the iterated learning results of the Fixed+Marker language with various levels of least-effort bias $\ell = \{1, 3, 5, 8\}$, which represent no pressure, low-, medium- and high-level pressure towards shorter utterances, respectively. The experiment without least-effort pressure

Algorithm 1: Shorter-sentence selection

Input: Trajectory t
Output: n sampled utterances $\{\hat{u}\}$

```

1 for  $j = 1 : n$  do
2   if shorter_selection then
3      $uttrs = \text{Adult.speaker}(t).\text{sample}(\ell)$ 
4      $uttr\_select = uttrs[0]$ 
5     for  $i = 1 : \ell$  do
6        $u = uttrs[i]$ 
7       if  $\text{len}(u) \leq \text{len}(uttr\_select)$  then
8          $uttr\_select = u$ 
9     else
10       $uttr\_select = \text{Adult.speaker}(t).\text{sample}(1)$ 
11     $\{\hat{u}\}.\text{append}(uttr\_select)$ 

```

($\ell = 1$) corresponds to the setup of Chaabouni et al. (2019b), in which the redundant language was found to remain stable across generations.

We find that, while speaking accuracy remains stable (2a), our least-effort pressure leads to a severe drop in listening accuracy (2b) and a dramatic increase of *other* types in the speaking adult agent starting from the fifth generation (2d). Stronger levels of pressure lead to a faster decrease of average utterance length (2c), which was expected. However, manual inspection of the utterances (Appendix B) reveals that agents start dropping entire phrases, thereby losing information, instead of either dropping markers or changing the word order.

5 Effect of Input Language Variability

In the languages of Chaabouni et al. (2019b), markers are either present and fully systematic, or not present at all. If there is no marker example in the initial language, it is unlikely an agent would suddenly invent it. Conversely, a fully systematic use of markers may be perfectly learnable by the agent, unlikely to change or disappear over generations. Unpredictable variation, instead, is a common feature of the artificial languages used in human learning studies. For example Fedzechkina et al. (2017) combine *optional* case marking with either fixed or free word order, while Smith and Wonnacott (2010) use two plural markers with different distributions over nouns. Inspired by this, we experiment with unpredictable variations in the use of markers, namely: (i) variability *among* utterances, where each utterance is consistent with one of the basic language types chosen at random, and (ii) variability *within* utterances, where the use of markers is also unpredictable within a single utterance.

5.1 Variability Among Utterances

For every trajectory in the initial training set, two utterances are generated for each of the three basic

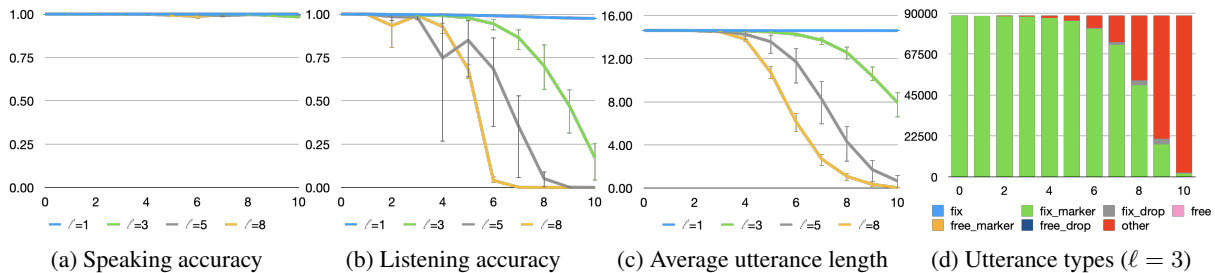


Figure 2: Iterated learning of the Fixed+Marker language with least-effort pressure of varying strength (ℓ), over 10 generations. Results in (a,b,c) are averaged over three random-seed initializations. (d) shows the distribution of utterance types of the speaking adult agents with $\ell = 3$ (one seed only).

language types: (i) Fixed-order+Marker, (ii) Fixed-order without markers and (iii) Free-order+Marker. Our goal here is to find out whether the agents will tend to prefer any of the three language types over generations, according to their inherent biases.

