Emerging linguistic universals in communicating neural network agents

Rahma Chaabouni

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Les universaux linguistiques émergeant dans les réseaux de neurones communicants

Emerging linguistic universals in communicating neural network agents

Soutenue par
Rahma CHAABOUNI
Le 17 mars 2021

Ecole doctorale n°158
ECOLE DOCTORALE CERVEAU-COGNITION-COMPORTEMENT (ED3C)

Spécialité
Apprentissage Automatique/Sciences Cognitives

Composition du jury :

Olivier, BONAMI PU, Université de Paris	*Rapporteur, Président*
Angeliki, LAZARIDOU CH, DeepMind London	*Rapporteuse*
Emmanuel, CHEMLA DR, Ecole Normale Supérieure	*Examinateur*
Olivier, PIETQUIN PU, Université de Lille, Google Research	*Examinateur*
Emmanuel, DUPOUX DE, ENS, EHESS, INRIA, Facebook	*Directeur de thèse*
Marco, Baroni PR, ICREA, Facebook	Co-*directeur de thèse*

ENS DEC DÉPARTEMENT D'ÉTUDES COGNITIVES

PSL RESEARCH UNIVERSITY PARIS

Emerging linguistic universals in communicating neural network agents

by

Rahma Chaabouni

March 2021

Abstract

The ability to acquire and produce a language is a key component of intelligence. If communication is widespread among animals, human language is unique in its productivity and complexity. By better understanding the source of natural language, one can use this knowledge to build better interactive AI models that can acquire human languages as rapidly and efficiently as children. In this manuscript, we build up on the emergent communication field to investigate the well-standing question of the source of natural language. In particular, we use communicating neural networks that can develop a language to solve a collaborative task. Comparing the emergent language properties with human cross-linguistic regularities can provide answers to the crucial questions of the origin and evolution of natural language. Indeed, if neural networks develop a cross-linguistic regularity spontaneously, then the latter would not depend on specific biological constraints. From the cognitive perspective, looking at neural networks as another expressive species can shed light on the source of cross-linguistic regularities – a fundamental research interest in cognitive science and linguistics. From the machine learning perspective, endowing artificial models with human constraints necessary to evolve communicative protocols as productive and robust as natural language would encourage the development of better interactive AI models.

In this manuscript, we focus on studying four cross-linguistic regularities related to word length, word order, semantic categorization, and compositionality. Across the different studies, we find that some of these regularities arise spontaneously while others are missing in neural networks' languages. We connect the former case to the presence of shared communicative constraints such as the discrete nature of the communication channel. On the latter, we relate the absence of human-like regularities to the lack of constraints either on the learners' side (e.g., the least-effort constraints) or language functionality (e.g., transmission of information). In sum, this manuscript provides several case studies demonstrating how we can use successful neural network models to tackle crucial questions about the origin and evolution of our language. It also stresses the importance of mimicking the way humans learn their language in artificial agents' training to induce better learning procedures for neural networks, so that they can evolve an efficient and open-ended communication protocol.

Résumé

La capacité d'acquérir et de produire un langage est un élément clé de l'intelligence humaine. En effet, même si de nombreuses espèces partagent un système de communication, le langage humain reste unique par sa productivité, sa récursivité ainsi que le nombre de symboles utilisés. En comprenant mieux les origines de l'apparition du langage, il sera possible de créer des modèles plus performants capable d'interagir et d'acquérir notre langage aussi rapidement et efficacement que nous le faisons en tant que bébé.

Dans ce manuscrit, nous utilisons des réseaux de neurones communicants qui peuvent développer et faire évoluer un langage pour nous éclairer sur la question de l'origine du langage naturel. Nous comparons ensuite les propriétés de leur langage émergeant avec les propriétés universelles du langage naturel. Si les réseaux de neurones produisent spontanément une propriété linguistique, celle-ci ne dépendrait pas alors des contraintes biologiques. Autrement, dans le cas où le langage artificiel dévie du langage humain pour une régularité donnée, cette dernière ne peut être considérée comme une conséquence des simples contraintes de communication. D'un point de vue cognitif, considérer les réseaux de neurones comme une autre espèce expressive peut nous éclairer sur la source des propriétés universelles. Du point de vue de l'apprentissage automatique, doter les modèles artificiels de contraintes humaines nécessaires pour faire évoluer des protocoles de communication aussi productifs et robustes que le langage naturel encouragerait le développement de meilleurs modèles d'IA interactifs.

Ce manuscrit traite de l'étude de quatre régularités linguistiques qui ont à voir avec la longueur des mots, l'ordre des mots, la catégorisation sémantique et la compositionnalité. Certains chapitres exemplifient des cas où les régularités apparaissent spontanément dans le langage émergeant, tandis que d'autres montrent des cas où les réseaux de neurones développent un langage qui dévie du langage naturel. Nous avons relié le premier cas à la présence de contraintes de communication telles que la nature discrète du canal de communication. Quant à l'absence de régularités naturelles, nous l'avons lié au manque de contraintes soit au niveau de l'apprenant (par exemple, la contrainte biologique de brièveté) soit au niveau de l'environnement (par exemple, la richesse d'environnement). Ainsi, cet ensemble de travaux fournit plusieurs études de cas démontrant l'intérêt d'utiliser des modèles de réseaux de neurones performants dans des tâches de traitement de texte pour aborder des questions cruciales sur l'origine et l'évolution de notre langage. Il souligne également l'importance d'entraîner les réseaux de neurones sous contraintes naturelles pour voir l'émergence d'un protocole de communication aussi efficace et productif que le langage naturel.

Acknowledgments

I have been looking forward to writing this section already when I knew I need to write a dissertation. However, now that I have finished the dissertation, thinking about all the support I had during and while before this wonderful three-year adventure, I realize that I can never be fair and thank enough the people that helped me, or even thank enough of them. Here I am, giving it a likely insufficient try;

I want to start by thanking my amazing PhD supervisors; Marco Baroni and Emmanuel Dupoux. I was very fortunate to have them as advisors who directed me when pitching a half-baked fuzzy idea, as collaborators who put a tremendous amount of time and effort into my papers, and as mentors who I appreciated their everyday care, assistance, and incredible positivity. I also want to thank my informal third supervisor Eugene Kharitonov, with whom I co-authored almost all my papers, starting from crazy discussions about solving AI. I have learned so much from these collaborations, both in terms of scientific writing and technical skills. I also sincerely appreciated his friendship and his funny ways to encourage me throughout my PhD. My three supervisors are responsible for the hopefully stronger person I have become after these incredible three years, full of tiredness and anger defeated at the end by the excitement and the satisfaction of research. Simply said, I have learned so much from Marco, Emmanuel, and Eugene and benefited from their complementary perspectives while admiring their immense kindness.

I want to thank Olivier Bonami, Angeliki Lazaridou, Olivier Pietquin, and Emmanuel Chemla for accepting to be on my jury and for their great comments. I am also very grateful for my collaborations with Alessandro Lazaric, who helped me define my PhD project, with Diane Bouchacourt, who was a great mentor to me, with Roberto Dessì and Mathieu Rita, who I admired their intellectual curiosity and incredible efficiency. Throughout the three years of PhD, I interacted with smart and passionate researchers and engineers both at CoML (LSCP, ENS) and FAIR Paris. The everyday discussions were a great source of motivation in the highs and resilience in the lows. I want to thank specifically CoML PhD students Juliette Millet and Robin

Algayres for the enlightening discussions during the reading groups, Rachid Riad for his recurrent participation in the writing club challenge motivating me during the dissertation and the endless color paper writing, Marvin Lavechin and Maureen de Seyssel for coming to the lab when the COVID19 situation allowed and contributing to sustaining my mental health during these challenging times. I also want to thank the FAIR Paris team, in particular, Jeremy Rapin and Louis Martin, for putting a lot of effort to limit our isolation during the sanitary crisis, the ingenious "$$Humans of Late Machine Learning$$" chat for supporting me with the funny GIFs and the serious information, Hervé Jégou for being a great mentor during the last weeks of my PhD, and Patricia Le Carré for her reconforting care and help with administrative tasks.

Next, I want to thank the "MOB", the "Supelec Tunisian gang", and the "RISK-becoming-7Wonders group" for being such great friends and organizing amazing adventures when I needed a break. A special mention to Yosr Zenzri and Zohour Hamza for believing in me and encouraging me when I needed. During this three-year adventure, I have been fully supported by my family. I want to thank my parents, whose perfection is still a mystery for me. I want to specifically thank them for their unconditional love and unlimited trust. They taught me that I can follow my dreams whatever it takes, and I would have never been able to even start this amazing opportunity I am living right now without their constant encouragement. I also thank my brother, who, to my great happiness, came to Paris to pursue his studies. He fed me with his delicious and very caloric dishes during my deadlines and made me laugh when I felt low. Finally, I am very lucky to have Malek as a husband, for supporting me at every decision I made, for trusting me about any project I pursue, for reading and criticizing each first draft I wrote, for making life as great and easy as no one can imagine for ME. *Thank you!!!*

Publications

Included in the main

- **Rahma Chaabouni**, Eugeny Kharitonov, Emmanuel Dupoux, Marco Baroni. *Anti-efficient encoding in emergent communication* . NeurIPS 2019.
- Mathieu Rita, **Rahma Chaabouni**, Emmanuel Dupoux, *Lazy and Impatient neural agents learn to communicate efficiently.* CoNLL 2020.
- **Rahma Chaabouni**, Eugeny Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, Marco Baroni. *Word-order biases in deep-agent emergent communication.* ACL 2019.
- **Rahma Chaabouni**, Eugeny Kharitonov, Emmanuel Dupoux, Marco Baroni, *Communicating artificial neural networks develop efficient color-naming systems.* PNAS.
- **Rahma Chaabouni***, Eugeny Kharitonov*, Diane Bouchacourt, Emmanuel Dupoux, Marco Baroni. *Compositionality and generalization in emergent languages.* ACL 2020.

Included in the appendices

- Eugeny Kharitonov, **Rahma Chaabouni**, Diane Bouchacourt, Marco Baroni. *EGG: a toolkit for research on Emergence of lanGuage in Games.* EMNLP 2019 (Demo).
- Eugeny Kharitonov, **Rahma Chaabouni**, Diane Bouchacourt, Marco Baroni. *Information Minimization In Emergent Languages.* ICML 2020

Not included in the manuscript

- Eugeny Kharitonov*, **Rahma Chaabouni***, *What they do when in doubt: a study of inductive biases in seq2seq learners.* ICLR2021.

- **Rahma Chaabouni***, Roberto Dessi, Eugeny Kharitonov*, *Can Transformers jump around right in natural language? Assessing performance transfer from SCAN.* Submitted.

Contents

Chapter 1

Introduction

Neural Networks (NNs) have reached astonishing success in natural language processing tasks, including machine translation [105], story generation [30], and open-domain dialog [1]. Such progress raises the question of how these models coincide with/differ from humans in language processing tasks. For example, can these powerful models evolve a communication protocol that shares natural language properties? Or expressed differently, which language properties are human-specific and which vary among language-capable learners (in this case NN learners)? Examining the uniqueness of natural language has been, and remains, the interest of various disciplines such as linguistics, cognitive science, philosophy, and AI (e.g., [100, 87, 20, 102]). It implies looking at several fundamental questions: How, among all possible expressive protocols, we converged to natural language? What is the source of language universals? Are they due to our innate biases, language functionality, or the constraints of our environment ...? In this work, we look at NNs as another expressive species that can develop and evolve a communication protocol, referred to as emergent language, to provide partial answers to these important questions. Specifically, we compare the emergent language developed by NN agents with natural language. On the one hand, if both languages share a property, then the latter would not depend on specific biological constraints. On the other hand, the fact that both languages diverge on a specific property suggests that the latter could not be a consequence of the communicative constraints shared with NNs. This thinking follows McCloskey's

view already proposed in 1991 [82], and in line with the methodology of cross-species comparative linguistics [112, 11]. Further, NN models have an appealing flexibility that we lack when experimenting with animals and humans. For example, one can easily impact the NNs environment, their "biological" constraints, the functionality of their emergent language, and so on, as we will see in the following chapters of the manuscript. Finally, from an AI and engineering point of view, the study of NN emergent language can provide guidance for developing better AI models. Concretely, one way to develop artificial models that interact with us and acquire our language as rapidly and efficiently as we do as children [28, 4], is to endow them with the human constraints necessary to evolve communicative protocols as productive and robust as natural language.

1.1 Universal language properties

There is considerable variation across the languages of the world. For example, a mammal that barks is referred to by English-speakers as "*dog*", while French-speakers employ the word "*chien*". More generally, languages differ in the set of sounds used to form words, the words themselves, and even in the ways in which these words are combined into a sentence. Yet, it is possible to discern common regularities/universal properties [41, 54]. The underlying source of these cross-linguistic regularities is the subject of a long-standing debate across different areas of linguistics. Some linguists suggest that language regularities derive from Universal Grammar, a set of innate constraints on language acquisition [19, 111, 33]. Others see these properties as arising from language use, governed by general cognitive constraints (i.e., unlike Universal Grammar that is specific to language) [66, 110, 40, 16]. Even within the latter view, it is not clear how domain-general cognitive processes interact with each other to shape these regularities. Computational modeling provides a flexible framework to test these assumptions, as one can directly intervene on different components that could impact the properties of the language. In this work, we experiment with NNs to investigate the source of certain universals. In particular, we look at four properties

that are well-suited to study with NN simulations, namely word length, word order, semantic categorization, and compositionality. As we will detail below, most of these properties have been argued to result from competing pressures on the speaker side and the listener side [25, 2, 49, 39].

Word Length A robust cross-linguistic universal is a statistical law popularized by George K. Zipf and known as Zipf's Law of Abbreviation (ZLA) [120]. ZLA states that more frequent words tend to be shorter. For example, in English, the most frequent word "*the*" is obviously shorter than "*something*" (the 219^{th} most frequent word).[1] Using a communication protocol that follows ZLA leads then to a low average length of messages, especially given Zipf's other contribution stating that words follow a power-law distribution (commonly known as the Zipf distribution) [119]. Assuming that shorter words are easier to produce, a ZLA-obeying language is an efficient encoding. However, if subject to the efficiency constraint alone, natural language would have been highly ambiguous, using one short word to refer to many meanings. Yet, natural language supports efficient *and* accurate communication, suggesting that the effort-minimization pressure is in direct conflict with the pressure for accurate communication. Importantly, ZLA was robustly attested across different languages [104, 106, 7], leading many researchers, starting with Zipf, to hypothesize that human languages are universally shaped by pressures toward minimizing effort (on the speaker side) and maximizing communication success (on the listener side) [119, 92, 80, 39]. On the other hand, the ubiquity of ZLA among different, possibly non-expressive, codes brought other researchers to propose an alternative explanation. For example, Miller and colleagues proved that a lexicon generated by randomly typing on a keyboard also obeys ZLA [84]. This suggests that ZLA might emerge naturally due to unrelated statistical processes, which weakens Zipf's hypothesis. As a response, Kanwal and colleagues, in a more recent work, experimented directly on human subjects, evaluating their preference for short vs. long word forms while controlling their effort-minimization pressure [57]. This work shows that sub-

[1]Source: project Gutenberg (`https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/PG/2006/04/1-10000`)

jects preferred the short form only when under *both* pressures of effort-minimization and successful communication. If these results constitute an important proof of concept for Zipf's hypothesis, the experiment was restricted to 2 possible objects to refer to with 2 *given* artificial words, limiting the generality of their finding. Also, experimenting with only adult subjects, who have already acquired a ZLA-following language (English in their case), can interfere with the experiment. In **Chapter 2**, we complement this line of work using NNs that need to communicate to solve a common task. Our study asks whether communicating NNs would develop a ZLA-like protocol from scratch when faced with different environments and subject to different constraints.

Word Order One cross-linguistic tendency related to word order is that there is an inverse correlation between the use of case and the use of fixed word order to disambiguate words' role in sentences. That is, languages with freer word order tend to use explicit grammatical marking (e.g., Japanese), whereas languages with more fixed word order rarely rely on case-marking (e.g., English) [23, 10, 99]. Another general word order tendency is the preference of local dependencies [48, 75, 107, 44]. Expressed differently, words that are syntactically linked together tend to occur close to each other. Such property was observed across 37 diverse languages [37].

If many typological studies identify word order regularities, fewer works have addressed the question of their underlying source. One common line of research, though, is the use of artificial miniature languages with human subjects. The idea is to investigate how robustly a "natural" miniature language is acquired/maintained by humans compared to a "non-natural" one (see [31] for a survey). For instance, regarding the word order and case marking trade-off, Fedzechkina and colleagues have found that human learners of free word order languages tend to retain more case-marking compared to learners of fixed word order languages [32]. In other words, human learners deviate from the actual miniature language to mirror the typological pattern of case and word order trade-off. Given this observation, Fedzechkina and colleagues suggested that such pattern is due to humans innate bias against ambiguous languages

(i.e., the ones that use neither case nor word order for marking grammatical functions) *and* redundant languages (i.e., the ones employing both means for marking grammatical functions), thus echoing respectively the listener and speaker pressures of Zipf's hypothesis [119]. However, one frequent criticism of this approach is that human subjects' preferences can be reduced to their native language biases [89]. In fact, it is hard to disentangle the preferences due to innate biases from the ones due to the *a priori* learned natural language. In **Chapter 3**, we extend this line of research and use NN agents with no *a priori* learned language to investigate the relationship between several word order patterns and NN "inductive" biases. In particular, we look at 3 word order patterns commonly observed in natural language: (1) iconicity, (2) the trade-off between case-marking and fixed word-order, and (3) local dependencies preference.

