Emergence of linguistic universals in neural agents via artificial language learning and communication

Lian, Y.

Chapter 2

Neural-Agent Iterated Language Learning: Three Missing Factors

Natural languages commonly display a trade-off among different strategies to convey constituent roles. A similar trade-off, however, has not been observed in recent simulations of iterated language learning with neural network based agents (Chaabouni et al., 2019b). In this chapter, we investigate whether introducing some essential human cognitive biases and common ALL design principles inspired by human experiments can lead to the emergence of a word-order/case-marking trade-off in an existing supervised neural-agent iterated language learning framework (Chaabouni et al., 2019b), which failed to find such a trade-off previously. Concretely, we ask:

RQ-A Can the introduction of more realistic simulation factors lead to the emergence of a word-order/case-marking trade-off in neural-agent iterated language learning?

Specifically, we re-evaluate Chaabouni et al. (2019b)'s finding in light of three factors known to play an important role in comparable experiments and simulations from the Language Evolution field. We first introduce a least-effort bias by hard-coding a short utterance selection algorithm to model the speaker bias towards efficient messaging. Then we introduce two levels of variability into

the input language – originally fully systematic in Chaabouni et al. (2019b) – to better simulate the unpredictable variation commonly present in human ALL experiments. Finally, we simulate a learning bottleneck – a key pressure driving language regularization (Smith et al., 2003; Brighton et al., 2005; Kirby et al., 2014) – under the variable input languages learning setup. 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.

Chapter adapted from:
Yuchen Lian, Arianna Bisazza, and Tessa Verhoef. 2021. The effect of efficient messaging and input variability on neural-agent iterated language learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10121–10129. Association for Computational Linguistics

2.1 Introduction

The world's languages show immense variety, but important universal tendencies in linguistic patterns have also been identified (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), such as the preference for efficient messaging. An important and well-known example of such universal 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, Tamil, Turkish) and, vice versa, fixed order is observed in languages with little or no case marking (e.g. in English or Chinese).

Researchers interested in the origins of human language and language universals 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 and language evolution. Most of the existing studies simulate the emergence of language by letting neural network agents play referential games and studying the signals that are used by these agents (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. Using this framework, they analyzed how specific properties of the initial languages affect learnability, and further investigated whether and how languages evolve across generations according to the agents' own biases. Among others, they studied whether neural agents tend to avoid redundant coding strategies as natural languages do. However, the word-order/case-marking trade-off did not clearly appear in their iterated learning experiments, as redundant languages (fixed-order and case marking) were found to survive across multiple generations.

Language Type	**Utterance**
Fix-order+Marker	*m1 up 2 m2 left 3 m3 down 1*
Fix-order	*up 2 left 3 down 1*
Free-order +Marker	*m1 up 2 m2 left 3 m3 down 1* *m1 up 2 m3 down 1 m2 left 3* *m2 left 3 m1 up 2 m3 down 1* *m2 left 3 m3 down 1 m1 up 2* *m3 down 1 m1 up 2 m2 left 3* *m3 down 1 m2 left 3 m1 up 2*

Table 2.1: Utterances corresponding to the trajectory 'UP UP LEFT LEFT LEFT DOWN', in three basic languages.

In this work, we re-evaluate this finding in light of three factors known to play

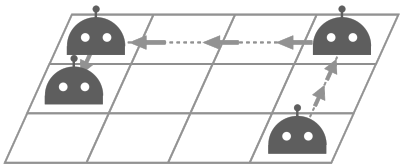


Figure 2.1: Schematic diagram of the trajectory 'UP UP LEFT LEFT LEFT DOWN'

an important role in comparable experiments and simulations in the Language Evolution field, namely: (i) the speaker bias towards efficient messaging (i Cancho and Solé, 2003), (ii) the variable and unpredictable nature of the initial languages (Smith and Wonnacott, 2010; Fedzechkina et al., 2017), and (iii) the exposure of learners to a relatively small set of example utterances, also known as 'learning bottleneck' (Kirby et al., 2014).