Results Results are shown in Fig. 3 (blue lines). The overall high speaking accuracy suggests that the agents can learn to imitate their parents’ language very well. We observe a slow, but steady, loss in listening accuracy, which we attribute to the random sampling errors from the parent speaker and the natural presence of errors in the neural network learning process. Besides a steady increase of uncategoryzable utterances (*other*) in Fig. 3d, the distribution of the three language types remains relatively stable even after 20 generations. We looked for sentences where only some of the markers are dropped (*free_drop*/*fix_drop*) but found almost none. See examples in Appendix B. These results show that mixing language types in the initial training set is not sufficient to induce the loss of redundant encoding with agent learners, as opposed to human learners (Fedzechkina et al., 2017).

Results with Least-Effort Bias We also study the combination of the two factors (Mix+pressure) using medium pressure ($\ell = 3$). This setup (Fig. 3, green lines) leads to a more efficient language during the first five generations, as shown by the initially stable speaking and listening accuracy and a decrease of average length. This phase corresponds to a proliferation of fixed-order no-marker sentences and the disappearance of the other language types (3e). In only two generations, this language has reached the shortest possible overall length while remaining intelligible. After a while, however, child agents start to receive shorter but incorrect utterances, resulting in a drop of listening accuracy and, finally, an unintelligible language.

5.2 Variability Within Utterances

We design a more unpredictable language where each phrase marker is randomly dropped according to a given probability (10%). Half of the utterances are fixed- and half are free-order. See examples in Appendix B. This language is closely inspired by those of Fedzechkina et al. (2017). We expect agents will either stop using markers completely over generations, or use them more consistently.

Results Despite the relatively small probability of dropping a marker, speaking and listening accuracies drop rapidly (3a and 3b, grey lines) and *other* utterances become dominant (Fig. 3f). This language becomes unintelligible before regularization is observed, again challenging our expectations.

6 Effect of Learning Bottleneck

Although real languages support the production of enormous sets of utterances, human learners master them by exposure to a limited number of examples. This *learning bottleneck* acts as a pressure forcing languages to regularize during cultural transmission (Smith et al., 2003; Brighton et al., 2005). Human-based experiments and computational simulations have found that this pressure can lead to increased structure in emerging language systems (Kirby et al., 2014). We apply a learning bottleneck to our mixed language experiment (Sect. 5.1) by randomly sub-sampling, at each iteration, 50% of the data used to train the next generation. Evaluation and training details are the same as in Sect. 5.1.

Results We find that training data sub-sampling leads to a slightly steeper drop in listening accuracy (yellow vs. blue line in Fig. 3b), but the respective distributions of utterance types (Fig. 3g vs. 3d) remain very similar, suggesting the learning bottle-