Semantic Categorization Natural language is grounded in a diverse and complex environment. Yet, it succeeds in conveying our experience accurately. Specifically, to deal with the complexity of our environment, we use naming systems to partition our world into semantic categories. Identifying which semantic properties are universals and which vary among human languages has been the interest of many researchers for decades (e.g., [85, 42, 24] among many others). To that end, a common practice is to compare how a set of meanings are grouped into categories across different languages [8, 46]. This approach allows examining which properties of semantic categories persist across languages and which do not, conjecturing a set of cross-linguistic regularities/variations. A commonly noticed regularity, deriving from this large body of studies, is that semantic categories are built to support efficient communication [58, 59, 96, 118]. That is, humans develop naming systems under the competing pressures of being simultaneously informative (maximizing communicative success on the listener side) and simple (minimizing the cognitive load on the speaker side).[2] Despite the ubiquity of this observation, the question of the origin of these pressures remains unanswered. Since naming systems establish themselves on an evolutionary scale, it

[2]The definition of simplicity varies across different works. We will return to this notion in **Chapter 4** and illustrate one previously introduced definition.

is difficult to study them experimentally with human subjects (but see, for example, [83]). One appealing alternative is to study computational communication systems, where we can intervene directly on various aspects of the simulation to ascertain their impact. In **Chapter 4**, we study NN's color-naming systems and compare them to human color-naming systems in terms of efficiency and simplicity. If our work focuses on the well-studied color domain, the introduced framework is not domain-specific and can be applied on other semantic categorization domains.

Compositionality Natural language is compositional as the meaning of an utterance is a function of its constituents and its structure [51]. Thus, by acquiring a set of finite rules and basic form-meaning mappings, a learner can refer to an infinite set of complex meanings. In addition to the obvious benefit of light memory-load, compositionality has long been thought of as the source of natural language productivity, enabling the transmission of an open-ended set of messages [34, 81, 88]. Such a property, missing in modern NNs [64, 117], is hence crucial for general AI [65, 4]. Despite the centrality of compositionality in language, there is still no definite answer to the question of its origin. One view argues that our strong innate biases, which could result from evolutionary processes, led to the emergence of a compositional code [93, 9]. Others claim that compositionality emerges from a trade-off between pressures for compressibility and expressivity [62]. Understanding the origin of this universal in natural language and endowing artificial agents with compositional reasoning is hence both beneficial for cognitive science and AI. In **Chapter 5**, we investigate if communicating NN agents develop spontaneously productive languages and if productivity requires compositionality. To this end, we introduce multiple measures that quantify different variants of compositionality, particularly those that are considered in most emergent languages studies [63, 67].

1.2 Why neural networks?

Where do these universal properties come from? Are they specific to human languages or derived from pressures that are common to any communicative system? Can other species develop emergent languages that also share these cross-linguistic regularities? If so, can it inform us about their underlying source?

To answer these questions, previous works have applied computational modeling to the language evolution field (e.g., [5, 108, 22, 94, 13, 103]). Considering some examples among these studies, some suggest that language regularities can derive from pragmatic properties of communicative interactions [101, 6], others, from cultural transmission across generations of learners [12, 14]. However, most of these works rely on task-tailored agents such as Bayesian models defining a pre-specific plausible hypothesis space. Instead, in this manuscript, we use existing generic and powerful models, namely NNs. NNs, unlike the models used in the former approach, were optimized for engineering purposes and showed astonishing capabilities in the processing of sequences in general, and of human languages in particular. Our approach is to ask if powerful natural language processing models can be used to study humans' language evolution.

1.3 Signaling Game

To make NNs develop a communication protocol, we use signaling games. The signaling game is a framework introduced by Davis Lewis in 1969 to study how individuals can evolve a language [70]. The game has two players: a speaker and a listener, and proceeds as follow:

1. The environment chooses a referent r at random, drawn from a specified distribution.

2. The speaker observes the sampled referent r and sends a message m to the listener, who does not have direct access to r.

3. The listener, based on m, needs to guess the referent r.

Thus, the signaling game is a collaborative game where both players have a common interest: If the listener infers the speaker's intent, both players succeed; otherwise, they both fail. Importantly, there is no pre-established set of messages associated to referents, and both players need to agree on the referent $\rightarrow$ message (i.e., concept $\rightarrow$ form) mapping to succeed in the game. In our work, we study whether cross-linguistic universals are also found in messages developed by *NN players*. To this end, we consider different variants of NN architectures such as LSTMs [50] and GRUs [18]. In most cases, both the speaker and the listener NNs have the same architecture, but we also explored the impact of asymmetric architectures in **Chapter 5**. Moreover, we restrict ourselves to analyze the emergent messages only when NN agents succeed in the signaling game.

Specifically, we examine two types of signaling games: reconstruction and discrimination games.

Reconstruction Game One variant of the Lewis game is the reconstruction game [69, 45, 60]. In this approach, for a given referent r_S, chosen randomly by the environment, the speaker chooses a message m (one or a sequence of discrete symbols). Provided with m, the listener needs to reconstruct back r_S. As the game is collaborative, if the listener's guess is equal to the speaker's referent, both agents succeed; otherwise, they fail.

Discrimination Game Another studied game is the discrimination game [68, 29, 47]. First, the environment provides a referent r_S to the speaker. Second, the speaker chooses a message m to describe r_S. However, unlike in the former game, now, given m, the listener needs to distinguish r_S among N given possible referents r_{L_1}, ... r_{L_N}. In other words, the listener chooses which referent included in $\{r_{L_1}, ...r_{L_N}\}$ corresponds to r_S. If this variant is more scalable (as it does not require the listener to choose from all possible referents in the environment), it adds further complications. Indeed, the emergent language in this variant depends, on top of other parameters, on N and the distribution of the listener's inputs [67].

In the following chapters, we consider different types of referents, from one-hot vectors in **Chapter 2** to continuous n-dimensional vectors in **Chapter 4**. Furthermore, we assume different distributions on the referents. We generally consider language emergence from scratch. However, in **Chapter 3**, different agents were seeded with different languages controlling specific properties. Note that even in that case, only the speaker learned a referent $\rightarrow$ message mapping, and still needs to transmit it to the listener. The imperfect transmission enables the study of emergent language evolution highlighting players' biases.

On the practical side, signaling games require non-conventional back-propagation optimization. That is, because of the discrete nature of the communication, NN players are optimized using either Monte Carlo approximation [115] or Gumbel-softmax continuous relaxation [55, 78], which, in turn, makes experimentation technically challenging. These technical challenges might discourage scientists from fields such as linguistics, cognitive science, or philosophy, whose interdisciplinary expertise would be precious to advance language evolution simulations. For this reason, we designed and open-sourced the EGG (**E**mergence of lan**G**uage in **G**ames) toolkit. The latter, based on Pytorch [91], provides an implementation of the signaling games variants while transparently taking care of the technical challenges related to NN optimization (see Appendix for more details). Most of our experiments are based on the EGG toolkit. We hope that the latter will lower the technical barrier and encourage interdisciplinary contributions in this field.

Chapter 2

Word Length

It is widely argued that our language can be reduced to general properties of the cognitive system. One well-studied pattern, hypothesized to be shaped by our cognitive system, is Zipf's Law of Abbreviation (ZLA). This law states that there is an inverse correlation between word frequency and length [119]. ZLA is argued to derive from a trade-off between communicative success and the Least Effort Principle [119, 106, 57]. On the one hand, a successful transmission requires low confusability over distinct words on the listeners' side. Yet, the shorter words are, the less distinct and more confusable they can be. On the other hand, the Least Effort Principle stipulates that speakers favor short words since they require less effort to produce. Thus, the Least Effort Principle is in direct conflict with communication success.

In this chapter, we ask whether high-performing NN agents that communicate to solve a task would develop a successful, ZLA-obeying protocol. **Section 2.1** shows that emergent NN languages exhibit an *anti*-ZLA pattern (that is, most frequent referents are associated with the longest words). We connect this to NNs' lack of "biological" pressure towards brevity (i.e., speaker NN has no incentive to produce short messages). The absence of this pressure is contrasted by NN (when listening) preference for long, more discriminable messages. We hence bring support for Zipf's hypothesis proving that ZLA does not emerge from trivial statistical properties, but from a delicate balance between the speaker and the listener preferences. On that basis, we introduce in **section 2.2** a new communication NN system where the speaker

NN is made increasingly *lazy* and the listener *impatient*. On top of its plausibility, our new communication system leads to the emergence of ZLA-like communication protocol, as efficient as natural language.

Anti-efficient encoding in emergent communication

Rahma Chaabouni[1,2], Eugene Kharitonov[1], Emmanuel Dupoux[1,2] and Marco Baroni[1,3]

[1]*Facebook AI Research*
[2]*Cognitive Machine Learning (ENS - EHESS - PSL Research University - CNRS - INRIA)*
[3]*ICREA*
{rchaabouni,kharitonov,dpx,mbaroni}@fb.com

Abstract

Despite renewed interest in emergent language simulations with neural networks, little is known about the basic properties of the induced code, and how they compare to human language. One fundamental characteristic of the latter, known as Zipf's Law of Abbreviation (ZLA), is that more frequent words are efficiently associated to shorter strings. We study whether the same pattern emerges when two neural networks, a "speaker" and a "listener", are trained to play a signaling game. Surprisingly, we find that networks develop an *anti-efficient* encoding scheme, in which the most frequent inputs are associated to the longest messages, and messages in general are skewed towards the maximum length threshold. This anti-efficient code appears easier to discriminate for the listener, and, unlike in human communication, the speaker does not impose a contrasting least-effort pressure towards brevity. Indeed, when the cost function includes a penalty for longer messages, the resulting message distribution starts respecting ZLA. Our analysis stresses the importance of studying the basic features of emergent communication in a highly controlled setup, to ensure the latter will not depart too far from human language. Moreover, we present a concrete illustration of how different functional pressures can lead to successful communication codes that lack basic properties of human language, thus highlighting the role such pressures play in the latter.

1 Introduction

There is renewed interest in simulating language emergence among neural networks that interact to solve a task, motivated by the desire to develop automated agents that can communicate with humans [e.g., Havrylov and Titov, 2017, Lazaridou et al., 2017, 2018, Lee et al., 2018]. As part of this trend, several recent studies analyze the properties of the emergent codes [e.g., Kottur et al., 2017, Bouchacourt and Baroni, 2018, Evtimova et al., 2018, Lowe et al., 2019, Graesser et al., 2019]. However, these analyses generally consider relatively complex setups, when very basic characteristics of the emergent codes have yet to be understood. We focus here on one such characteristic, namely the length distribution of the messages that two neural networks playing a simple signaling game come to associate to their inputs, in function of input frequency.

In his pioneering studies of lexical statistics, George Kingsley Zipf noticed a robust trend in human language that came to be known as Zipf's Law of Abbreviation (ZLA): There is an inverse (non-linear) correlation between word frequency and length [Zipf, 1949, Teahan et al., 2000, Sigurd et al., 2004, Strauss et al., 2007]. Assuming that shorter words are easier to produce, this is an efficient encoding strategy, particularly effective given Zipf's other important discovery that word distributions are highly skewed, following a power-law distribution. Indeed, in this way language approaches an optimal code in information-theoretic terms [Cover and Thomas, 2006]. Zipf, and many after him, have thus used ZLA as evidence that language is shaped by functional pressures toward effort

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

minimization [e.g., Piantadosi et al., 2011, Mahowald et al., 2018, Gibson et al., 2019]. However, others [e.g., Mandelbrot, 1954, Miller et al., 1957, Ferrer i Cancho and del Prado Martín, 2011, del Prado Martín, 2013] noted that some random-typing distributions also respect ZLA, casting doubts on functional explanations of the observed pattern.

We study a *Speaker* network that gets one out of $1K$ distinct one-hot vectors as input, randomly drawn from a power-law distribution (so that frequencies are extremely skewed, like in natural language). Speaker transmits a variable-length *message* to a *Listener* network. Listener outputs a one-hot vector, and the networks are rewarded if the latter is identical to the input. There is no direct supervision on the message, so that the networks are free to create their own "language". The networks develop a successful communication system that does *not* exhibit ZLA, and is indeed *anti-efficient*, in the sense that all messages are long, and the most frequent inputs are associated to the longest messages. Interestingly, a similar effect is observed in artificial human communication experiments, in conditions in which longer messages do not demand extra effort to speakers, so that they are preferred as they ease the listener discrimination task [Kanwal et al., 2017]. Our Speaker network, unlike humans, has no physiological pressure towards brevity [Chaabouni et al., 2019], and our Listener network displays an *a priori* preference for longer messages. Indeed, when we penalize Speaker for producing longer strings, the emergent code starts obeying ZLA. We examine the implications of our findings in the Discussion.

2 Setup

2.1 The game

We designed a variant of the Lewis signaling game [Lewis, 1969] in which the input distribution follows a power-law distribution. We think of these inputs as a vocabulary of distinct abstract *word types*, to which the agents will assign specific word forms while learning to play the game. We leave it to further research to explore setups in which word type and form distributions co-evolve [Ferrer i Cancho and Díaz-Guilera, 2007]. Importantly, our basic inefficient encoding result also holds when the inputs are uniformly distributed (Appendix A.1.5). Formally, the game proceeds as follows:

1. The Speaker network receives one of $1K$ distinct one-hot vectors as input i. Inputs are not drawn uniformly, but, like in natural language, from a power-law distribution. That is, the r^{th} most frequent input i_r has probability $\frac{1}{r \times \sum_{k=1}^{1000} \frac{1}{k}}$ to be sampled, with $r \in [\![1, ..., 1000]\!]$. Consequently, the probability of sampling the 1^{st} input is 0.13 while the probability of sampling the 1000^{th} one is 1000 times lower.
2. Speaker chooses a sequence of symbols from its alphabet $A = \{s_1, s_2..., s_{a-1}, \texttt{eos}\}$ of size $|A| = a$ to construct a message m, terminated as soon as Speaker produces the 'end-of-sequence' token `eos`. If Speaker has not yet emitted `eos` at `max_len` $- 1$, it is stopped and `eos` is appended at the end of its message (so that all messages are suffixed with `eos` and no message is longer than `max_len`).
3. The Listener network consumes m and outputs $\hat{i}$.
4. The agents are successful if $i = \hat{i}$, that is, Listener reconstructed Speaker's input.

The game is implemented using the EGG toolkit [Kharitonov et al., 2019], and the code can be found at `https://github.com/facebookresearch/EGG/tree/master/egg/zoo/channel`.

2.2 Architectures

As standard in current emergent-language simulations [e.g., Lazaridou et al., 2018], both agents are implemented as single-layer LSTMs [Hochreiter and Schmidhuber, 1997]. Speaker's input is a $1K$-dimensional one-hot vector i, and the output is a sequence of symbols, defining message m. This sequence is generated as follows. A linear layer maps the input vector into the initial hidden state of Speaker's LSTM cell. Next, a special start-of-sequence symbol is fed to the cell. At each step of the sequence, the output layer defines a Categorical distribution over the alphabet. At training time, we sample from this distribution. During evaluation, we select the symbol greedily. Each selected symbol is fed back to the LSTM cell. The dimensionalities of the hidden state vectors are part of the hyper-parameters we explore (Appendix A.1.1). Finally, we initialize the weight matrices of our

agents with a uniform distribution with support in $[-\frac{1}{\sqrt{\texttt{input\_size}}}, \frac{1}{\sqrt{\texttt{input\_size}}}]$, where `input_size` is the dimensionality of the matrix input (Pytorch default initialization).

Listener consumes the entire message m, including `eos`. After `eos` is received, Listener's hidden state is passed through a fully-connected layer with softmax activation, determining a Categorical distribution over $1K$ indices. This distribution is used to calculate the cross-entropy loss w.r.t. the ground-truth input, i.

The joint Speaker-Listener architecture can be seen as a discrete auto-encoder [Liou et al., 2014].

2.3 Optimization

The architecture is not directly differentiable, as messages are discrete-valued. In language emergence, two approaches are dominantly used: Gumbel-Softmax relaxation [Maddison et al., 2016, Jang et al., 2016] and REINFORCE [Williams, 1992]. We also experimented with the approach of Schulman et al. [2015], combining REINFORCE and stochastic backpropagation to estimate gradients. Preliminary experiments showed that the latter algorithm (to be reviewed next) results in the fastest and most stable convergence, and we used it in all the following experiments. However, the main results we report were also observed with the other algorithms, when successful.

We denote by $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_l$ the Speaker and Listener parameters, respectively. $\mathcal{L}$ is the cross-entropy loss, that takes the ground-truth one-hot vector i and Listener's output $L(m)$ distribution as inputs. We want to minimize the expectation of the cross-entropy loss $\mathbb{E}\ \mathcal{L}(i, L(m))$, where the expectation is calculated w.r.t. the joint distribution of inputs and message sequences. The gradient of the following surrogate function is an unbiased estimate of the gradient $\nabla_{\boldsymbol{\theta}_s \cup \boldsymbol{\theta}_l} \mathbb{E}\ \mathcal{L}(i, L(m))$:

$$\mathbb{E}\left[\mathcal{L}(i, L(m; \boldsymbol{\theta}_l)) + (\{\mathcal{L}(i, L(m; \boldsymbol{\theta}_l)\} - b) \log P_s(m|\boldsymbol{\theta}_s)\right] \tag{1}$$

where $\{\cdot\}$ is the stop-gradient operation, $P_s(m|\boldsymbol{\theta}_s)$ is the probability of producing the sequence m when Speaker is parameterized with vector $\boldsymbol{\theta}_s$, and b is a running-mean baseline used to reduce the estimate variance without introducing a bias. To encourage exploration, we also apply an entropy regularization term [Williams and Peng, 1991] on the output distribution of the speaker agent.

Effectively, under Eq. 1, the gradient of the loss w.r.t. the Listener parameters is found via conventional backpropagation (the first term in Eq. 1), while Speaker's gradient is found with a REINFORCE-like procedure (the second term). Once the gradient estimate is obtained, we feed it into the Adam [Kingma and Ba, 2014] optimizer. We explore different learning rate and entropy regularization coefficient values (Appendix A.1.1).

We train agents for 2500 episodes, each consisting of 100 mini-batches, in turn including 5120 inputs sampled from the power-law distribution with replacement. After training, we present to the system each input once, to compute accuracy by giving equal weight to all inputs, independently of amount of training exposure.

2.4 Reference distributions

As ZLA is typically only informally defined, we introduce 3 reference distributions that display efficient encoding and arguably respect ZLA.