We follow the iterated learning setup of Chaabouni et al. (2019b) where neural network agents are trained to communicate about trajectories in a simple gridworld, giving and receiving instructions in miniature languages (Table 2.1 and Figure 2.1).[1]

2.2 Miniature Languages

Word order and case marking are two different mechanisms to convey sentence constituent roles, both widely attested among world languages. In fact, cross-linguistic studies have revealed that children are equally prepared to acquire both fixed-order and inflectional languages (Slobin and Bever, 1982).

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 {LEFT, RIGHT, UP, DOWN}. Each

[1]The original framework 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.

utterance or sentence (S) consists of several phrases (P), which in turn are composed of a command (C) and a quantifier (Q). Below is the basic grammar of this miniature language:

$$S \rightarrow P_i P_j P_k ... \tag{2.1}$$

$$P_i | P_j | P_k | ... \rightarrow CQ \tag{2.2}$$

$$C \rightarrow (left | right | up | down) \tag{2.3}$$

$$Q \rightarrow (1|2|3) \tag{2.4}$$

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 Table 2.1. Below we describe these language variants in more detail.

2.2.1 Fixed-order vs. Free-order

This concerns Rule 2.1 of the grammar: In a fixed-order language, the order of phrases is fixed and corresponds to the temporal order of instructions in the trajectory.[2] Free-order languages, instead, allow any permutation of phrases. For instance, the example in Figure 2.1 has three phrases, corresponding to six possible free-order utterances.

2.2.2 Case Marking

In a case-marking language, each phrase is preceded by a temporal marker indicating its role. Thus, Rule 2.2 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 free-order language without markers would be unintelligible, as the correct order of instructions cannot be conveyed.

[2] This 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.

2.3 Neural-Agents Iterated Learning

We strictly follow the iterated learning setup of Chaabouni et al. (2019b) except when explicitly noted. Below, we provide a short explanation of this framework.

2.3.1 Agent architecture

Agents are trained to communicate about trajectories, and are implemented as one-layer attention-enhanced Seq2Seq (Sutskever et al., 2014; Bahdanau et al., 2015) LSTM (Hochreiter and Schmidhuber, 1997) networks. Each agent acts as both speaker and listener: As a speaker, it takes trajectories as input and expresses them using utterances. As a listener, it receives utterances and try to induce the corresponding trajectories. To jointly train an agent to speak and listen, input and output vocabularies both contain all possible actions and words. Furthermore, embeddings of the encoder input and decoder output are tied (Press and Wolf, 2017).

2.3.2 Individual and iterated learning

Given trajectory-utterance pairs, agents are trained by teacher forcing (Goodfellow et al., 2016) in both listening and speaking mode, using the same early-stopping and optimizer settings as in Chaabouni et al. (2019b). In order to handle one-to-many 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 makes it possible to support more complex languages without major changes to the training procedure.

Iterated learning (Kirby, 2001) is achieved by letting a trained adult agent teach a randomly initialized child agent, and repeating this process for a number of iterations (i.e. 'generations'). Specifically, at each generation, two steps are performed: First, a trained adult agent receives a batch of trajectories and generate its own utterances by sampling from its decoder outputs. Next, a randomly initialized child agent is trained by these agent-specific trajectory-

utterance pairs as training data. One exception is the data used to train the generation-0 agent. As there is no ancestor for this first agent, it directly learns from the training corpus generated by the given miniature language grammar.

2.3.3 Evaluation

In all experiments, agents are evaluated by sentence-level accuracy. During each evaluation, we first ask the speaking or listening agent to generate an output sequence by selecting the symbol with highest probability at each time step (greedy decoding). The listener evaluation is similar to that of standard Seq2Seq models as the true meaning of an utterance, i.e. the corresponding trajectory, is unique. For speakers, instead, multiple utterances may be acceptable for a given trajectory, according to the language type. Therefore, when evaluating the very first-generation speaker, we consider all correct utterances according to the language grammar. When evaluating speakers in later generations, we sample k utterances from the parent's speaking network and consider those as correct. k is set to $i!$ where i is the maximum number of phrases per trajectory. Thus, speaker accuracy reflects the extent to which a child agent's language departs from that of its parent. Validation for early stopping is performed similarly to this evaluation procedure.

These evaluation procedures allow a child agent's language to deviate from the parent language according to its inherent biases, even while achieving perfect accuracy. With our experiments, we study whether these patterns of language change result in more human-like artificial languages.