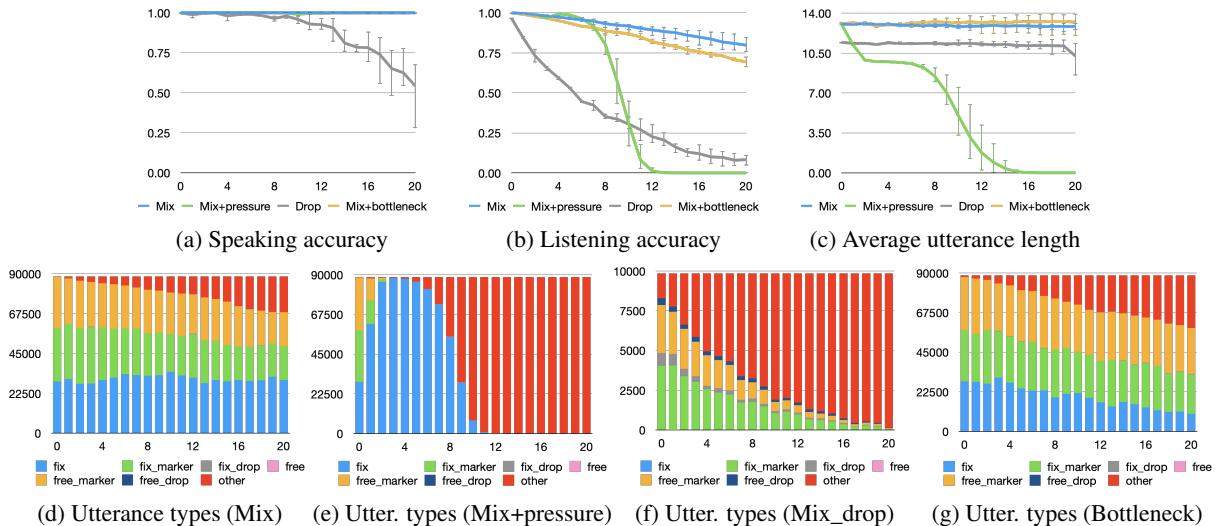


Figure 3: Mixed language learning without and with least-effort pressure, drop-marker language without least-effort pressure, and mixed language with learning bottleneck. Results in (a,b,c) are averaged over three random seeds. (d,e,f,g) show the respective distributions of utterance types in the speaking adult agents (one seed only).⁴

neck does not result in a more structured language.

7 Discussion and Conclusions

Neural-agent iterating learning is a promising framework to study the impact of social processes on the emergence of patterns and language universals, like the word order/case marking trade-off. However, previous work with LSTM-based agents (Chaabouni et al., 2019b) has failed to replicate this human-like pattern. We re-evaluated this finding by (i) hard-coding a least-effort bias into our agents, (ii) designing less systematic input languages, and (iii) introducing a learning bottleneck. In all cases, our agents proved to be accurate learners, but the patterns of language change over generations did not match our expectations. Specifically, least-effort bias (§4) and highly unpredictable input language (§5.2) led to communication failure, whereas moderate input language variability (§5.1) and learning bottleneck (§6) led to a stable language distribution, confirming previous observations on the survival of redundant coding strategies in neural-agent iterated learning (Chaabouni et al., 2019b). Only combining least-effort bias with moderate language variability (§5.1) led to a temporary optimization of the language, but that was again followed by communication failure.

In real language use, the pressure to reduce effort is balanced with communicative needs (Kirby et al., 2015; Regier et al., 2015) and does normally not

lead to severe language degradation. Future work should design subtler least-effort biases, for instance considering efficiency in terms of grammatical structure and cognitive effort. Moreover, our results with non fully systematic languages show that agents strive to preserve the initial utterance type distribution. In human learning, this behavior is called probability matching and is affected by task complexity: more difficult tasks tend to regularization or over-matching (Ferdinand et al., 2019; Kam and Newport, 2009), where the more frequent variant is chosen more often than it appeared in the input. Over many generations, even a slight over-matching can lead to the emergence of linguistic regularities (Smith and Wonnacott, 2010) like the word order/case marking trade-off observed in human learners by Fedzechkina et al. (2017). We conclude that the current neural-agent iterated learning framework is not yet ready to simulate language evolution processes in a human-like way. More natural cognitive biases supporting efficiency should be modeled, while the speaker objective should be balanced with a measure of communicative success, such as the likelihood of a message to be understood by the listener (Goodman and Frank, 2016; Scontras et al., 2021).

Acknowledgements

Lian was partly funded by the China Scholarship Council (CSC 201906280463). Bisazza was partly funded by the Netherlands Organization for Scientific Research (NWO project 639021646).

⁴Speaking accuracies of Mix+pressure, Mix+bottleneck, Drop in 3a are hidden behind the accuracy of Mix (blue line).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Henry Brighton, Kenny Smith, and Simon Kirby. 2005. Language as an evolutionary system. *Physics of Life Reviews*, 2(3):177–226.
- Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019a. [Anti-efficient encoding in emergent communication](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6290–6300.
- Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. 2019b. [Word-order biases in deep-agent emergent communication](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5166–5175, Florence, Italy. Association for Computational Linguistics.
- Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2021. [Co-evolution of language and agents in referential games](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2993–3004, Online. Association for Computational Linguistics.
- Bart de Boer. 2006. Computer modelling as a tool for understanding language evolution. In *Evolutionary epistemology, language and culture*, pages 381–406. Springer.
- Maryia Fedzechkina, Elissa L Newport, and T Florian Jaeger. 2017. Balancing effort and information transmission during language acquisition: Evidence from word order and case marking. *Cognitive science*, 41(2):416–446.
- Vanessa Ferdinand, Simon Kirby, and Kenny Smith. 2019. The cognitive roots of regularization in language. *Cognition*, 184:53–68.
- Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. [Quantifying word order freedom in dependency corpora](#). In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden. Uppsala University, Uppsala, Sweden.
- Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829.
- Joseph Harold Greenberg. 1963. Universals of language.
- Serhii Havrylov and Ivan Titov. 2017. [Emergence of language with multi-agent games: Learning to communicate with sequences of symbols](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2149–2159.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Ramon Ferrer i Cancho and Ricard V Solé. 2003. Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences*, 100(3):788–791.
- Carla L Hudson Kam and Elissa L Newport. 2009. Getting it right by getting it wrong: When learners change languages. *Cognitive psychology*, 59(1):30–66.
- Jasmeen Kanwal, Kenny Smith, Jennifer Culbertson, and Simon Kirby. 2017. Zipf’s law of abbreviation and the principle of least effort: Language users optimise a miniature lexicon for efficient communication. *Cognition*, 165:45–52.
- Simon Kirby. 2001. Spontaneous evolution of linguistic structure—an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110.
- Simon Kirby, Tom Griffiths, and Kenny Smith. 2014. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. [Emergence of linguistic communication from referential games with symbolic and pixel input](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Gary Lupyan and Morten H. Christiansen. 2002. Case, word order, and language learnability: Insights from connectionist modeling. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*.