2.4.1 Optimal code

Based on standard coding theory [Cover and Thomas, 2006], we design an *optimal code* (OC) guaranteeing the shortest average message length given a certain alphabet size and the constraint that all messages must end with `eos`. The *shortest* messages are deterministically associated to the *most frequent* inputs, leaving longer ones for less frequent ones. The length of the message associated to an input is determined as follows. Let $A = \{s_1, s_2 ... s_{a-1}, \texttt{eos}\}$ be the alphabet of size a and i_r be the r^{th} input when ranked by frequency. Then i_r is mapped to a message of length

$$l_{i_r} = min\{n : \sum_{k=1}^{n} (a-1)^{k-1} \geq r\} \tag{2}$$

For instance, if $a = 3$, then there is only one message of length 1 (associated to the most frequent referent), 2 of length 2, 4 of length 3 etc.[1] Section 2 of Ferrer i Cancho et al. [2013] presents a proof of how this encoding is the maximally efficient one.

2.4.2 Monkey typing

Natural languages respect ZLA without being as efficient as OC. It has been observed that *Monkey typing* (MT) processes, whereby a monkey hits random typewriter keys including a space character, produce word length distributions remarkably similar to those attested in natural languages [Simon, 1955, Miller et al., 1957]. We thus adapt a MT process to our setup, as a less strict benchmark for network efficiency.[2]

We first sample an input without replacement according to the power-law distribution, then generate the message to be associated with it. We repeat the process until all inputs are assigned a unique message. The message is constructed by letting a monkey hit the a keys of a typewriter uniformly at random ($p = 1/a$), subject to these constraints: (i) The message ends when the monkey hits `eos`. (ii) A message cannot be longer than a specified length `max_len`. If the monkey has not yet emitted `eos` at `max_len` $- 1$, it is stopped and `eos` is appended at the end of the message. (iii) If a generated message is identical to one already used, it is rejected and another is generated.

For a given length l, there are only $(a-1)^{l-1}$ different messages. Moreover, for a random generator with the `max_len` constraint, the probability of generating a message of length l is:

$$P_l = p \times (1-p)^{l-1}, \text{if } l < \texttt{max\_len} \text{ and } P_{\texttt{max\_len}} = (1-p)^{\texttt{max\_len}-1} \quad (3)$$

From these calculations, we derive two qualitative observations about MT. First, as we fix `max_len` and increase a (decrease $p = 1/a$), more generated messages will reach `max_len`. Second, when a is small and `max_len` is large (as in early MT studies where `max_len` was infinite), a ZLA-like distribution emerges, due to the finite number of *different* messages of length l. Indeed, for any l less than `max_len`, P_l strictly decreases as l grows. Then, for given inputs, the monkey is likely to start by generating messages of the most probable length (that is, 1). As we exhaust all unique messages of this length, the process starts generating messages of the next probable length (i.e., 2) and so on. Figure A1 in Appendix A.1.2 confirms experimentally that our MT distribution respects ZLA for $a \leq 10$ and various `max_len`.

2.4.3 Natural language

We finally consider word length distributions in natural language corpora. We used pre-compiled English, Arabic, Russian and Spanish frequency lists from `http://corpus.leeds.ac.uk/serge/`, extracted from corpora of internet text containing between $200M$ (Russian) and $16M$ words (Arabic). For direct comparability with input set cardinality in our simulations, we only looked at the distribution of the top 1000 most frequent words, after merging lower- and upper-cased forms, and removing words containing non-alphabetical characters. The resulting word frequency distributions obeyed power laws with exponents between -0.81 and -0.92 (we used -1 to generate our inputs). Alphabet sizes are as follows: 30 (English), 31 (Spanish), 47 (Russian), 59 (Arabic). These are larger than normative sizes, as unfiltered Internet text will occasionally include foreign characters (e.g., accented letters in English text). Contrary to previous reference distributions, we cannot control `max_len` and alphabet size. We hence compare human and network distributions only in the adequate settings. In the main text, we present results for the languages with the smallest (English) and largest (Arabic) alphabets. The distributions of the other languages are comparable, and presented in Appendix A.1.3.

3 Experiments

3.1 Characterizing the emergent encoding

We experiment with alphabet sizes $a \in [3, 5, 10, 40, 1000]$. We chose mainly small alphabet sizes to minimize a potential bias in favor of long messages: For high a, randomly generating long messages becomes more likely, as the probability of outputting `eos` at random becomes lower. At the other

[1] There is always only one message of length 1 (that is, `eos`), irrespective of alphabet size.

[2] No actual monkey was harmed in the definition of the process.

extreme, we also consider $a = 1000$, where the Speaker could in principle successfully communicate using at most 2-symbol messages (as Speaker needs to produce `eos`). Finally, $a = 40$ was chosen to be close to the alphabet size of the natural languages we study (mean alphabet size: 41.75).

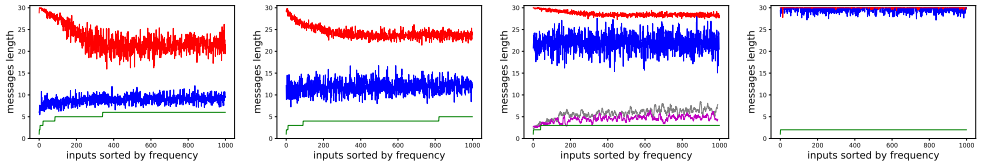
After fixing a, we choose `max_len` so that agents have enough capacity to describe the whole input space ($|I| = 1000$). For a given a and `max_len`, Speaker cannot encode more inputs than the message space size $M_a^{\texttt{max\_len}} = \sum_{j=1}^{\texttt{max\_len}} (a-1)^{j-1}$. We experiment with $\texttt{max\_len} \in [2, 6, 11, 30]$. We couldn't use higher values because of memory limitations. Furthermore, we studied the effect of $D = \frac{M_a^{\texttt{max\_len}}}{|I|}$. While making sure that this ratio is at least 1, we experiment with low values, where Speaker would have to use nearly the whole message space to successfully denote all inputs. We also considered settings with significantly larger D, where constructing $1K$ distinct messages might be an easier task.

We train models for each $(\texttt{max\_len}, a)$ setting and agent hyperparameter choice (4 seeds per choice). We consider runs successful if, after training, they achieve an accuracy above 99% on the full input set (i.e., less than 10 miss-classified inputs). As predicted, the higher D is, the more accurate the agents become. Indeed, agents need much larger D than strictly necessary in order to converge. We select for further analysis only those $(\texttt{max\_len}, a)$ choices that resulted in more than 3 successful runs (mean number of successful runs across the reported configurations is 25 out of 48). Moreover, we focus here on configurations with $\texttt{max\_len} = 30$, as the most comparable to natural language.[3] We present results for all selected configurations (confirming the same trends) in Appendix A.1.4.

Figure 1 shows message length distribution (averaged across all successful runs) in function of input frequency rank, compared to our reference distributions. The MT results are averaged across 25 different runs. We show the Arabic and English distributions in the plot containing the most comparable simulation settings $(30, 40)$.

Across configurations, we observe that Speaker messages greatly depart from ZLA. There is a clear general preference for longer messages, that is strongest *for the most frequent inputs*, where Speaker outputs messages of length `max_len`. That is, in the emergent encoding, more frequent words are longer, making the system obey a sort of "*anti*-ZLA" (see Appendix A.1.6 for confirmation that this anti-efficient pattern is statistically significant). Consequently, the emergent language distributions are well above all reference distributions, except for MT with $a = 1000$, where the large alphabet size leads to uniformly long words, for reasons discussed in Section 2.4.2. Finally, the lack of efficiency in emergent language encodings is also observed when inputs are uniformly distributed (see Appendix A.1.5).

Although some animal signing systems disobey ZLA, due to specific environmental constraints [e.g., Heesen et al., 2019], a large survey of human and animal communication did not find any case of significantly *anti*-efficient systems [Ferrer i Cancho et al., 2013], making our finding particularly intriguing.



(a)`max_len`=30, a=5 (b)`max_len`=30, a=10 (c)`max_len`=30, a=40 (d)`max_len`=30, a=1000

Figure 1: Mean message length across successful runs as a function of input frequency rank, with reference distributions. For readability, we smooth natural language distributions by reporting the sliding average of 10 consecutive lengths.

[3]Natural languages have no rigid upper bound on length, and 30 is the highest `max_len` we were able to train models for. Qualitative inspection of the respective corpora suggest that 30 is anyway a reasonable "soft" upper bound on word length in the languages we studied (longer strings are mostly typographic detritus).

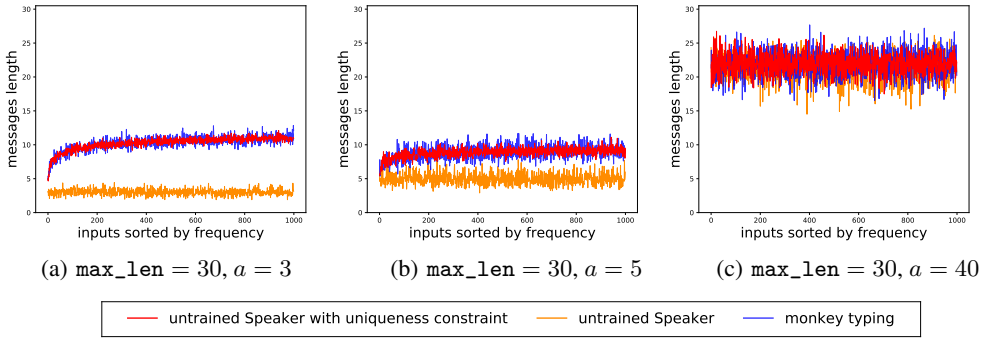
3.2 Causes of anti-efficient encoding

We explore the roots of anti-efficiency by looking at the behavior of untrained Speakers and Listeners. Earlier work conjectured that ZLA emerges from the competing pressures to communicate in a perceptually distinct *and* articulatorily efficient manner [Zipf, 1949, Kanwal et al., 2017]. For our networks, there is a clear pressure from Listener in favour of ease of message discriminability , but Speaker has no obvious reason to save on "articulatory" effort. We thus predict that the observed pattern is driven by a Listener-side bias.

3.2.1 Untrained Speaker behavior

For each i drawn from the power-law distribution without replacement, we get a message m from 90 distinct *untrained* Speakers (30 speakers for each hidden size in $[100, 250, 500]$). We experiment with 2 different association processes. In the first, we associate the first generated m to i, irrespective of whether it was already associated to another input. In the second, we keep generating a m for i until we get a message that was not already associated to a distinct input. The second version is closer to the MT process (see Section 2.4.2). Moreover, message uniqueness is a reasonable constraint, since, in order to succeed, Speakers need first of all to keep messages denoting different inputs apart.

Figure 2 shows that untrained Speakers have no prior toward outputting long sequences of symbols. Precisely, from Figure 2 we see that the untrained Speakers' average message length coincides with the one produced by the random process defined in Eq. 3 where $p = \frac{1}{a}$.[4] In other words, untrained Speakers are equivalent to a random generator with uniform probability over symbols.[5] Consequently, when imposing message uniqueness, non-trained Speakers become identical to MT. Hence, Speakers faced with the task of producing distinct messages for the inputs, if vocabulary size is not too large, would naturally produce a ZLA-obeying distribution, that is radically altered in joint Speaker-Listener training.



(a) `max_len` $= 30$, $a = 3$ (b) `max_len` $= 30$, $a = 5$ (c) `max_len` $= 30$, $a = 40$

Figure 2: Average length of messages by input frequency rank for untrained Speakers, compared to MT. See Appendix A.1.7 for more settings.

3.2.2 Untrained Listener behavior

Having shown that untrained Speakers do not favor long messages, we ask next if the emergent anti-efficient language is easier to discriminate by untrained Listeners than other encodings. To this end, we compute the average pairwise L2 distance of the hidden representations produced by untrained Listeners in response to messages associated to all inputs.[6] Messages that are further apart in the representational space of the untrained Listener should be easier to discriminate. Thus, if Speaker associates such messages to the inputs, it will be easier for Listener to distinguish them.

[4] Note that we did not use the uniqueness-of-messages constraint to define P_l.

[5] We verified that indeed untrained Speakers have uniform probability over the different symbols.

[6] Results are similar if looking at the softmax layer instead.

Specifically, we use 50 distinct untrained Listeners with 100-dimensional hidden size.[7] We test 4 different encodings: (1) emergent messages (produced by *trained* Speakers) (2) MT messages (25 runs) (3) OC messages and (4) human languages. Note that MT is equivalent to untrained Speaker, as their messages share the same length *and* alphabet distribution (see Section 3.2.1). We study Listeners' biases with `max_len` $= 30$ while varying a as messages are more distinct from reference distributions in that case (see Figure A3 in Appendix A.1.4). Results are reported in Figure 3. Representations produced in response to the emergent messages have the highest average distance. MT only approximates the emergent language for $a = 1000$, where, as seen in Figure 1 above, MT is anti-efficient. The trained Speaker messages are hence *a priori* easier for non-trained Listeners. The length of these messages could thus be explained by an intrinsic Listener's bias, as conjectured above. Also, interestingly, natural languages are not easy to process by Listeners. This suggests that the emergence of "natural" languages in LSTM agents is unlikely, without imposing *ad-hoc* pressures.

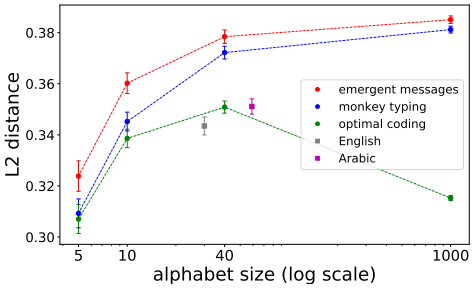


Figure 3: Average pairwise distance between messages' representation in Listener's hidden space, across all considered non-trained Listeners. Vertical lines mark standard deviations across Listeners.

3.2.3 Adding a length minimization pressure

We next impose an artificial pressure on Speaker to produce short messages, to counterbalance Listener's preference for longer ones. Specifically, we add a regularizer disfavoring longer messages to the original loss:

$$\mathcal{L}'(i, L(m), m) = \mathcal{L}(i, L(m)) + \alpha \times |m| \tag{4}$$

where $\mathcal{L}(i, L(m))$ is the cross-entropy loss used before, $|.|$ denotes length, and α is a hyperparameter. The non-differentiable term $\alpha \times |m|$ is handled seamlessly as it only depends on Speaker's parameters $\boldsymbol{\theta}_s$ (which specify the distribution of the messages m), and the gradient of the loss w.r.t. $\boldsymbol{\theta}_s$ is estimated via a REINFORCE-like term (Eq. 1). Figure 4 shows emergent message length distribution under this objective, comparing it to other reference distributions in the most human-language-like setting: (`max_len`=30, a=40). The same pattern is observed elsewhere (see Appendix A.1.8, that also evaluates the impact of the α hyperparameter). The emergent messages clearly follow ZLA. Speaker now assigns messages of ascending length to the 40 most frequent inputs. For the remaining ones, it chooses messages with relatively similar, but notably shorter, lengths (always much shorter than MT messages). Still, the encoding is not as efficient as the one observed in natural language (and OC). Also, when adding length regularization, we noted a slower convergence, with a smaller number of successful runs, that further diminishes when α increases.

3.3 Symbol distributions in the emergent code

We conclude with a high-level look at what the long emergent messages are made of. Specifically, we inspect symbol unigram and bigram frequency distributions in the messages produced by trained Sender in response to the $1K$ inputs (the `eos` symbol is excluded from counts). For direct comparability with natural language, we report results in the (`max_len`=30,a=40) setting, but the patterns are general. We observe in Figure 5(a) that, even if at initialization Speaker starts with a uniform distribution over its alphabet (not shown here), by end of training it has converged to a very skewed one. Natural languages follow a similar trend, but their distributions are not nearly as skewed (see

[7] We fix this value because, unlike for Speaker, it has considerable impact on performance, with 100 being the preferred setting.

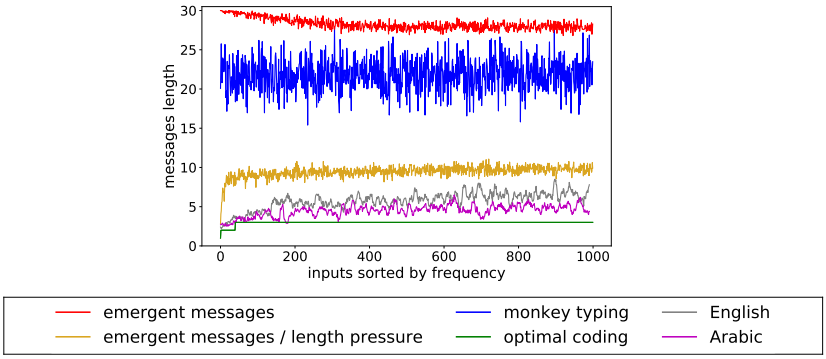
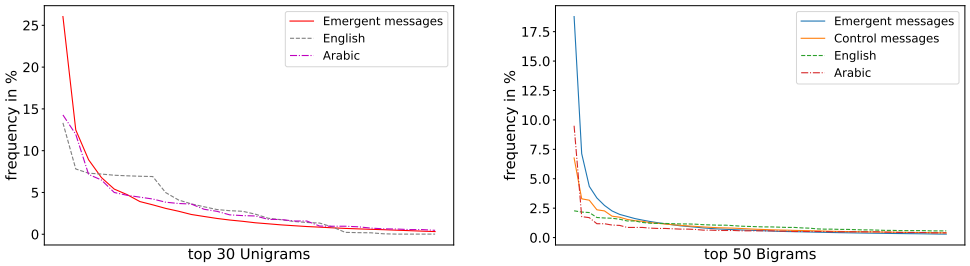


Figure 4: Mean length of messages across successful runs as a function of input frequency rank for `max_len` $= 30$, $a = 40$, $\alpha = 0.5$. Natural language distributions are smoothed as in Fig. 1.