For each experiment, we report speaking accuracy, listening accuracy, as well as average utterance length across generations. To get more insight into how the language is changing, we also analyze the utterances generated by the adult speaking agent at each generation. Specifically, we count how often an utterance belongs to one of the basic language types (*fix, fix_marker, free, free_marker*), or 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'*. The distribution of such utterance types across

generations is plotted for each experiment. Example utterances at various generations are provided in Table 2.4, where 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.

2.3.4 Training details

Following Chaabouni et al. (2019b) we limit the number of segments per trajectory to 5 and at most 3 steps per phrase, resulting in a total of 89k possible trajectories and $k = 120$. As an exception, for the drop-marker language (Section 2.5.2) we limit the number of phrases 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 Figure 2.2).

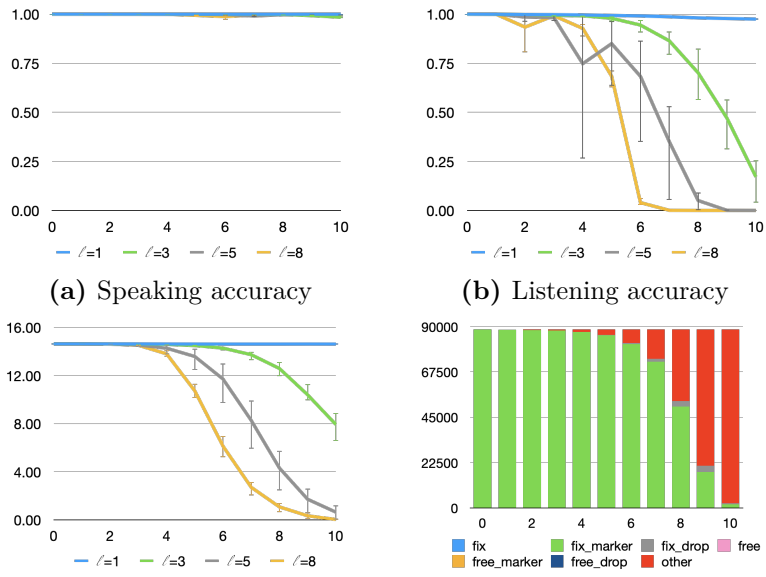
2.4 Effect of Least-Effort Bias

A bias towards efficient messaging, or least-effort bias, 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 least-effort bias in neural networks explain the survival of redundant languages across generations? To verify this, we design a simple mechanism to simulate an agent's preference to minimize utterance length, based on Chaabouni et al. (2019b)'s framework.

Algorithm 1 Shorter-sentence selection

Input: Trajectory t
Output: n sampled utterances $\{\hat{u}\}$
for $j = 1 : n$ **do**
  if *shorter_selection* **then**
    $uttrs$ = Adult.speaker(t).sample(ℓ)
    $uttr\_select = uttrs[0]$
    min_length = len($uttr\_select$)
    for $i = 1 : \ell$ **do**
      $u = uttrs[i]$
      if $len(u) \leq min\_length$ **then**
        $uttr\_select = u$
        min_length = len(u)
      end
    end
  else
    $uttr\_select$ = Adult.speaker(t).sample(1)
  end
  $\{\hat{u}\}$.append($uttr\_select$)
end

At each generation of the iterated learning process, a sample of the parent language is required to train the next generation agent. Specifically, given a trajectory **t**, an adult agent generates n (possibly identical) utterances $\{\hat{u}\} = \{\hat{u_1}, \hat{u_2}, ...\hat{u_n}\}$ by sampling from its trained speaking network. Instead of modifying the training procedure, we take advantage of the diversity occurring among 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 the 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 pressure 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. When $\ell = 1$, there is no pressure and the whole process is equivalent to that of Chaabouni et al. (2019b).



(a) Speaking accuracy **(b)** Listening accuracy

(c) Average utterance length **(d)** Utterance types ($\ell = 3$)

Figure 2.2: 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).

We expect this least-effort bias will cause the redundant disambiguation mechanism to gradually disappear across generations. More specifically, we expect the fixed-order strategy to become dominant as that always leads to shorter utterances.