- Ofir Press and Lior Wolf. 2017. [Using the output embedding to improve language models](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.
- Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. [On the convergence of adam and beyond](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Terry Regier, Charles Kemp, and Paul Kay. 2015. Word meanings across languages support efficient communication. *The handbook of language emergence*, 87:237.
- Ruslan Salakhutdinov. 2014. [Deep learning](#). In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, page 1973. ACM.
- Gregory Scontras, Michael Henry Tessler, and Michael Franke. 2021. [A practical introduction to the rational speech act modeling framework](#). *ArXiv preprint*, abs/2105.09867.
- Kaius Sinnemäki. 2008. Complexity trade-offs in core argument marking. *Language complexity: Typology, contact, change*, 67:88.
- Dan I. Slobin and Thomas G. Bever. 1982. Children use canonical sentence schemas: A crosslinguistic study of word order and inflections. *Cognition*, 12(3):229–265.
- Kenny Smith, Simon Kirby, and Henry Brighton. 2003. Iterated learning: A framework for the emergence of language. *Artificial life*, 9(4):371–386.
- Kenny Smith and Elizabeth Wonnacott. 2010. Eliminating unpredictable variation through iterated learning. *Cognition*, 116(3):444–449.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

A Training details

Following [Chaabouni et al. \(2019b\)](#) we limit the maximum number of segments per trajectory, i , to 5 and at most 3 steps per phrase, resulting in a total of 89k possible trajectories. Subsequently, the number of candidate target utterances during evaluation is set to $k = i! = 120$. As an exception, for the drop-marker language (Section 5.2) we limit i to 4 instead of 5 due to the computational cost of enumerating all correct utterances for a trajectory in this language during validation (accordingly, k is reduced to 24). The trajectory-utterance pairs are randomly split into training, validation and test sets with a proportion of 80%, 10% and 10% respectively.

We fix the hidden layer size (20) and batch size (16) for all experiments. Similar to [Chaabouni et al. \(2019b\)](#), we use the Amsgrad optimizer ([Reddi et al., 2018](#)). For each generation, the maximum number of training epochs is set to 100 and we stop the training if both speaking and listening accuracy on development set have no improvement over 5 epochs. To ensure the reliability of our results, we repeat each experiment with 3 different random seeds and observe trends over 20 generations (unless trends are already very clear after 10, as in Fig. 2).

B Example utterances at various generations

We let each agent generate six utterances corresponding to the trajectory ‘RIGHT UP UP DOWN RIGHT RIGHT RIGHT’. As some of these utterances are identical, we remove the duplicates and only list unique ones in Tab. 3.