Figure 8(a) in Appendix A.2.1 for entropy analysis). We then investigate message structure by looking at symbol bigram distribution. To this end, we build 25 randomly generated *control codes*, constrained to have the same mean length and unigram symbol distribution as the emergent code. Intriguingly, we observe in Figure 5(b) a significantly more skewed emergent bigram distribution, compared to the controls. This suggests that, despite the lack of phonetic pressures, Speaker is respecting "phonotactic" constraints that are even sharper than those reflected in the natural language bigram distributions (see Figure 8(b) in Appendix A.2.1 for entropy analysis). In other words, the emergent messages are clearly not built out of random unigram combinations. Looking at the pattern more closely, we find the skewed bigram distribution to be due to a strong tendency to repeat the same character over and over, well beyond what is expected given the unigram symbol skew (see typical message examples in Appendix A.2). More quantitatively, across all runs with `max_len`=30, if we denote the 10 most probable symbols with $s_1, ..., s_{10}$, then we observe $P(s_r, s_r) > P(s_r)^2$ with $r \in [\![1, .., 10]\!]$, in more than 97.5% runs. We leave a better understanding of the causes and implications of these distributions to future work.



(a) Symbol unigram distributions

(b) Symbol bigram distributions

Figure 5: Distribution of top symbol unigrams and bigrams (ordered by frequency) in different codes. Emergent and control messages are averaged across successful runs and different simulations respectively in the (`max_len`=30,a=40) setting.

4 Discussion

We found that two neural networks faced with a simple communication task, in which they have to learn to generate messages to refer to a set of distinct inputs that are sampled according to a power-law distribution, produce an *anti-efficient* code where more frequent inputs are significantly associated to longer messages, and all messages are close to the allowed maximum length threshold. The results are

stable across network and task hyperparameters (although we leave it to further work to replicate the finding with different network architectures, such as transformers or CNNs). Follow-up experiments suggest that the emergent pattern stems from an *a priori* preference of the listener network for longer, more discriminable messages, which is not counterbalanced by a need to minimize articulatory effort on the side of the speaker. Indeed, when an artificial penalty against longer messages is imposed on the latter, we see a ZLA distribution emerging in the networks' communication code.

From the point of view of AI, our results stress the importance of controlled analyses of language emergence. Specifically, if we want to develop artificial agents that naturally communicate with humans, we want to ensure that we are aware of, and counteract, their unnatural biases, such as the one we uncovered here in favor of anti-efficient encoding. We presented a proof-of-concept example of how to get rid of this specific bias by directly penalizing long messages in the cost function, but future work should look into less *ad hoc* ways to condition the networks' language. Getting the encoding right seems particularly important, as efficient encoding has been observed to interact in subtle ways with other important properties of human language, such as regularity and compositionality [Kirby, 2001]. We also emphasize the importance of using power-law input distributions when studying language emergence, as the latter are a universal property of human language [Zipf, 1949, Baayen, 2001] largely ignored in previous simulations, that assume uniform input distributions.

ZLA is observed in all studied human languages. As mentioned above, some animal communication systems violate it [Heesen et al., 2019], but such systems are 1) limited in their expressivity; and 2) do not display a significantly *anti*-efficient pattern. We complemented this earlier comparative research with an investigation of emergent language among artificial agents that need to signal a large number of different inputs. We found that the agents develop a successful communication system that does *not* exhibit ZLA, and is actually significantly anti-efficient. We connected this to an asymmetry in speaker vs. listener biases. This in turn suggests that ZLA in communication in general does not emerge from trivial statistical properties, but from a delicate balance of speaker and listener pressures. Future work should investigate emergent distributions in a wider range of artificial agents and environments, trying to understand which factors are determining them.

5 Acknowledgments

We would like to thank Fermín Moscoso del Prado Martín, Ramon Ferrer i Cancho, Serge Sharoff, the audience at REPL4NLP 2019 and the anonymous reviewers for helpful comments and suggestions.

References

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Proceedings of NIPS*, pages 2149–2159, Long Beach, CA, 2017.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. Emergent translation in multi-agent communication. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of EMNLP*, pages 2962–2967, Copenhagen, Denmark, 2017.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of EMNLP*, pages 981–985, Brussels, Belgium, 2018.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In *Proceedings of AAMAS*, pages 693–701, Montreal, Canada, 2019.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. `https://arxiv.org/abs/1901.08706`, 2019.

George Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Boston, MA, 1949.

William J Teahan, Yingying Wen, Rodger McNab, and Ian H Witten. A compression-based algorithm for chinese word segmentation. *Computational Linguistics*, 26(3):375–393, 2000.

Bengt Sigurd, Mats Eeg-Olofsson, and Joost Van Weijer. Word length, sentence length and frequency–zipf revisited. *Studia Linguistica*, 58(1):37–52, 2004.

Udo Strauss, Peter Grzybek, and Gabriel Altmann. Word length and word frequency. In Peter Grzybek, editor, *Contributions to the Science of Text and Language*, pages 277–294. Springer, Dordrecht, the Netherlands, 2007.

Thomas Cover and Joy Thomas. *Elements of Information Theory, 2nd ed.* Wiley, Hoboken, NJ, 2006.

Steven T Piantadosi, Harry Tily, and Edward Gibson. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529, 2011.

Kyle Mahowald, Isabelle Dautriche, Edward Gibson, and Steven Piantadosi. Word forms are structured for efficient use. *Cognitive Science*, 42:3116–3134, 2018.

Edward Gibson, Richard Futrell Steven Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. How efficiency shapes human language. *Trends in Cognitive Science*, 2019. In press.

Benoit Mandelbrot. Simpie games of strategy occurring in communication through natural languages. *Transactions of the IRE Professional Group on Information Theory*, 3(3):124–137, 1954.

George A Miller, E Newman, and E Friedman. Some effects of intermittent silence. *American Journal of Psychology*, 70(2):311–314, 1957.

Ramon Ferrer i Cancho and Fermín Moscoso del Prado Martín. Information content versus word length in random typing. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(12): L12002, 2011.

Fermín Moscoso del Prado Martín. The missing baselines in arguments for the optimal efficiency of languages. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35, 2013.

Jasmeen Kanwal, Kenny Smith, Jennifer Culbertson, and Simon Kirby. Zipf's law of abbreviation and the principle of least effort: Language users optimise a miniature lexicon for efficient communication. *Cognition*, 165:45–52, 2017.

Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. Word-order biases in deep-agent emergent communication. In *Proceedings of ACL*, pages 5166–5175, Florence, Italy, 2019.

David Lewis. Convention: A philosophical study, 1969.

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

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Egg: a toolkit for research on emergence of language in games. *arXiv preprint arXiv:1907.00852*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

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

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

Ramon Ferrer i Cancho, Antoni Hernández-Fernández, David Lusseau, Govindasamy Agoramoorthy, Minna J Hsu, and Stuart Semple. Compression as a universal principle of animal behavior. *Cognitive Science*, 37(8):1565–1578, 2013.

Herbert A Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

Raphaela Heesen, Catherine Hobaiter, Ramon Ferrer-i Cancho, and Stuart Semple. Linguistic laws in chimpanzee gestural communication. *Proceedings of the Royal Society B*, 286(1896):20182900, 2019.

Simon Kirby. 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, 2001.

Harald Baayen. *Word Frequency Distributions*. Kluwer, Dordrecht, The Netherlands, 2001.

A.1 Supplementary

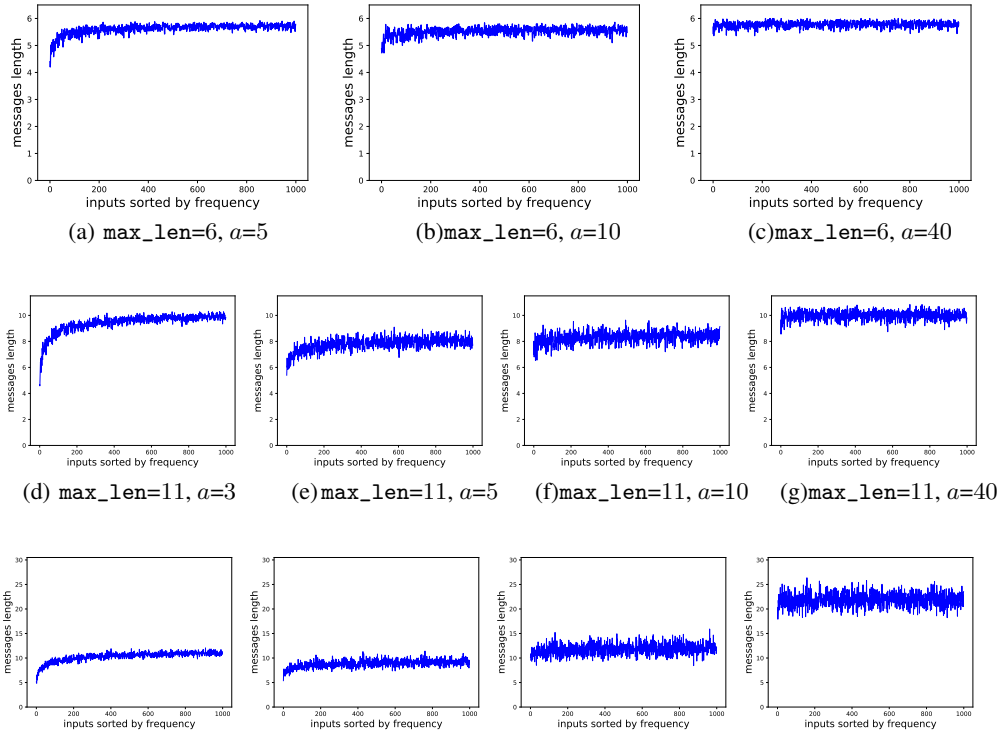
A.1.1 Hyperparameters

Both speaker and listener agents are single-layer LSTMs [Hochreiter and Schmidhuber, 1997]. We experiment with the combinations (Speaker's hidden size, Listener's hidden size) in $[(100, 100), (250, 100), (250, 250), (500, 250)]$. We only experiment with combinations where Speaker's hidden-size is bigger or equal to Listener's, because of the asymmetry in their tasks. Indeed, as discussed in Section 3.1 of the main paper, the Speaker's search space $M_a^{\texttt{max\_len}}$ is generally larger than the one of the Listener R.

We use the Adam optimizer, with learning rate 0.001. We apply entropy regularization to Speaker's optimization. The values of the regularization's coefficient are chosen in $[1, 1.5, 2]$. We run the simulation with each hyperparameter setting 4 times with different random seeds.

A.1.2 Monkey typing

We adapt the Monkey typing (MT) process by adding the `max_len` constraint. This makes it a ZLA-like distribution only when vocabulary size a is small. Figure A1 illustrates this behavior. We see that the higher a is, the further the MT distribution departs from a ZLA pattern.



(a) `max_len`=6, a=5 (b)`max_len`=6, a=10 (c)`max_len`=6, a=40

(d) `max_len`=11, a=3 (e)`max_len`=11, a=5 (f)`max_len`=11, a=10 (g)`max_len`=11, a=40

(h)`max_len`=30, a=3 (i)`max_len`=30, a=5 (j)`max_len`=30, a=10 (k)`max_len`=30, a=40

Figure A1: Monkey typing encoding: Mean message length across 50 simulations as a function of input frequency rank.

A.1.3 Natural language distributions

We report in Figure A2 word length distributions for all the natural languages we considered, and compare them with (1) optimal encoding (OC) and (2) emergent language in the most comparable simulation setting: $(\texttt{max\_len} = 30, a = 40)$. Despite their different alphabet sizes, natural languages pattern similarly: They follow ZLA, and approximate OC.

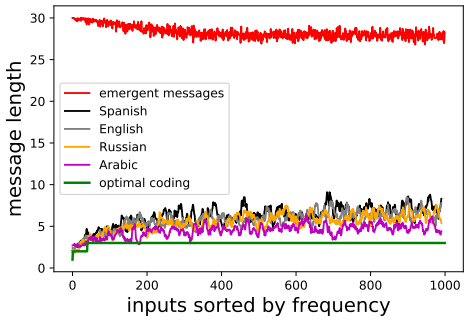


Figure A2: Word length in natural languages in function of word frequency rank, compared to average emergent code and OC in the ($\texttt{max\_len} = 30, a = 40$) setting. For readability, we smooth natural language distributions by reporting the sliding average of 10 consecutive lengths.

A.1.4 Anti-efficient emergent language

Figure A3 shows message length distribution (averaged across all successful runs) in function of input frequency rank, and compares it with some reference distributions. The results are in line with our finding in Section 3.1 of the main paper.

A.1.5 Emergent language with uniform input distribution

Agents' messages are very long also when the input distribution is uniform, see Figure A4. Their average length is significantly larger than MT messages with uniform inputs (t-test, $p < 10^{-9}$).

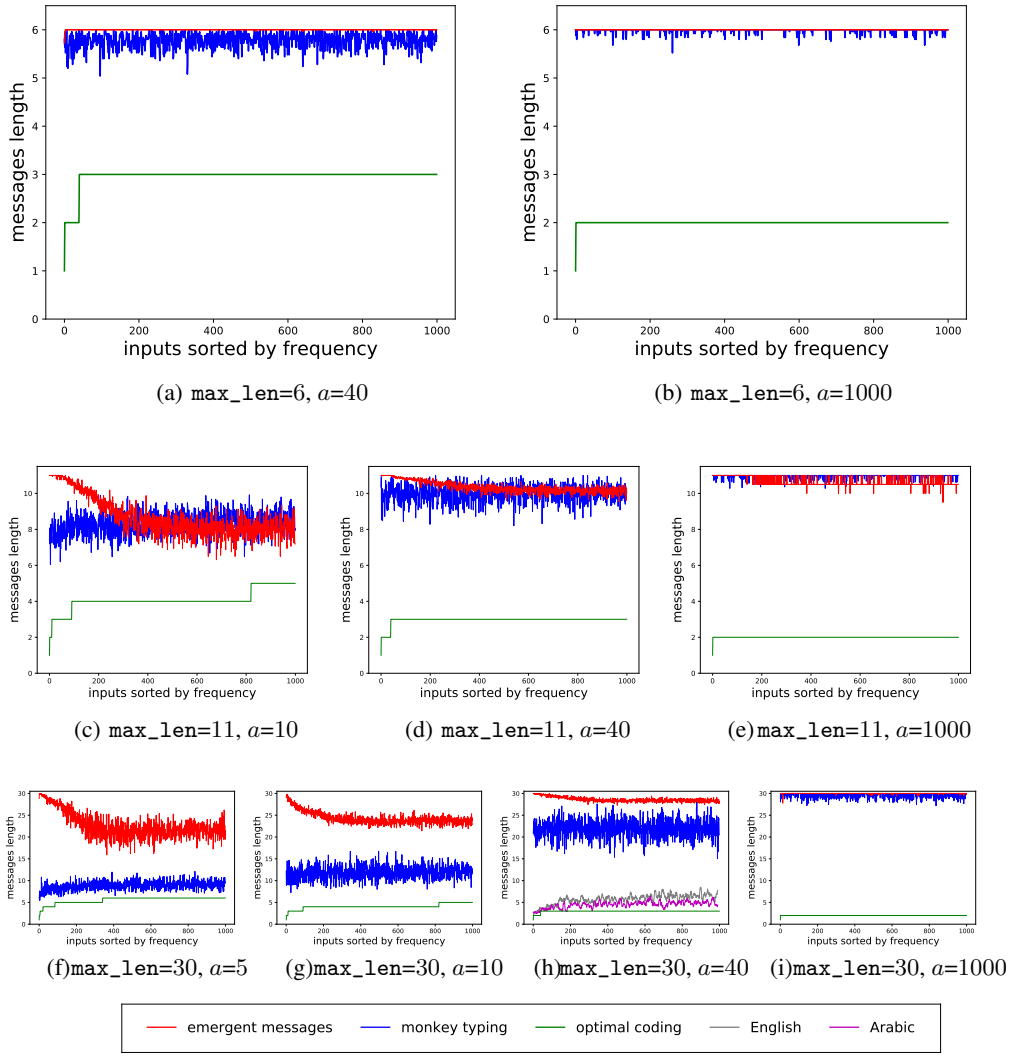
A.1.6 Randomization test

In the main paper, we observe a tendency for Speaker to use longer messages for frequent inputs, making its code obey a sort of "*anti*-ZLA". In this section, we provide quantitative support for this observation. We run the randomization test of Ferrer i Cancho et al. [2013]. We note $E = \sum_{i=1}^{1000} p_i \times l_i$ the mean length of messages, where p_i is the probability of the type i and l_i is the length of the corresponding message. A language that respects ZLA is characterized by a small E (optimal coding, OC, is associated with $min(E)$). Under H_0, the mean length of the encoding coincides with the mean length of a random permutation of messages across types. To be comparable with Ferrer i Cancho et al. [2013], we use the same number of permutations ($= 10^5$). Also, we adopt their definition of "left p-value" and "right p-value". If left p-value≤ 0.005, the studied encoding is *significantly small* (characterized by significantly smaller E than random permutations), if right p-value≤ 0.005, it is *significantly large*, corresponding to our notion of anti-efficiency.

We observe in Table A.1 that H_0 is only rejected for MT with $a \geq 40$, which, as we mentioned in the main paper, approaches a random length distribution for those cases, and for emergent messages with $a = 1000$. OC, natural languages, and emergent language *with* Speaker-length regularization are, in all the considered settings, significantly more efficient than chance. Importantly, the Emergent language results confirm LSTMs' natural preference for long messages (E approaching $\texttt{max\_len}$) and *significant* anti-efficiency for $a \leq 40$ (right p-value≈ 0). When $a = 1000$, there is no frequency rank/length relation and all lengths $\approx \texttt{max\_len}$.

A.1.7 Speaker initial length distribution

Figure A5 plots message length in function of input frequency rank for several settings. In particular, we report *all* settings ($\texttt{max\_len}, a$) that succeeded when training the Speaker-Listener system. Here, however, no training is performed, so that we can observe Speaker's initial biases. The results are in line with our finding in Section 3.2.1 of the main paper.