Results

Figure 2.2 shows the iterated learning results of the Fixed+Marker language with various levels of least-effort bias, namely $\ell = \{1, 3, 5, 8\}$, which represent no pressure, low- , medium- and high-level pressure towards shorter utterances, respectively. The experiment without least-effort pressure ($\ell = 1$) 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 (Figure 2.2a), our least-

effort pressure leads to a severe drop in listening accuracy (Figure 2.2b) and a dramatic increase of uncategorizable ('*other*') utterances in the speaking adult agent starting from the fifth generation (Figure 2.2d). Stronger levels of pressure lead to a faster decrease of average utterance length (Figure 2.2c), which was expected. However, manual inspection of the utterances (see examples in Table 2.4 at the end of this chapter) reveals that the agents start dropping entire phrases, thereby losing information, instead of either dropping markers or changing the word order.

2.5 Effect of Input Language Variability

Besides the lack of efficient messaging pressure, we noticed another possible reason why a trade-off did not appear in Chaabouni et al. (2019b): In their proposed languages, 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, and therefore unlikely to change or disappear over generations.

By contrast, in artificial language learning studies with human participants, unpredictable variation is one of the common features for designing the languages. For example, both languages used by Fedzechkina et al. (2017) contain *optional* case marking in combination with either fixed or free word order, while the languages of Smith and Wonnacott (2010) have two plural markers with different distributions over all nouns.

Inspired by this body of work, we modify our initial languages by introducing unpredictable variation in the use of markers. Specifically, we consider two kinds of variability: (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 the single utterance.

Mix	Mix_drop
m1 up 2 m2 left 3 m3 down 1	*m1 up 2 left 3 down 1*
m1 up 2 m2 left 3 m3 down 1	*m1 up 2 m2 left 3 m3 down 1*
up 2 left 3 down 1	*up 2 m2 left 3 m3 down 1*
up 2 left 3 down 1	*m2 left 3 m1 up 2 down 1*
m2 left 3 m1 up 2 m3 down 1	*down 1 m2 left 3 m1 up 2*
m3 down 1 m2 left 3 m1 up 2	*left 3 down 1 up 2*

Table 2.3: Example utterances corresponding to 'UP UP LEFT LEFT LEFT DOWN' in the Mix and Mix_drop language.