	Fix+Marker with pressure ($\ell = 3$)	Mix	Mix with pressure ($\ell = 3$)
Input	M1 right 1 M2 up 2 M3 down 1 M4 right 3	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up 2 M4 right 3 M3 down 1 M3 down 1 M1 right 1 M4 right 3 M2 up 2	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M2 up 2 M4 right 3 M3 down 1 M1 right 1 M4 right 3 M2 up 2 M3 down 1 M1 right 1
Iter_0	M1 right 1 M2 up 2 M3 down 1 M4 right 3	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up 2 M4 right 3 M3 down 1	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up 2 M4 right 3 M3 down 1
Iter_1	M1 right 1 M2 up 2 M3 down 1 M4 right 3	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M2 up 2 M1 right 1 M4 right 3 M3 down 1 M3 down 1 M2 up 2 M4 right 3 M1 right 1	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M3 down 1 M2 up 2 M4 right 3 M1 right 1
Iter_5	M1 right 1 M2 up 2 M3 down 1 M4 right 3 M5 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up M3 down 1 M4 right 3 M5 3	M3 down 1 M4 right 3 M2 up 2 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 right 1 up 2 down 1 right 3 M1 right 1 M4 right 3 M3 down 1 M2 up 2 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M4 right 3 M3 down 1 M1 right 1 M2 up 2	right 1 up 2 down 1 right 3
Iter_10	M1 right 1 M2 up 2 M3 down 1 M1 right 1 M2 up 2 M1 right 1	right 1 up 2 down 1 right 3 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M1 right 1 M3 down 1 M4 right 3 M2 up 2 M1 right 1 M3 down 1 M4 right 3 M2 up 2 right 1 up 2 down 1 right 3	right 1 up 2 down 1 right 3 right 1 up 2 right 1

	Mix_drop	Mix with learning bottleneck
Input	M1 right 1 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up 2 down 1 M4 right 3 M2 up 2 right 1 M4 right 3 M3 down 1 down 1 M1 right 1 up 2 M4 right 3 M1 right 1 M4 right 3 M2 up 2 M3 down 1	right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M3 down 1 M1 right 1 M2 up 2 M4 right 3 M4 right 3 M2 up 2 M3 down 1 M1 right 1
Iter_0	right 3 M2 up 2 M3 down 1 M4 right 3 M1 right 1 up 1 M3 down 1 M2 up 2 M4 right 3 M1 right 1 M4 right 3 M3 down 1 M2 up 2 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M3 down 1 M1 right 1 M4 right 3 M2 up 2	right 1 up 2 down 1 right 3 M3 down 1 M1 right 1 M2 up 2 M4 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3
Iter_1	M3 down 1 M1 right 1 M2 up 2 M4 right 3 M2 up 2 M4 right 3 M1 right 1 M3 down 1 M2 up 2 M1 right 1 M3 down 1 M4 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3	M3 down 1 M4 right 3 M2 up 2 M1 right 1 M4 right 3 M3 down 1 M2 up 2 M1 right 1 M4 right 3 M3 down 1 M1 right 1 M2 up 2 right 1 up 2 down 1 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3
Iter_5	M3 down 1 right 3 M1 right 1 M2 up 2 M2 up 2 M1 right 1 M3 down 1 M4 right 3 right 2 M2 up 2 M3 down 1 M4 right 3 M1 right 1 M2 up 3 M3 down 1 M4 right 3 M1 right 1 M2 up 2 M3 down 1 M4 right 3 M3 down 1 right 3 M4 right 3 M1 right 1	M2 up 2 M4 right 3 M1 right 1 M5 right 3 M2 up 2 M3 down 1 M4 right 3 M1 right 1 right 1 up 2 down 1 right 3 M1 right 1 M3 down 1 M4 right 3 M2 up 2
Iter_10	M1 right 1 M3 down 1 M4 right 3 M1 right 1 M4 right 3 down 1 M1 right 1 M2 down 1 down 1 M4 right 3 M2 up 2 M4 right 3 M1 right 2 M2 up 3 M3 down 1 M4 right 3 M3 down 1 M1 right 2 M4 right 3 M2 up 1 M1 right 1 M3 down 2 M4 right 3 M3 down 1	M1 right 1 M2 up 2 M3 down 1 M4 right 3

Table 3: Utterances sampled from the agents’ speaking network given the trajectory ‘**RIGHT UP UP DOWN RIGHT RIGHT RIGHT**’ in Fix+Marker language learning with pressure (§4), Mix language learning without and with pressure (§5.1), Mix_drop language learning (§5.2) and Mix language with learning bottleneck (§6). For each experiment and each generation, we show six randomly sampled utterances (duplicates are omitted for clarity).