(a) max_len=6, a=40 (b) max_len=6, a=1000

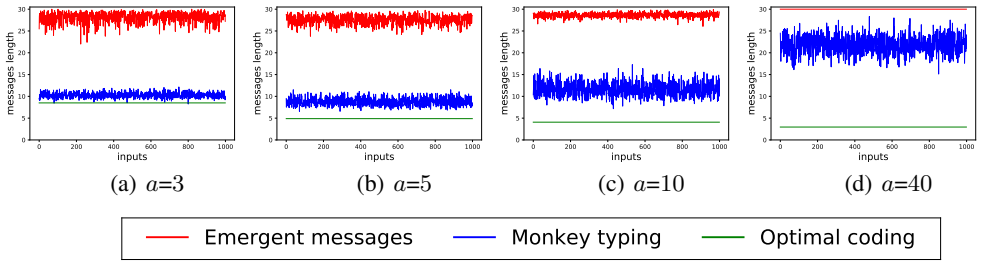
(c) max_len=11, a=10 (d) max_len=11, a=40 (e) max_len=11, a=1000

(f) max_len=30, a=5 (g) max_len=30, a=10 (h) max_len=30, a=40 (i) max_len=30, a=1000

Figure A3: Mean message length across successful runs as a function of input frequency rank, with reference distributions. Natural language distributions are smoothed as in Fig. A2.

A.1.8 The effect of length regularization

We look here at the effect of the regularization coefficient α on the nature of the emergent encoding. To this end, we consider the setting that is least efficient when no optimization is applied: ($\texttt{max\_len} = 30, a = 1000$). The same pattern is also observed with different choices of max_len and a. Figure A6 shows, for $\alpha = 1$, that emergent messages *approximate optimal coding*. For even larger values, we were not able to successfully train the system to communicate. This is in line with Zipf's view of *competing* pressures for accurate communication vs. efficiency. The emergent messages follow ZLA only when both pressures are at work. If the efficiency pressure is not present, agents come up with a communicatively effective but non-efficient encoding, as shown in Section A.1.4 and Section 3.1 of the main paper. However, if the efficiency pressure is too high, agents cannot converge on a protocol that is successful from the point of view of communication.



(a) a=3 (b) a=5 (c) a=10 (d) a=40

Figure A4: Mean message length per input across successful runs for `max_len`=30 and different a. Inputs are uniformly distributed.

Table A.1: Results of the randomization test for different codes when `max_len` $= 30$ and with different alphabet sizes a. Left/right p-values significant at $\alpha = 0.01$ suffixed by asterisk. See Table 1 of Ferrer i Cancho et al. [2013] for more codes to be compared with our results.

Setting	Code	E	Left p-Value	Right p-Value
$a = 5$	OC	3.55	$< 10^{-5}*$	$> 1 - 10^{-5}$
	MT	7.56	$< 10^{-5}*$	$> 1 - 10^{-5}$
	Emergent	26.98	$> 1 - 10^{-5}$	$< 10^{-5}*$
$a = 10$	OC	2.82	$< 10^{-5}*$	$> 1 - 10^{-5}$
	MT	11.27	0.0002*	0.998
	Emergent	26.73	$> 1 - 10^{-5}$	$< 10^{-5}*$
$a = 40$	OC	2.29	$< 10^{-5}*$	$> 1 - 10^{-5}$
	MT	21.30	0.814	0.186
	Emergent	29.40	$> 1 - 10^{-5}$	$< 10^{-5}*$
	Regularized (α=0.5)	7.22	$< 10^{-5}*$	$> 1 - 10^{-5}$
	English	3.68	$< 10^{-5}*$	$> 1 - 10^{-5}$
	Arabic	3.14	$< 10^{-5}*$	$> 1 - 10^{-5}$
$a = 1000$	OC	1.86	0.001*	0.999
	MT	29.67	0.750	0.250
	Emergent	29.98	0.072	0.928

A.2 Repetition in emergent messages

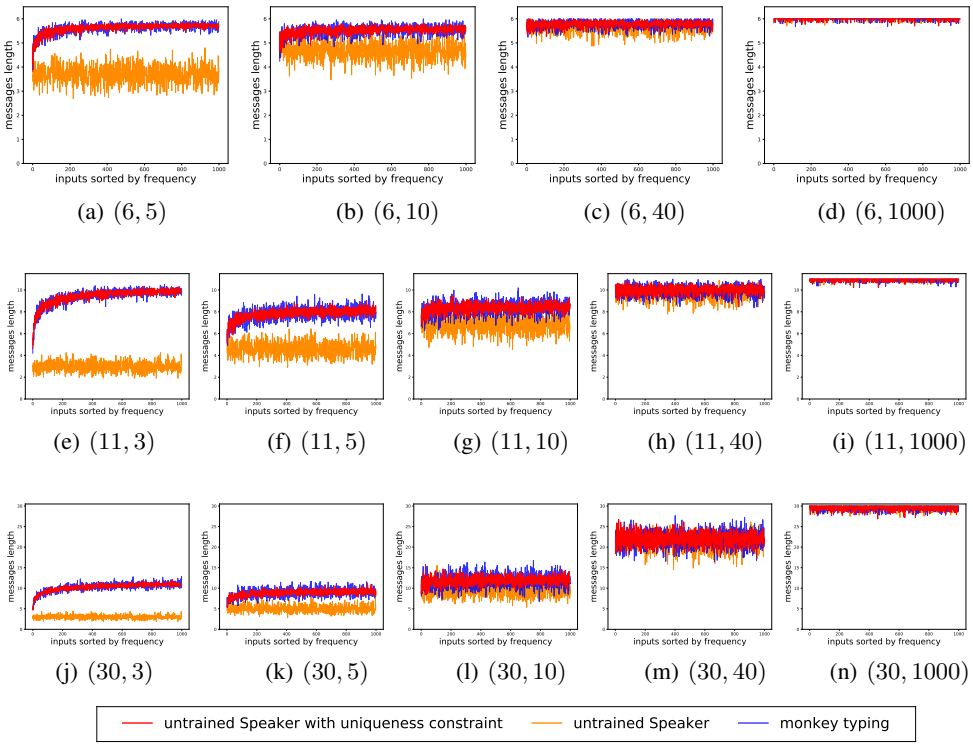
We report in listings 1, 2, 3 and 4 examples of emergent messages in different settings. We notice that the agents extensively use repetition, even when a (vocabulary size) is large. This repetition, that results in the very skewed bigram distributions presented in Section 3.3 of the main paper, increases with higher `max_len`, as shown in figure A7. Moreover, from figure A7, we see that, unlike in emergent codes, this sort of repetition does not appear in natural language.

Listing 1: Emergent messages for the 4 most frequent inputs (`max_len`:11 and a:40).

```
m1: 18,5,36,36,5,5,10,5,32,8,eos
m2: 1,36,2,36,10,13,9,29,33,eos
m3: 29,1,8,1,39,39,9,15,10,19,eos
m4: 29,1,36,36,36,36,5,8,13,9,eos
```

Listing 2: Emergent messages for the 4 most frequent inputs (`max_len`:11 and a:1000).

```
m1: 431,431,305,305,70,70,331,391,134,581,eos
m2: 867,288,466,466,466,737,113,77,615,615,eos
m3: 288,466,466,466,418,144,113,615,638,615, eos
m4: 4,4,152,152,152,468,642,615,422,134,eos
```

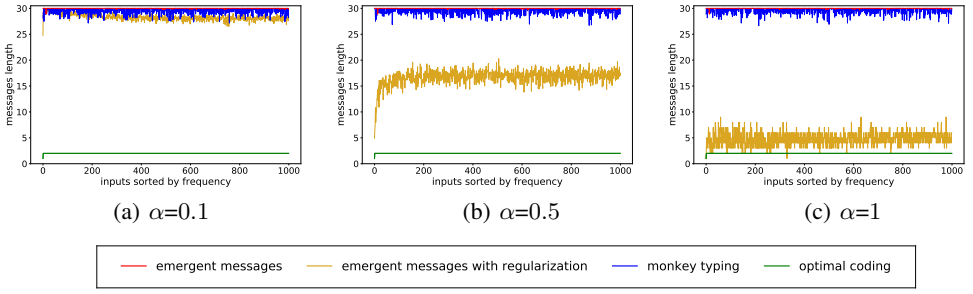


(a) $(6, 5)$ (b) $(6, 10)$ (c) $(6, 40)$ (d) $(6, 1000)$

(e) $(11, 3)$ (f) $(11, 5)$ (g) $(11, 10)$ (h) $(11, 40)$ (i) $(11, 1000)$

(j) $(30, 3)$ (k) $(30, 5)$ (l) $(30, 10)$ (m) $(30, 40)$ (n) $(30, 1000)$

Figure A5: Average length of messages in function of input frequency rank for untrained Speakers, compared to MT. In each figure we report the results in a specific setting $(\texttt{max\_len}, a)$.



(a) α=0.1 (b) α=0.5 (c) α=1

Figure A6: Length of messages as a function of input frequency for $\texttt{max\_len} = 30$ and $a = 1000$, when varying α in the length regularization case.

Listing 3: Emergent messages for the 4 most frequent inputs (`max_len`:30 and a:5).

```
m1: 3,4,4,4,1,1,1,1,1,1,1,1,4,4,4,4,4,4,4,4,4,4,4,4,3,4,3,4,eos
m2: 3,1,3,3,1,1,1,1,1,1,1,4,4,4,4,4,2,4,2,4,2,4,2,4,2,4,3,2,eos
m3: 1,4,4,1,1,1,1,1,1,1,1,4,4,4,4,4,2,4,4,4,4,4,4,4,4,2,4,3,1,eos
m4: 1,4,4,1,1,1,1,1,1,1,1,4,4,4,4,4,4,4,4,4,4,4,4,2,4,2,2,4,1,4,eos
```

Listing 4: Emergent messages for the 4 most frequent inputs (`max_len`:30 and a:40).

```
m1: 11,11,12,24,8,8,12,24,12,12,12,12,12,12,36,24,24,35,35,35,36,36,20,15,36,19,11,31,13, eos
m2: 13,31,31,24,8,8,8,8,8,8,8,8,8,19,24,3,3,36,36,19,29,15,31,30,31,15,19,11,13,eos
m3: 39,8,12,8,8,8,8,25,25,25,25,25,25,25,36,24,12,12,35,35,35,18,18,11,3,7,11,7,11,eos
m4: 14,31,8,8,8,8,8,8,24,25,25,25,36,36,36,36,36,36,36,36,36,3,2,35,30,31,21,29,eos
```

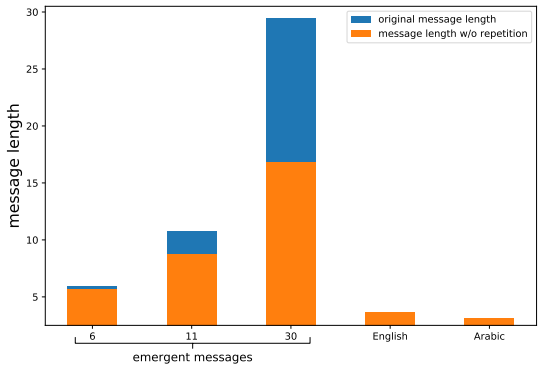
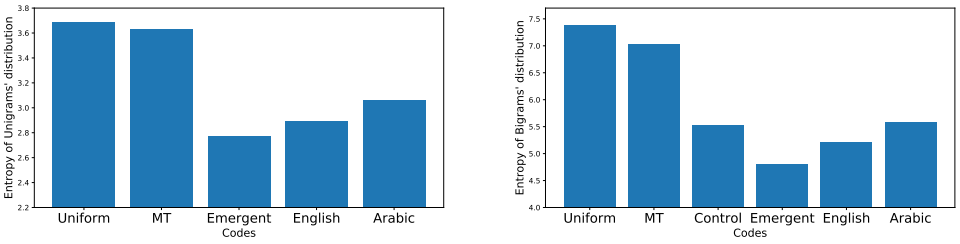


Figure A7: Mean message length (weighted by input probability, and averaged across successful runs) for various `max_len` and fixed $a = 40$, before and after removing all repetitions. A repetition here refers to a sequence of 2 or more consecutive identical symbols. Emergent messages are indexed by their `max_len`, and we add the same statistics in two human languages for comparison.

A.2.1 Entropy of symbol distributions in different codes

We report the entropy of symbol unigram and bigram distributions for different codes in figures 8(a) and 8(b), respectively. We observe that, in both cases, the emergent code symbol distribution is more skewed than in any considered reference code.



(a) Entropy of unigram distributions

(b) Entropy of bigram distributions

Figure A8: Entropy of symbol unigram and bigram distributions for different codes (in natural log). The higher the entropy, the more uniform the corresponding distribution is. The entropy of the uniform code is computed by assuming a uniform distribution over 40 symbols (unigram) and 1600 sequences of 2 symbols (bigram). MT and control messages (see Section 3.3 of main text) are averaged across 25 different simulations in the (`max_len`=30,a=40) setting. Emergent messages are averaged across successful runs in the same setting.

"LazImpa": *Lazy* and *Impatient* neural agents learn to communicate efficiently

Mathieu Rita[1] **Rahma Chaabouni**[1,2] **Emmanuel Dupoux**[1,2]
[1]Cognitive Machine Learning (ENS/EHESS/PSL Research University/CNRS/INRIA)
[2]Facebook AI Research
mathieu.rita@polytechnique.edu, {rchaabouni,dpx}@fb.com

Abstract

Previous work has shown that artificial neural agents naturally develop surprisingly non-efficient codes. This is illustrated by the fact that in a referential game involving a speaker and a listener neural networks optimizing accurate transmission over a discrete channel, the emergent messages fail to achieve an optimal length. Furthermore, frequent messages tend to be longer than infrequent ones, a pattern contrary to the Zipf Law of Abbreviation (ZLA) observed in all natural languages. Here, we show that near-optimal and ZLA-compatible messages can emerge, but only if both the speaker and the listener are modified. We hence introduce a new communication system, "LazImpa", where the speaker is made increasingly *lazy*, i.e., avoids long messages, and the listener *impa*tient, i.e., seeks to guess the intended content as soon as possible.

1 Introduction

Recent emergent-communication studies, renewed by the astonishing success of neural networks, are often motivated by a desire to develop neural network agents eventually able to verbally interact with humans (Havrylov and Titov, 2017; Lazaridou et al., 2017). To facilitate such interaction, neural networks' emergent language should possess many natural-language-like properties. However, it has been shown that, even if these emergent languages lead to successful communication, they often do not bear core properties of natural language (Kottur et al., 2017; Bouchacourt and Baroni, 2018; Lazaridou et al., 2018; Chaabouni et al., 2020).

In this work, we focus on one basic property of natural language that resides on the tendency to use messages that are close to the informational optimum. This is illustrated in the Zipf's law of Abbreviation (ZLA), an empirical law that states that in natural language, the more frequent a word is, the shorter it tends to be (Zipf, 1949; Teahan et al., 2000; Sigurd et al., 2004; Strauss et al., 2007). Crucially, ZLA is considered to be an *efficient* property of our language (Gibson et al., 2019). Besides the obvious fact that an efficient code would be easier to process for us, it is also argued to be a core property of natural language, likely to be correlated with other fundamental aspects of human communication, such as regularity and compositionality (Kirby, 2001). Encouraging it might hence lead to emergent languages that are also more likely to develop these other desirable properties.

Despite the importance of such property, Chaabouni et al. (2019) showed that standard neural network agents, when trained to play a simple signaling game (Lewis, 1969), develop an inefficient code, which even displays an *anti*-ZLA pattern. That is, counterintuitively, more frequent inputs are coded with longer messages than less frequent ones. This inefficiency was related to neural networks' "innate preference" for long messages. In this work, we aim at understanding which constraints need to be introduced on neural network agents in order to overcome their innate preferences and communicate efficiently, showing a proper ZLA pattern.

To this end, we use a reconstruction game where we have two neural network agents: speaker and listener. For each input, the speaker outputs a sequence of symbols (which constitutes the message) sent to the listener. The latter needs then to predict the speaker's input based on the given message. Also, similarly to the previous work, inputs are drawn from a power-law distribution.

We first describe the experimental and optimization framework (see Section 2). In particular, we introduce a new communication system called 'LazImpa', comprising two different constraints (a) *Laz*iness on the speaker side and (b) *Impa*tience on the listener side. The former constraint is inspired by the least-effort principle which is attested to be a ubiquitous pressure in human communication (Piantadosi et al., 2011; Zipf, 1949; Kanwal et al., 2017).

However, if such a constraint is applied too early, the system does not learn an efficient system. We

show that incrementally penalizing long messages in the cost function enables an early exploration of the message space (a kind of 'babbling phase') and prevents converging to an inefficient local minimum.

The other constraint, on the listener side, relies on the prediction mechanism, argued to be important in language comprehension (e.g., Federmeier, 2007; Altmann and Mirković, 2009), and is achieved by allowing the listener to reconstruct the intended input as soon as possible. We also provide a two-level analytical method: first, metrics quantifying the efficiency of a code; second, a new protocol to measure its informativeness (see Section 3). Applying these metrics, we demonstrate that, contrary to the standard speaker/listener agents, our new communication system 'LazImpa' leads to the emergence of an efficient code. The latter follows a *ZLA-like* distribution, close to natural languages (see Sections 4.1 and 4.2). Besides the plausibility of the introduced constraints, our new communication system is, first, task- and architecture-agnostic (requires only communicating with sequences of symbols), and second allows stable optimization of the speaker/listener. We also show how both listener and speaker constraints are fundamental to the emergence of a ZLA-like distribution, as efficient as natural language (see Section 4.3).

2 Experimental framework

We explore the properties of emergent communication in the context of referential games where neural network agents, Speaker and Listener, have to cooperatively communicate in order to win the game.

Speaker network receives an input $i \in \mathcal{I}$ and generates a message m of maximum length `max_len`. The symbols of the message belong to a vocabulary $V = \{s_1, s_2, ..., s_{\texttt{voc\_size}-1}, \texttt{EOS}\}$ of size `voc_size` where `EOS` is the 'end of sentence' token indicating the end of Speaker's message. Listener network receives and consumes the message m. Based on this message, it outputs $\hat{i}$. The two agents are successful if Listener manages to guess the right input (i.e., $\hat{i} = i$).