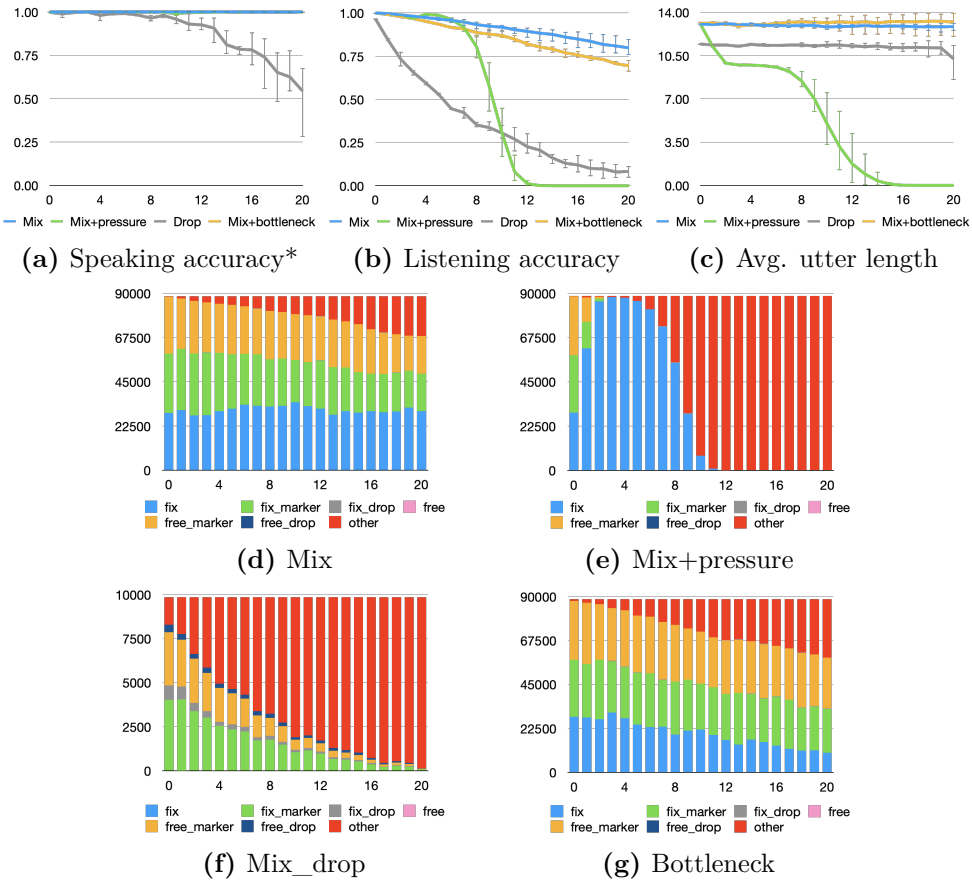
2.5.1 Variability Among Utterances

We design a mixed language containing utterances from the three basic language types, as shown in Table 2.3. Specifically, for every trajectory in the initial training set, an equal number of utterances (two) is generated for: (i) the redundant Fixed-order+Marker, (ii) Fixed-order without markers and (iii) Free-order+Marker. This means that the first child agent will be exposed to case marking 2/3 of the times, and to fixed-order 2/3 of the times. 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

The results for this input language (called 'Mix') are shown in Figures 2.3a, 2.3b and 2.3c (blue lines). The distribution of utterance types is shown in 2.3d. 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 uncategorizable utterances (*other*) in Fig. 2.3d, 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 also example utterances in Table 2.4.

These results show that presenting a mix of the three language types in the initial training set is not sufficient to induce the loss of redundant encoding predicted by efficient coding theories (i Cancho and Solé, 2003; Kanwal et al., 2017).



(a) Speaking accuracy* **(b)** Listening accuracy **(c)** Avg. utter length

(d) Mix **(e)** Mix+pressure

(f) Mix_drop **(g)** Bottleneck

Figure 2.3: Iterated learning of the mixed language 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). * The speaking accuracies for Mix+pressure, Drop and Mix+bottleneck in (a) are not visible because they are very similar to the accuracy of Mix (blue line).

Results with Least-Effort Bias

According to our experiments so far, neither a hard-coded least-effort bias nor the variability among utterances yield the expected patterns of language change. We then experiment with the combination of this two factors (Mix + pressure) using a medium-level pressure ($\ell = 3$). Results are shown in Figure 2.3.

Indeed, this setup leads to a more efficient language in the course of the first five generations, as shown by the initially stable speaking and listening accuracy and a fast decrease of average length (green lines in 2.3a, 2.3b, 2.3c, respectively). This phase corresponds to a rapid increase in the proportion of fixed-order no-marker sentences and the disappearance of the other two types of languages (2.3e). Already by the second generation, this language has reached the shortest possible overall length while serving its communication needs. After a few stable generations, however, child agents start to be exposed to shorter but incorrect utterances, resulting in a rapid drop of listening accuracy and, eventually, to a non-intelligible language.

2.5.2 Variability Within Utterances

While the language in Sect. 2.5.1 is a mix of three language types, each utterance consistently uses only one strategy. To introduce more unpredictable variation, we design another mixed language where the case marker of each phrase is randomly dropped according to a given probability (10%). See examples in Table 2.3 (Mix_drop language). Half of the utterances are fixed- and half are free-order. This language type is closely inspired by those used by Fedzechkina et al. (2017). We expect the agents will either drop the use of markers completely over generations, or start to use them more consistently.

Results

Despite the relatively small probability of dropping a marker, speaking and listening accuracies are heavily affected (grey lines in 2.3a and 2.3b). Average length is overall stable (2.3c). As shown by the fast increase of *other* utterance

types (Fig. 2.3f), this language becomes unintelligible before any regularization can be observed, once again challenging our expectations.

2.6 Effect of Learning Bottleneck

Even though real languages support the production of enormously large sets of utterances, human learners can master them after being exposed to only a limited number of example utterances. This poverty of the stimuli is referred to as the learning bottleneck, which acts as a pressure forcing language to generalize during cultural transmission (Smith et al., 2003; Brighton et al., 2005). Human-based experiments and computational simulations have found that the learning bottleneck can lead to increased structure in emerging language systems (Kirby et al., 2014), making it a key factor in the evolution of language. We introduce such a learning bottleneck in our mixed language experiment (Sect. 2.5.1) by randomly sub-sampling, at each iteration, only 50% of the data used to train the next generation. Evaluation and other training details are the same as in Sect. 2.5.1.