We make two main assumptions. First inputs are drawn from $\mathcal{I}$ following a power-law distribution, where $\mathcal{I}$ is composed of 1000 one-hot vectors.

Consequently, the probability of sampling the k^{th} most frequent input is: $\frac{1/k}{\sum_{j=1}^{1000} 1/j}$ modelling words' distribution in natural language (Zipf, 2013) (see details in Appendix A.1.1). Second, we experiment in the main paper with `max_len` $= 30$ and `voc_size` $= 40$.[1] We further discuss the influence of these assumptions in Appendix. A.4.2 and show the robustness of our results to assumptions change.

In our analysis, we only consider the successful runs, i.e., the runs with a uniform accuracy strictly higher than 97% over all possible 1000 inputs. An emergent language consists then of the input-message mapping. That is, for each input $i \in \mathcal{I}$ fed to Speaker after successful communication, we note its output m.

By $\mathcal{M}$, we define the set of messages m used by our agents after succeeding in the game.

2.1 Agent architectures

In our experiments, we compare two communication systems:

- Standard Agents: as a baseline, composed of Standard Speaker and Standard Listener;
- 'LazImpa': composed of *Lazy* Speaker and *Impa*tient Listener.

For both Speaker and Listener, we experiment with either standard or modified LSTM architectures (Hochreiter and Schmidhuber, 1997).

2.1.1 Standard Agents

Standard Speaker. Standard Speaker is a single-layer LSTM. First, Speaker's inputs i are mapped by a linear layer into an initial hidden state of Speaker's LSTM cell. Then, the message m is generated symbol by symbol: the current sequence is fed to the LSTM cell that outputs a new hidden state. Next, this hidden state is mapped by a linear layer followed by a softmax to a Categorical distribution over the vocabulary. During the training phase, the next symbol is sampled from this distribution. During the testing phase, the next symbol is deterministically selected by taking the argmax of the distribution.

Standard Listener. Standard Listener is also a single-layer LSTM. Once the message m is generated by Speaker, it is entirely passed to Standard Listener. Standard Listener consumes the symbols one by one, until the `EOS` token is seen (the latter is included and fed to Listener). At the end, the final hidden state is mapped to a Categorical distribution $L(m)$ over the input indices (linear layer + softmax). This distribution is then used during the training to compute the loss. During the testing phase, we take the argmax of the distribution as a reconstruction candidate.

Standard loss $\mathcal{L}_{std}$. For Standard Agents, we merely use the cross-entropy loss between the ground truth one-hot vector i and the output Categorical distribution of Listener $L(m)$.

[1] This combination makes our setting comparable to natural languages; the latter has no upper bound on the maximum length, also a vocabulary size of 40 is close to the alphabet size of the natural languages we study of mean vocabulary size equal to 41.75. See Chaabouni et al. (2019) for more details.

2.1.2 LazImpa

Lazy Speaker. Lazy Speaker has the same architecture as Standard Speaker. The 'Laziness' comes from a cost on the length of the message m directly applied to the loss.

Impatient Listener. We introduce Impatient Listener, designed to guess the intended content as soon as possible. As shown in Figure 1, Impatient Listener consists of a modified Standard Listener that, instead of guessing i after consuming the entire message $m = (m_0, ..., m_t)$, makes a prediction $\hat{i}_k$ for each symbol m_k.[2] This modification takes advantage of the recurrent property of the LSTM, however, could be adapted to any causal sequential neural network model.

At training, a prediction of Impatient Listener, at a position k, is a Categorical distribution $L(m_{:k})$, constructed using a shared single linear layer followed by a softmax (with $m_{:k} = (m_0, ..., m_k)$). Eventually, we get a sequence of $t+1$ distributions $L(m) = (L(m_{:0}), ..., L(m_{:t}))$, one for each reading position of the message.

At test time, we only take the argmax of the distribution generated by Listener when it reads the `EOS` token.

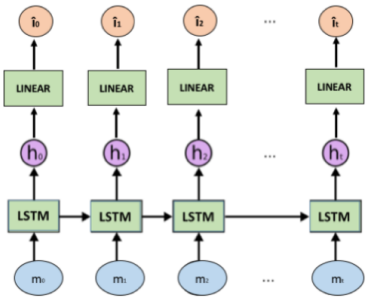


Figure 1: Impatient Listener architecture. The agent is composed of a single-layer LSTM cell and one shared linear layer followed by a softmax. It generates a prediction at each time step.

LazImpa Loss $\mathcal{L}_{laz}$. LazImpa loss is composed of two parts that model 'Impatience' ($\mathcal{L}_{laz/L}$) and 'Laziness' ($\mathcal{L}_{laz/S}$), such that,

$$\mathcal{L}_{laz}(i, m, L(m)) = \mathcal{L}_{laz/L}(i, L(m)) + \mathcal{L}_{laz/S}(m). \quad (1)$$

On one hand, $\mathcal{L}_{laz/L}$ forces Impatient Listener to guess the right candidate as soon as possible when reading the message m. For this purpose, with i the ground-truth input and $L(m) = (L(m_{:0}), ..., L(m_{:t}))$ the sequence of intermediate distributions, the Impatience Loss is defined as the mean cross-entropy loss between i and the intermediate distributions:

$$\mathcal{L}_{laz/L}(i, L(m)) = \frac{1}{t+1} \sum_{k=0}^{t} \mathcal{L}_{std}(i, L(m_{:k})), \quad (2)$$

Hence, all the intermediate distributions contribute to the loss function according to the following principle: the earlier the Listener predicts the correct output, the larger the reward is.

On the other hand, $\mathcal{L}_{laz/S}$ consists of an adaptive penalty on message lengths. The idea is to first let the system explore long and discriminating messages (**exploration step**) and then, once it reaches good enough communication performances, we apply a length cost (**reduction step**). With $|m|$ the length of the message associated with the input i and 'acc' the estimation of the accuracy (proportion of inputs correctly communicated weighted by appearance frequency), the Laziness Loss is defined as:

$$\mathcal{L}_{laz/S}(m) = \alpha(\text{acc})|m| \quad (3)$$

To schedule this two-step training, we model α as shown in Figure 2. The regularization is mainly composed of two branches: (1) exploration step and (2) reduction step. The latter starts only when the two agents become successful.

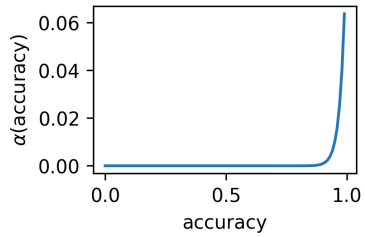


Figure 2: Scheduling of the regularization parameter α as a function of the accuracy. We distinguish two different regimes: the exploration and the reduction regimes. See the mathematical description in Appendix A.1.4

2.2 Optimization

The overall setting, which can be seen as a discrete auto-encoder, cannot be differentiated directly, as the latent space is discrete. We use a hybrid optimization between REINFORCE for Speaker (Williams, 1992) and classic back-propagation for Listener (Schulman et al., 2015).

With $\mathcal{L}$ the loss of the system, i the ground-truth input and $L(m)$ the output distribution of Listener that takes the message m as input, the training task consists in minimizing the expectation of the loss $\mathbb{E}[\mathcal{L}(i, L(m))]$. The expectation is

[2] m_t=`EOS` by construction.

computed w.r.t the joint distribution of the inputs and the message sequences. Let's denote θ_L and θ_S Listener and Speaker parameters respectively. The optimization task requires to compute the gradient $\nabla_{\theta_S \cup \theta_L} \mathbb{E}[\mathcal{L}(i, L(m))]$. An unbiased estimate of this gradient is the gradient of the following function:

$$\mathbb{E}[\underbrace{\mathcal{L}(i, L(m; \theta_L))}_{(A)} + \underbrace{(\{\mathcal{L}(i, L(m; \theta_L))\} - b) \log P_S(m|\theta_S)}_{(B)}], \tag{4}$$

where $\{.\}$ is the stop-gradient operation, $P_S(m|\theta_S)$ the probability that Speaker generates the message m, b the running-mean baseline used to reduce variance (Williams, 1992). We also promote exploration by encouraging Speaker's entropy (Williams and Peng, 1991).

The gradient of (4) w.r.t θ_L is found via conventional back-propagation (A) while gradient w.r.t θ_S is found with a REINFORCE-like procedure estimating the gradient via a Monte-Carlo integration calculated over samples of the messages (B). Once the gradient is estimated, it is eventually passed to the Adam optimizer (Kingma and Ba, 2014).

In Appendix A.3.1, we show that LazImpa leads to a stable convergence. We use the EGG toolkit (Kharitonov et al., 2019) as a starting framework. For reproducibility, the code can be found at `https://github.com/MathieuRita/Lazimpa` and the set of hyper-parameters used is presented in Appendix A.1.

3 Analytical method

As ZLA is defined informally, we first introduce reference distributions for comparison. Then, we propose some simple metrics to evaluate the overall efficiency of our emergent codes. Eventually, we provide a simple protocol to analyze the distribution of information within the messages.

3.1 Reference distributions

We compare the emergent languages to the reference distributions introduced in Chaabouni et al. (2019). We provide below a brief description of the different distributions, however, we invite readers to refer to the reference paper for more details.

Optimal Coding (Cover and Thomas, 2006) guarantees the shortest average message length with `max_len` = 30 and `voc_size` = 40. To do so, we deterministically associate the shortest messages to the most frequent inputs. See Ferrer i Cancho et al. (2013) for more details about the derivation of Optimal Coding.

Natural Language We also compare emergent languages with several human languages. In particular, we consider the same languages of the reference paper (English, Arabic, Russian, and Spanish). These references consist of the mapping from the frequency of the top 1000 most frequent words in each language to their length (approximated by the number of characters of each word).[3]

3.2 Efficiency metrics

In this work, we examine the constraints needed for neural agents to develop efficient languages. We use three metrics to evaluate how efficient the different codes are.
For all metrics, N denotes the total number of messages (=1000) and $l(m)$ the length of a message m.

Mean message length L_{type}: measures the mean length of the messages assuming a *uniform* weight for each input/message:

$$L_{type} = \frac{1}{N} \sum_{m \in \mathcal{M}} l(m), \tag{5}$$

Mean weighted message length L_{token} : measures the average length of the messages weighted by their generation frequency:

$$L_{token} = \sum_{m \in \mathcal{M}} p(m) l(m), \tag{6}$$

where $p(m)$ is the probability of message m (equal to the probability of input i denoted by m) such that $\sum_{m \in \mathcal{M}} p(m) = 1$. Formally, the message m referring to the k^{th} most frequent input would have a probability $\frac{1/k}{\sum_1^{1000} 1/j}$.
Note that, the Optimal Coding is the one that minimizes L_{token} (Cover and Thomas, 2006; Ferrer i Cancho et al., 2013).

ZLA significance score p_{ZLA}: Let's note $(l_i)_{i \in \mathcal{I}}$ a distribution of message lengths of a code. As a ZLA distribution is the one that minimizes L_{token}, we can check if $(l_i)_{i \in \mathcal{I}}$ follows ZLA by testing if its L_{token} is lower than any random permutation of its frequency-length mapping. This is the idea of the randomization test proposed by Ferrer i Cancho et al. (2013).

The test checks whether L_{token} coincides with $\sum_{i \in \mathcal{I}} l_i f_{\sigma(i)}$, with $\sigma(i)$ a random permutation of inputs. We can eventually compute a p-value p_{ZLA} (at threshold α) that measures to which extent L_{token} is likely to be smaller than any other weighted mean message length of a frequency-length mapping. $p_{ZLA} < \alpha$ indicates that any random permutation would have most likely longer weighted mean length. Thus $(l_i)_{i \in \mathcal{I}}$ follows *significantly* a ZLA distribution. Additional details are provided in Appendix A.3.2.

3.3 Information analysis

We also provide an analytical protocol to evaluate how information is distributed within the

[3] We use the frequency lists from `http://corpus.leeds.ac.uk/serge/`.

messages. We consider a symbol to be informative if replacing it randomly has an effect on Listener's prediction. Formally, let's take the message $m = (m_0, ..., m_t)$ associated to the ground truth input i after training. To evaluate the information contained in the symbol at position k, m_k, we substitute it randomly by drawing another symbol r_k uniformly from the vocabulary (except the `EOS` token). Then, we feed this new message $\tilde{m} = (m_1, ..., r_k, ..., m_t)$ into Listener that outputs $\tilde{o}_{m,k}$ (index m indicates that the original message was m, index k indicates that the k^{th} symbol of the original message has been replaced). We define $\Lambda_{m,k}$ a boolean score that evaluates whether the symbol replaced at position k has an impact on the prediction, such that $\Lambda_{k,m} = \mathbf{1}(\tilde{o}_{m,k} \neq i)$. If $\Lambda_{m,k} = 1$, the k^{th} symbol of message m is considered as informative. If $\Lambda_{m,k} = 0$, it is considered as non-informative. We do not consider misreconstructed inputs, neither the position t, as m_t=`EOS`.[4] This token is needed for Listener's prediction at test time.

This test allows us to introduce some variables that quantify to which extent information is effectively distributed within the messages. As previously, we note $l(m)$ the length of message m and N the total number of messages.

Positional encoding $(\Lambda_{.,k})_{1 \leq k \leq \texttt{max\_len}}$: analyzes the position of informative symbols within an emergent code. We assign a score $\Lambda_{.,k}$ for each position k that counts the proportion of informative symbols over all the messages of a language:

$$\Lambda_{.,k} = \frac{1}{N(k)} \sum_{m \in \mathcal{M}} \Lambda_{m,k}, \tag{7}$$

where $N(k)$ is the number of messages that have a symbol (different from `EOS`) at position k.

Effective length L_{eff}: measures the mean number of informative symbols by message:

$$L_{eff} = \frac{1}{N} \sum_{m \in \mathcal{M}} \sum_{k=1}^{l(m)-1} \Lambda_{m,k}. \tag{8}$$

L_{eff} counts the average number of symbols Listener relies on (removing all the uninformative symbols for which $\Lambda_{m,k} = 0$). A message with only informative symbols would have $L_{eff} = L_{type} - 1$.[5]

Information density ρ_{inf} : measures the fraction of informative symbols in a language:

$$\rho_{inf} = \frac{1}{N} \sum_{m \in \mathcal{M}} \frac{1}{l(m) - 1} \sum_{k=1}^{l(m)-1} \Lambda_{m,k}. \tag{9}$$

We integrate over the first $l(m) - 1$ positions as we disregard `EOS` that occurs in all messages.[6] $0 \leq \rho_{inf} \leq 1$. If $\rho_{inf} = 1$, messages are limited to the informative symbols (all used by Listener to decode the message). The lower ρ_{inf} is, the more non-informative symbols are in the message.

As we do not have Listener when generating Optimal Coding, we compute these metrics for the latter reference by considering all symbols, but `EOS`, informative.

4 Results

In this section, we study the code of our new communicative system, LazImpa, and compare it to the Standard Agents baseline and the different reference distributions. We show that LazImpa leads to near-optimal and ZLA-compatible languages. Eventually, we demonstrate how both Impatience and Laziness are required to get human-level efficiency. All the quantitative results of the considered codes are gathered in Table 1.

4.1 LazImpa vs. Standard Agents

We compare here LazImpa to the baseline system Standard Agents both in terms of the length efficiency and the allocation of information.

Length efficiency of the communication. Contrary to Standard Agents, LazImpa develops an efficient communication as presented in Figure 3. Indeed, its average length of the messages is significantly lower than the Standard Agents system (average L_{type}=29.6 for Standard Agents vs. L_{type}=5.49 for LazImpa). The latter demonstrates length distributions almost constant and close to the maximum length we set (=30). We demonstrate in Appendix A.2.1 how the exploration of long messages in Standard Agents is key for agents' success in the reconstruction game, even though, in theory, shorter messages are sufficient.

Interestingly, both systems do not only differ by their average length, but also by the distribution of messages length. Specifically, the Standard Agents system follows significantly an anti-ZLA distribution (see Appendix A.3.2 for quantitative support of this claim) while LazImpa has an average L_{token}=3.78 showing a ZLA pattern: the shortest messages are associated to the most frequent inputs. The randomization test gives quantitative support of this observation ($p_{ZLA} < 10^{-5}$).

Informativeness of the communication. When considering how Standard Agents system allocates information, shown in Figure 4a, we can make two striking observations. First, only a very small part of the messages are informative (on average $\rho_{inf} = 11\%$). Therefore, even if long messages

[4] As we only consider successful runs, more than 97% of inputs are, by definition, well-reconstructed.

[5] We subtract 1 as we disregard `EOS` in all messages.

[6] By convention, for the case where m=(`EOS`), $\frac{0}{0}$=1.

Class	Code	L_{type}	L_{token}	p_{ZLA}	L_{eff}	ρ_{inf}
Emergent	Standard Agents	29.6 ± 0.4	29.91 ± 0.07	$> 1 - 10^{-5}$	3.33 ± 0.46	0.11 ± 0.02
	LazImpa	5.49 ± 0.67	3.78 ± 0.34	$< 10^{-5}$*	2.67 ± 0.07	0.60 ± 0.07
References	Mean natural languages	5.46 ± 0.61	3.55 ± 0.14	$< 10^{-5}$*	/	/
	Optimal Coding	2.96	2.29	$< 10^{-5}$*	1.96	1.00

Table 1: Efficiency and information analysis of emergent codes and reference distribution. For each metric, we report the mean value and the standard deviation when relevant (across seeds when experimenting with emergent languages and across the natural languages presented in Section 3.1 for Mean natural languages). L_{type} is the mean message length, L_{token} is the mean weighted message length, p_{ZLA} the ZLA significance score, L_{eff} the effective length and ρ_{inf} the information density. '/' indicates that the metric cannot be computed. For p_{ZLA}, '*' indicates that the p-value is significant (< 0.001).

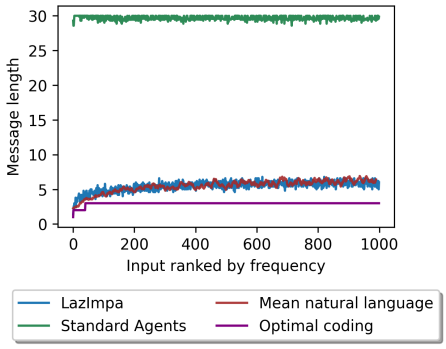


Figure 3: Average message length across successful runs as a function of input frequency rank.

seem necessary for the agents to succeed, most of the symbols are not used by Listener. In particular, if $L_{type} = 29.6$ on average, the average number of symbols used by Standard Listener (L_{eff}) is only equal to 3.33 (which is even smaller than natural languages' mean message length $L_{type} = 5.46$). Surprisingly, we also observe that, if we restrict the messages to their informative symbols (i.e. removing positions k with $\Lambda_{k,.} = 0$), the length statistics follow a ZLA-like distribution (see Figure 9 in Appendix A.2.2). Second, in all our experiments, the information is localized at the very end of the messages. That is, there is almost no actual information in the messages about Speaker's inputs before the last symbols.

Contrarily, Figure 4d shows a completely different spectrum for LazImpa. Indeed, Impatient Listener relies on $\rho_{inf} = 60\%$ of the symbols. This corresponds to a big increase compared to $\rho_{inf} = 19\%$ when using Standard Agents. Yet, we are still far from the 100% observed in Optimal Coding. That is, even with the introduction of a length cost (with Lazy Speaker), we still encounter non-informative symbols. Finally, these informative symbols are localized in the first positions, opposite to what we observed with Standard Agents. We will show in Section 4.3 how this immediate presence of information is crucial for the length reduction of the messages.

In sum, if we consider only *informative/effective* positions, Standard Agents use efficient and ZLA-like (effective) communicative protocol. However, they make it maximally long adding non-informative symbols at the beginning of each message. Introducing LazImpa reverses the length distribution. Indeed, we observe with LazImpa the emergence of efficient and ZLA-obeying languages, with significantly larger ρ_{inf}.

4.2 LazImpa vs. reference distributions

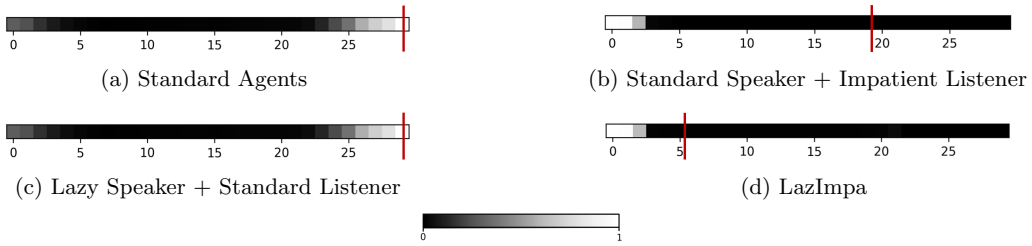
We demonstrated above how LazImpa leads to codes with length significantly shorter than the one obtained with Standard Agents.

We compare it here with stricter references, namely natural languages and Optimal Coding. We show that LazImpa results in languages as efficient as natural languages both in terms of length statistics and symbols distribution. However, agents do not manage to reach optimality.

Comparison with natural languages. We see in Figure 5a that the message lengths in the emergent communication are analogous to the words lengths in natural languages: close average L_{token} and L_{type} (see Table 1).

We further compare their unigram distributions. Chaabouni et al. (2019) showed that Standard Agents develop repetitive messages with a skewed unigram distribution. Our results, in Figure 5b, show that, on top of a ZLA-like code, LazImpa enables the emergence of natural-language-like unigram distribution, without any particular repetitive pattern. Intriguingly, this similarity with natural languages is an unexpected property as a uniform distribution of unigrams would lead to a more efficient protocol.

Comparison with Optimal Coding. If LazImpa leads to significantly more efficient languages compared to Standard Agents, these emergent languages are still not as efficient as Optimal Coding (see Figure 3). One obvious source of sub-optimality is the addition of uninformative sym-



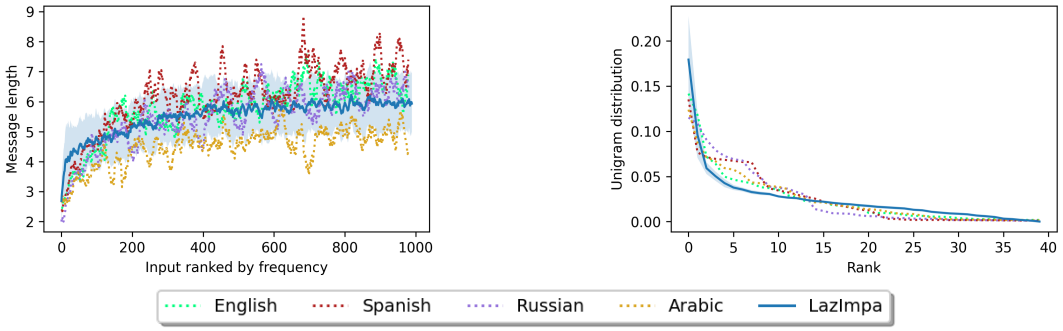
(a) Standard Agents

(b) Standard Speaker + Impatient Listener

(c) Lazy Speaker + Standard Listener

(d) LazImpa

Figure 4: Fraction of informative symbols at each position k averaged across all emergent messages of successful runs ($(\Lambda_{k,.})_{0\leq k\leq 29}$). Each box represents the proportion of informative symbols at a given position $\Lambda_{k,.}$ mapped to a color according to a gray gradient (black=0 ; white=1). The red vertical lines mark the mean message length L_{type} across successful runs.



(a) Message length of natural languages and LazImpa (averaged across successful runs) as a function of input frequency rank. For readability, the curves have been smoothed using a sliding average of 20 consecutive lengths, see the real curves in Appendix A.4.3. The light blue interval shows 1 standard deviation for LazImpa's distribution.

(b) Unigrams distribution of natural languages and LazImpa (averaged across successful messages) ranked by unigram frequency. The light blue interval shows 1 standard deviation for LazImpa's unigrams distribution.

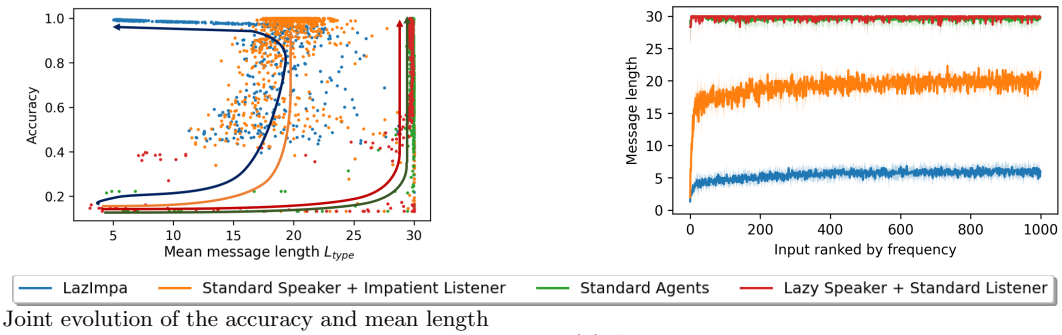
Figure 5: Comparison of LazImpa' statistics and natural languages.

bols at the end of the messages (i.e. the difference between L_{eff}=2.67 and L_{type}-1=4.49). Interestingly, when analyzing the intermediate predictions of Impatient Listener, we see that this model is actually able to guess the right input only reading approximately the L_{eff} first positions (see Appendix A.4.1 for details). However, we still can note that the informative length L_{eff} is slightly sub-optimal (L_{eff} = 2.67 for LazImpa, L_{eff} = 1.96 for Optimal Coding). This difference can be explained by the non-uniform use of unigrams. Specifically, we show in Appendix A.4.1 that effective lengths of LazImpa messages approximate Optimal Coding when the latter uses the same skewed unigram distribution.

4.3 Ablation study

We have just seen that our new communication system LazImpa allows agents to develop an efficient and ZLA-obeying language whose statistical properties are close to those of natural languages. In this section, we analyze the effects of the modeling choices we have made.

We first look at the effect of Laziness. To do so, we compare LazImpa to the system "Standard Speaker + Impatient Listener" (i.e. removing the length regularization). Figure 6a shows the joint evolution of the mean length of messages (L_{type}) and game accuracy. We observe that our non-regularized system, similarly to LazImpa, initially explores long messages while being more successful (exploration step). Surprisingly, even in the absence of Laziness, the exploration step does not continue to maximally long messages, as it is the case for Standard Agents, but breaks at length $\approx$ 20. However, *contrary to LazImpa*, "Standard Speaker + Impatient Listener" does not present a reduction step (a reduction of mean length for a fixed good accuracy). Thus, as expected, the introduction of Laziness in LazImpa is responsible for the reduction step, and hence for a shorter and more efficient communication protocol. However, we note in Figure 6b, that Impatience alone is sufficient for the emergence of ZLA. Moreover, when looking at the information spectrum, comparing "Standard Speaker + Impatient Listener" (Figure 4b) to LazImpa (Figure 4d), we observe how alike both systems allocate information and differ only by their mean length.



(a) Joint evolution of the accuracy and mean length for the different models. Each point shows the couple (L_{type},accuracy) of one training episode. Arrows represent the average joint evolution of the two variables.

(b) Average message length as a function of input frequency rank for the different systems. Light color intervals show 1 standard deviation.

Figure 6: Comparison of different communication systems.

Second, we investigate the role of Impatience. We see in Figure 6a that the system "Lazy Speaker + Standard Listener" admits a visually different dynamic compared to LazImpa. In particular, the exploration step leads to significantly longer messages, close to `max_len`. Interestingly, if we demonstrated above the necessity of Laziness for the reduction step, alone, it does not induce it: no reduction step in the "Lazy Speaker + Standard Listener" system is observed. This is due to the necessity of long messages when experimenting with Standard Listener. Specifically, as informative symbols are present only at the last positions (see Figure 4c), introducing a length regularization provokes a drop in accuracy, which in turn cancels the regularization. In other words, the length regularization scheduling stops at the exploration step, which makes the system almost equivalent to Standard Agents (this could be also seen experimentally in Figures 6a and 6b).

Taken together, our analysis emphasizes the importance of both Impatience *and* Laziness for the emergence of efficient communication.

5 Conclusion

We demonstrated that a standard communication system, where standard Speaker and Listener LSTMs are trained to solve a simple reconstruction game, leads to long messages, close to the maximal threshold. Surprisingly, if these messages are long, LSTM agents rely only on a small number of informative message symbols, located at the end. We then introduce LazImpa, a constrained system that consists of *Lazy* Speaker and *Impa*tient Listener. On the one hand, Lazy Speaker is obtained by introducing a cost on messages length once the communication is successful. We found that early exploration of potentially long messages is crucial for successful convergence (similar to the exploration in RL settings). On the other hand, Impatient Listener aims to succeed at the game as soon as possible, by predicting Speaker's input at each message's symbol.

We show that both constraints are *necessary* for the emergence of a ZLA-like protocol, as efficient as natural languages. Specifically, Lazy Speaker alone would fail to shorten the messages. We connect this to the importance of the Impatience mechanism to locate useful information at the beginning of the messages. If the function of this mechanism is subject to a standing debate (e.g., Jackendoff, 2007; Anderson and Chemero, 2013), many prior works had pointed to its necessity to human language understanding (e.g., Friston, 2010; Clark, 2013). We augment this line of works and suggest that impatience could be at play in the emergence of ZLA-obeying languages. However, if impatience leads to ZLA, it is not sufficient for human-level efficiency. In other words, efficiency needs constraints *both* on Speaker and Listener sides.

Our work highlights the importance of introducing the right pressures in the communication system. Indeed, to construct automated agents that would eventually interact with humans, we need to introduce task-agnostic constraints, allowing the emergence of more human-like communication. Moreover, while being general, LazImpa provides a more stable optimization compared to the unconstrained system. Finally, this study opens several lines of research. One would be to investigate further the gap from optimality. Indeed, while LazImpa emergent languages show human-level efficiency, they do not reach optimal coding. Specifically, emergent languages still have non-informative symbols at the end of the messages. If these additional non-useful symbols drift the protocol from optimality, we encounter similar trend in human (Marslen-Wilson, 1987) and animal communication (McLachlan and Magrath, 2020). We leave the understanding of the role of these non-informative symbols and how we can reach optimal coding for future works. A second

line of research would be to apply this system to other games or NLP problems and study how it affects other properties of the language such as regularity or compositionality.

Acknowledgments

We would like to thank Emmanuel Chemla, Marco Baroni, Eugene Kharitonov, and the anonymous reviewers for helpful comments and suggestions.

This work was funded in part by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-17-EURE-0017 Frontcog, ANR-10-IDEX-0001-02 PSL*, ANR-19-P3IA-0001 PRAIRIE 3IA Institute) and grants from CIFAR (Learning in Machines and Brains), Facebook AI Research (Research Grant), Google (Faculty Research Award), Microsoft Research (Azure Credits and Grant), and Amazon Web Service (AWS Research Credits).

References

Gerry TM Altmann and Jelena Mirković. 2009. Incrementality and prediction in human sentence processing. *Cognitive science*, 33(4):583–609.

Michael L Anderson and Tony Chemero. 2013. The problem with brain guts: Conflation of different senses of" prediction" threatens metaphysical disaster. *Behavioral and Brain Sciences*, 36(3):204.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. pages 981–985.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124*.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019. Anti-efficient encoding in emergent communication.

Andy Clark. 2013. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.

Thomas Cover and Joy Thomas. 2006. *Elements of Information Theory, 2nd ed.* Wiley, Hoboken, NJ.

Kara D Federmeier. 2007. Thinking ahead: The role and roots of prediction in language comprehension. *Psychophysiology*, 44(4):491–505.

Ramon Ferrer i Cancho, Antoni Hernández-Fernández, David Lusseau, Govindasamy Agoramoorthy, Minna Hsu, and Stuart Semple. 2013. Compression as a universal principle of animal behavior. *Cognitive Science*, 37(8):1565–1578.

Karl Friston. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138.

Edward Gibson, Richard Futrell, Steven P Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in cognitive sciences*, 23(5):389–407.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. pages 2149–2159.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ray Jackendoff. 2007. A parallel architecture perspective on language processing. *Brain research*, 1146:2–22.

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. Copyright © 2017 Elsevier B.V. All rights reserved.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on emergence of language in games. In *Proceedings of EMNLP (System Demonstrations)*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

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.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

David Lewis. 1969. *Convention.* Harvard University Press, Cambridge, MA.

William D Marslen-Wilson. 1987. Functional parallelism in spoken word-recognition. *Cognition*, 25(1-2):71–102.

Jessica R McLachlan and Robert D Magrath. 2020. Speedy revelations: how alarm calls can convey rapid, reliable information about urgent danger. *Proceedings of the Royal Society B*, 287(1921):20192772.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. 2015. Gradient estimation using stochastic computation graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, page 3528–3536, Cambridge, MA, USA. MIT Press.

Bengt Sigurd, Mats Eeg-Olofsson, and Joost Van Weijer. 2004. Word length, sentence length and frequency–zipf revisited. *Studia Linguistica*, 58(1):37–52.

Udo Strauss, Peter Grzybek, and Gabriel Altmann. 2007. Word length and word frequency. In *Contributions to the science of text and language*, pages 277–294. Springer.

William J Teahan, Yingying Wen, Rodger McNab, and Ian H Witten. 2000. A compression-based algorithm for chinese word segmentation. *Computational Linguistics*, 26(3):375–393.

Ronald Williams and Jing Peng. 1991. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3:241–.

Ronald J. Williams. 1992. Simple statistical gradientfollowing algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

George Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Boston, MA.

George Kingsley Zipf. 2013. *The psycho-biology of language: An introduction to dynamic philology*. Routledge.

A Appendix

A.1 Experimental settings

A.1.1 Input space

The input space $\mathcal{I}$ is composed of 1000 one-hot vectors. Each of them has to be communicated by Speaker to Listener. In order to fit the distribution of words in natural languages, the inputs are fed from a power-law distribution. Indeed, as demonstrated in Figure 7, distribution of words in natural languages follow power-laws with exponents k between -0.79 (Arabic) and -0.96 (Russian). In our experiment, we choose $k = -1$.

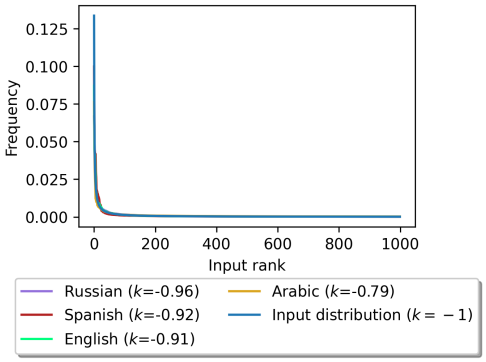


Figure 7: Comparison between the input distribution of our artificial environment and the distribution of the 1000 most frequent words in different natural languages (the coefficient k refers to the coefficient of the power-law for each language when fitted by a linear regression).

A.1.2 Agents

In all our experiments, we fix the architecture of the agents. Speaker is a 1-layer LSTM (Hochreiter and Schmidhuber, 1997) with a hidden size equal to 100. Listener is also a 1-layer LSTM with a hidden size equal to 600.

A.1.3 Optimization

For the training, we use the Adam optimizer (Kingma and Ba, 2014) with a learning rate equal to 0.001. We train the agents for 1500 epochs. During one episode, the system is fed with 100 batches of 512 inputs sampled with replacement from the power-law distribution. In addition, we enforce exploration with an entropy regularization coefficient equal to 2 (Williams and Peng, 1991).

To ensure the robustness of our results, we ran the experiments with 6 different random seeds. All the experiments have been successful, i.e. they reach an accuracy of 99%. This accuracy is weighted by the frequency of inputs. On average, more than 97.5% of inputs are well communicated.