Results

Comparing the yellow line to the blue line in Fig. 2.3b, we see that training data sub-sampling leads to a slightly steeper drop in listening accuracy. However, the respective distributions of utterance types across generations (Fig. 2.3g vs. 2.3d) are very similar, which means this learning bottleneck does not result in a more structured language.

2.7 Discussion and Conclusions

Neural-agent iterating learning is a promising framework to study the impact of social processes on the emergence of linguistic structure and language universals, such as the trade-off between case marking and word order as redundant strategies to encode constituent roles. 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 more realistic input languages with different levels of variability, 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 (§2.4) and highly unpredictable input language (§2.5.2) lead to a collapse of the communication system, whereas moderate input language variability (§2.5.1) and learning bottleneck (§2.6) lead to a stable language distribution, confirming previous observations on the survival of redundant coding strategies in neural-agent iterated learning (Chaabouni et al., 2019b). Among all our experiments, only the one where hard-coded least-effort bias was combined with moderate language variability (§2.5.1) led to a temporary optimization of the language in terms of both efficiency and communicative success. However, after a few stable generations, shorter but incorrect utterances became dominant causing the communication system to collapse.

In real language use, a pressure for reducing effort is balanced with communicative needs (Kirby et al., 2015; Regier et al., 2015), and this would normally not lead to a severe language degradation. Future work should therefore design more subtle least-effort biases, for instance by considering efficiency at the level of grammatical structures and cognitive effort besides shallow properties of the language production like utterance length.

Additionally, our results with non fully systematic input languages show that neural agents strive to preserve the initial distribution of utterance types. In human learning, this is called probability matching: reproducing input variability in a way that the distribution of each type is matched. This behavior is affected by task complexity, since more difficult tasks tend to lead to regularization or over-matching behavior instead (Kam and Newport, 2009; Ferdinand et al., 2019), where the more frequent variant is chosen more often than it appeared in the input. Even a very small amount of over-matching can, over multiple generations, lead to significant changes in structure and to the emergence of linguistic regularities (Smith and Wonnacott, 2010; Fedzechkina et al., 2017). In artificial language experiments with human learners, this even led to

the emergence of the balance in use of strategies to convey constituent roles that is found in natural language (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. Before these human-like results can be replicated with neural agents, more natural cognitive biases supporting efficiency need to be modeled, while the speaker training objective needs to 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).

	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	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 M3 down 1 M4 right 3
		M1 right 1 M2 up 2 M4 right 3 M3 down 1	M2 up 2 M4 right 3 M3 down 1 M1 right 1
		M3 down 1 M1 right 1 M4 right 3 M2 up 2	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	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 M3 down 1 M4 right 3
		M1 right 1 M2 up 2 M4 right 3 M3 down 1	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	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 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
		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	M3 down 1 M4 right 3 M2 up 2 M1 right 1	right 1 up 2 down 1 right 3
	M1 right 1 M2 up 2 M3 down 1 M4 right 3	M2 up 2 M3 down 1 M4 right 3 M1 right 1	
	M1 right 1 M2 up M3 down 1 M4 right 3 M5 3	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	
Iter_10	M1 right 1 M2 up 2 M3 down 1	right 1 up 2 down 1 right 3	right 1 up 2 down 1 right 3
	M1 right 1 M2 up 2	M2 up 2 M3 down 1 M4 right 3 M1 right 1	right 1 up 2
	M1 right 1	M1 right 1 M3 down 1 M4 right 3 M2 up 2	right 1
		M1 right 1 M3 down 1 M4 right 3 M2 up 2	
		right 1 up 2 down 1 right 3	

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

Table 2.4: Utterances sampled from the agents' speaking network given the trajectory **'right up up down right right right'** in Fix+Marker language learning with pressure (§2.4), Mix language learning without and with pressure (§2.5.1), Mix_drop language learning (§2.5.2) and Mix language with learning bottleneck (§2.6). For each experiment and each generation, we show six randomly sampled utterances (duplicates are omitted for clarity).