A.1.4 Adaptive regularization coefficient

As defined in the main paper, the adaptive regularization coefficient is scheduled as a function of the accuracy in order to have the following two-step scheme:

- **Exploration step**: during the first part of the training (low accuracy), the regularization coefficient is almost null
- **Reduction step**: Once the communication becomes successful (high accuracy), we start introducing a regularization.

A fair equation to model this two-step scheme is:

$$\alpha(\text{accuracy}) = \frac{\text{accuracy}^{\beta_1}}{\beta_2} \tag{10}$$

where $(\beta_1, \beta_2) \in \mathbb{R}^2$ is a new couple of hyper-parameters. Intuitively, the two parameters allow to control (a) the threshold from which the regularization becomes effective (with β_1) and (b) the intensity of the regularization (with β_2). In our experiments, we introduce a late regularization choosing: $\beta_1 = 45$. We set $\beta_2 = 10$ in order to enables the system to reach an accuracy close to 1. Note that other regularization scheduling can be applied. The only requirement is that the agents successfully communicate before the start of the reduction step.

A.2 Characterization of the emergent communication with Standard Agents

In this section, we report complements about the characterization of the emergent communication with Standard Agents.

A.2.1 Quick use of long messages

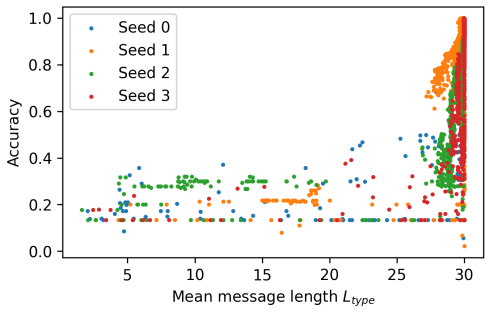


Figure 8: Accuracy as a function of the mean length for 4 different seeds. Each point represents a couple (accuracy, mean length).

To bring more insights about the length inefficiency observed in the main paper, we characterize each episode by the couple accuracy (i.e. the proportion of inputs correctly communicated by

the agents weighted by the frequency of appearance) and mean length (i.e. the average length of the messages generated by the Speaker).

During the training time, we analyze how this couple evolves. The results with four randomly selected seeds are shown in Figure 8. As we can see, at the beginning of the learning process (low accuracies), both the mean length of the messages and the accuracy are quite low (the lowest accuracy value 0.13 corresponds to the good prediction of the most frequent input). Then, the mean message length is increasing without a strong effect on the accuracy. It is only when the agents start to use long messages (higher than 25 for a maximum length of 30) that the communication becomes successful. Therefore, we see that exploration of long messages seems key for the agents to reach high accuracies.

A.2.2 Efficient *informative* symbols

We analyze the statistical properties of the informative parts of the messages that emerge from Standard Agents. As defined in the main paper, we consider a symbol informative if it is used by Listener for the reconstruction. We remove all the non-informative symbols from the messages (i.e. positions k with $\Lambda_{k,.} = 0$). In Figure 9, we plot the length of informative parts of messages associated to inputs ranked by frequency (average distribution over the different runs). We compare it to the average words length distribution of natural languages and to Optimal Coding. As we can see in the figure, even though Standard Agents produce an inefficient code (as seen in the main paper) the length statistic of the informative parts is close to Optimal Coding. Interestingly, we even note an emergent code more efficient than natural languages. In addition, even if no constraint is applied on informative parts, we observe that it follows ZLA.

A.3 Comparing communication systems

A.3.1 Convergence

We check here the convergence and robustness of our introduced communication system, LazImpa. As a preliminary analysis, we compare the convergence results of: Standard Agents, (Standard Speaker + Impatient Listener), (Lazy Speaker + Standard Listener) and LazImpa. In Figure 10, we show the accuracy as a function of the training episodes for 3 randomly selected seeds. We see that the convergence dynamic is sensitive to the initialization but that in the end, the three systems converge.

Moreover, we observe a gain of stability for the systems with the Impatient Listener. Indeed, as shown in Figure 10, Standard Agents demonstrate a less smooth accuracy curve compared to both

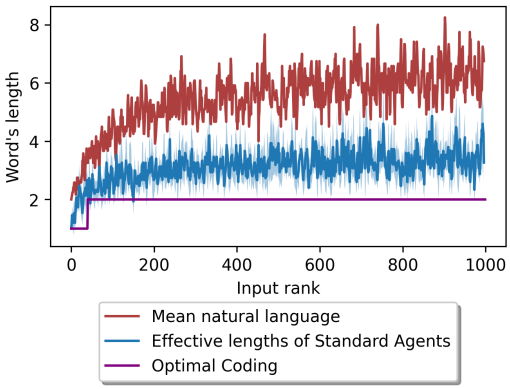


Figure 9: Average length distribution of informative parts in Standard Agents code compared to the mean words distribution of natural languages and Optimal Coding. The light blue interval shows 1 standard deviation. For readability, the natural language distribution have been smoothed with a sliding average of 3 consecutive lengths.

(Standard Speaker + Impatient Listener) and LazImpa. We quantify the stability by introducing a coefficient δ_{stab} that measures the local variations of the accuracy curves. Formally, we compute the mean square error between the original accuracy curve and the smoothed curve obtained by averaging 10 consecutive score values:

$$\delta_{stab} = \frac{1}{n}\sum_{i=1}^{n}(f(i) - \tilde{f}(i))^2 \qquad (11)$$

where n is the total number of episodes, $f(.)$ the accuracy curve (as a function of the number of episode), $\tilde{f}(i)$ the curve obtained by averaging $f(.)$ over with 11 consecutive episodes centered in i. The lower δ_{stab} is, the smoother the system is .

Results are reported in Table 2. δ_{stab} for systems with Impatient Listener are smaller than the one with Standard Listener confirming the stability of the former. It is important noticing that, contrary to (Chaabouni et al., 2019)'s setting where they managed to have more efficient languages at the cost of stable convergence, our new communicative system, on top of leading to efficient languages, has positive impact on the convergence.

A.3.2 Complement on randomization test

To be comparable with Ferrer i Cancho et al. (2013), we perform the randomization test with 10^{-5} permutations. In the reference article, for a threshold α they introduce two types of p-values:

- Left p-value: if left p-value $< \alpha$, the code is characterized by L_{token} significantly smaller than the average weighted message length of any random permutation, corresponding to our notion of *ZLA code*.

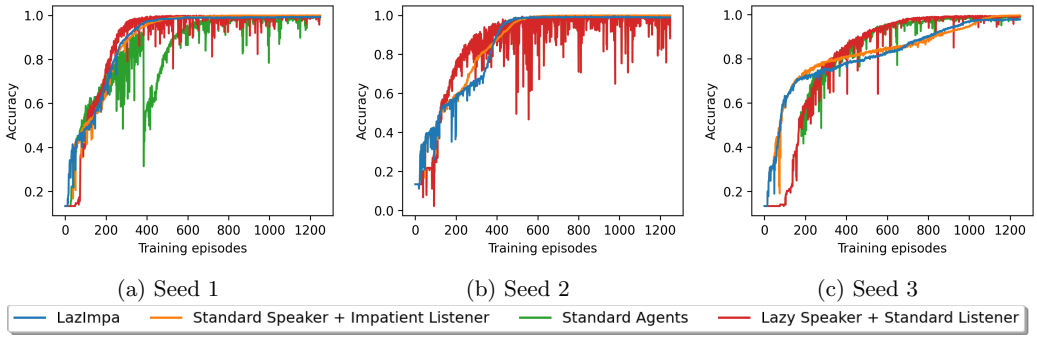


Figure 10: Evolution of the accuracy of the three systems for 3 randomly selected seeds.

	Standard Agents	Lazy Speaker + Standard Listener	Standard Speaker + Impatient Listener	LazImpa
δ_{stab}	$1.16 \pm 0.78 \times 10^{-3}$	$1.75 \pm 0.60 \times 10^{-3}$	$9.84 \pm 5.81 \times 10^{-5}$	$9.79 \pm 7.35 \times 10^{-5}$

Table 2: Average MSE between the original and smoothed accuracy curve

- Right p-value: if right p-value $< \alpha$, the code is characterized by L_{token} significantly higher than the average weighted message length of any random permutation, corresponding to our notion of *anti-ZLA code*.

In the main text, we only report the value of the ZLA significance score p_{ZLA} that is equivalent to Ferrer i Cancho et al. (2013)'s left p-value. However, when also considering right p-value (not shown here), we note for Standard Agents a value smaller than 10^{-5} asserting that the system shows a significantly anti-ZLA patterns.

A.4 Complements on LazImpa

A.4.1 minimal required length by Impatient Listener

Thanks to the incremental predictive mechanism of Impatient Listener, it is possible to analyze its intermediate guesses at each reading time. In particular, we are able to spot at which position Impatient Listener is first able to predict the correct output (we verify experimentally that, if Listener finds the correct output at position i, it always predicts the right output at position $j > i$). From these intermediate predictions, we define a distribution called 'minimal required length' of all the positions at which Impatient Listener is able to first predict the correct output (note that this distribution matches the distribution of the number of informative symbols by message).

We observe that Impatient Listener was often able to find the correct candidate before reading the `EOS` token. The resulting minimal length is presented in Figure 11 where we show the length distribution of the messages ranked by input frequency and the actual length required by the Impatient Listener to discriminate the messages. We see that the minimal required length by the Impatient Listener is slightly higher than the Optimal Coding. Interestingly, the difference can be partially explained by the use of a skewed distribution of the unigrams across the messages (the Optimal Coding relies on a uniform use of the symbols). Indeed, we compute an effective vocabulary size

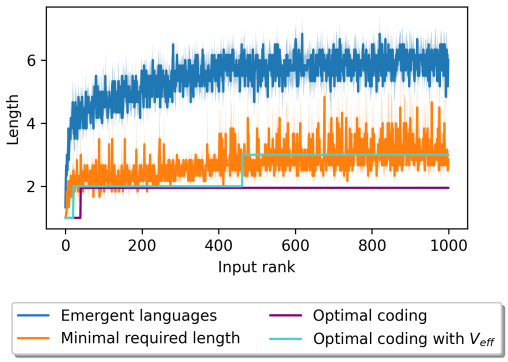


Figure 11: Comparison between the length distribution of the messages and the minimal required length for Impatient Listener to discriminate the messages. The blue curve shows average length distribution function of the inputs frequency ranks. The orange curve represents the average minimal required length by Impatient Listener to decode messages. The purple curve shows the Optimal Coding with the original vocabulary size. The red curve represents the Optimal Coding for the effective vocabulary size V_{eff}. Light intervals show 1 standard deviation.

V_{eff}, solution of Equation 12:

$$-\sum_{i=1}^{V_{eff}} \frac{1}{V_{eff}} \log\left(\frac{1}{V_{eff}}\right) = \mathcal{H}(\mathcal{U}), \qquad (12)$$

where V_{eff} is the effective vocabulary size, and $\mathcal{H}(\mathcal{U})$ the entropy of the unigram distribution $\mathcal{U}$ in the emergent communication.

In other words, we search for V_{eff} for which the entropy of a uniform unigram distribution (the left side of Equation 12) is equal to emergent languages average unigram distribution (the right side of Equation 12).

We plot in Figure 11 a new Optimal Coding with V_{eff} (Optimal Coding with V_{eff}). The distribution 'minimal required length' almost fits the Optimal Coding with this vocabulary size. As shown in Table 3, the average mean length L_{type} of minimal required length is almost equal to L_{type} of Optimal Coding with V_{eff}.

A.4.2 LazImpa robustness to parameters assumptions

In this section, we analyze LazImpa robustness to parameters changes. In the main paper, we made two main assumptions:

1. Samples are drawn according to a powerlaw;
2. `voc_size` = 40 and `max_len` = 30.

In the main paper, we demonstrated that LazImpa is able to reach efficient performances with this set of assumptions. We now want to test whether the system is robust to changes of these parameters, i.e. is LazImpa able to produce efficient and successful codes when inputs are drawn uniformly and/or for different values of `voc_size` ? We report the results of all our experiments in Table 4. Curves associated to experiments with variations of vocabulary size are shown in Figure 12. All these results have been obtained by averaging the results over 3 different seeds by each set of parameters.

Effect of `voc_size` :
As we can observe in Figure 12, emergent codes still respects ZLA for the various tested values of vocabulary size. This is confirmed by the ZLA significance score p_{ZLA} stored in Table 4a. Additionally, we can see a correlation between the size of the vocabulary and the efficiency of the emergent code: the emergent code is more efficient for large sizes of vocabulary. Indeed, we observe that L_{type}, L_{token} and L_{eff} are increasing functions of the vocabulary size. This is expected as the number of messages of a given length increases with the vocabulary size. Thus, the set of 'short' messages is higher for a large vocabulary size. Naturally, the same trend is observed with Optimal Coding.

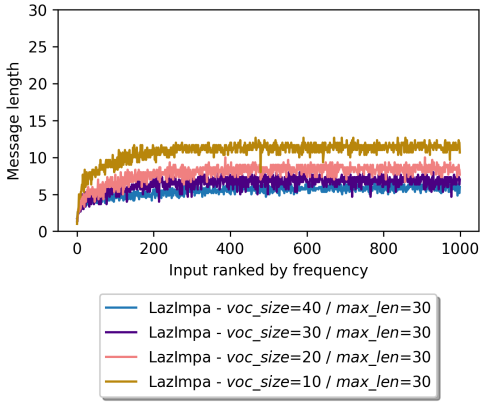


Figure 12: Comparison of LazImpa's average message length for different vocabulary sizes.

Moreover, we note a decrease of ρ_{inf} as a function of `voc_size` for the LazImpa system, suggesting that the smaller the vocabulary size is the more noninformative positions are used.

Effect of `max_len`: We can note in Table 4b that LazImpa is even closer to Optimal Coding when setting `max_len` = 20. L_{type}, L_{token} and L_{eff} are slightly smaller compared to experiments with `max_len` = 30. Thus, agents regularization seems to be easier when setting smaller values of `max_len`. Nevertheless, the results are very close. In particular, we can note that information density values ρ_{inf} are very similar suggesting that sub-optimality issues are independent of the parameter `max_len`. Note that we only explore two values of `max_len` in Table 4b because small and large values of `max_len` lead respectively to a small and large message space and thus optimization issues (H-parameters tuning is required to favor respectively exploration and exploitation).

Effect of input distribution: As we observe in Table 4c, LazImpa's performances are quite similar when dealing with inputs drawn from a uniform or a powerlaw distribution. In particular, with a uniform distribution, we observe a gain of efficiency for L_{type} and a loss of efficiency for L_{token} while L_{eff} is almost unchanged. All these results are expected. Equal L_{eff} means that Impatient Listener relies on the same number of symbols on average. In the main paper, we have shown that L_{eff} is mostly influenced by the entropy of the unigram distribution. Since, there is no change of `voc_size`, we do not expect major changes of entropy and thus no change for L_{eff}. Then, the difference of L_{token} and L_{type} is explained by the reduction step. For uniformly drawn inputs, the regularization is uniformly applied on the inputs ; for inputs drawn from a powerlaw, the regularization mostly focuses on the most frequent inputs

	Minimal required length	Opt. coding with V	Opt. coding with V_{eff}
L_{type}	2.74 ± 0.08	1.69	2.50

Table 3: Comparison of the average length L_{type} of different encoding. 'Opt. coding with V' to the Optimal Coding obtained with vocabulary V, 'Opt. coding with V_{eff}' to the Optimal Coding obtained with vocabulary V_{eff}. We also report standard deviation over all the experiments.

because they have larger weights in the loss. Consequently, we expect a lower L_{token} when experimenting with a powerlaw distribution, compared to the uniform setting, but a larger L_{type}. Eventually, we observe a significant gain of information density ρ_{inf} for LazImpa with a uniform distribution. This is mainly explained by ρ_{inf} computation that takes into account message lengths without involving their frequency.

As a remark, let's precise that we do not explore a larger set of non-uniform input distributions. In theory, the shape of the length distribution should not be impacted by the input distribution because the optimization problem is only dependent of the frequency ranks (mapping of the shortest messages to the most frequent inputs).

A.4.3 Statistical comparison between LazImpa and natural languages

Figure 13 shows the words length as a function of their frequency for both natural languages and the emergent language. This figure completes our comparison made in the main paper between LazImpa and natural languages where curves were smoothed. Here we show the raw natural languages distribution. The additional observation that we can make is that the variance of the words length is larger for the natural languages.

voc_size	System	L_{type}	L_{token}	p_{ZLA}	L_{eff}	ρ_{inf}
40	LazImpa	5.49 ± 0.67	3.78 ± 0.34	$< 10^{-5}$*	2.67 ± 0.07	0.60 ± 0.07
	Optimal Coding	2.96	2.29	$< 10^{-5}$*	1.96	1
30	LazImpa	6.49 ± 1.20	4.14 ± 0.43	$< 10^{-5}$*	2.71 ± 0.22	0.53 ± 0.07
	Optimal Coding	3.09	2.35	$< 10^{-5}$*	2.09	1.
20	LazImpa	7.91 ± 0.71	4.80 ± 0.30	$< 10^{-5}$*	2.98 ± 0.07	0.45 ± 0.04
	Optimal Coding	3.59	2.51	$< 10^{-5}$*	2.59	1.
10	LazImpa	10.82 ± 0.28	6.54 ± 0.06	$< 10^{-5}$*	3.87 ± 0.10	0.40 ± 0.005
	Optimal Coding	4.08	2.82	$< 10^{-5}$*	3.08	1.

(a) Variations of vocabulary size `voc_size`. By default, the input distribution is a powerlaw and `max_len` = 30.

max_len	System	L_{type}	L_{token}	p_{ZLA}	L_{eff}	ρ_{inf}
30	LazImpa	5.49 ± 0.67	3.78 ± 0.34	$< 10^{-5}$*	2.67 ± 0.07	0.60 ± 0.07
	Optimal Coding	2.96	2.29	$< 10^{-5}$*	1.96	1
20	LazImpa	4.36 ± 0.11	3.12 ± 0.06	$< 10^{-5}$*	2.40 ± 0.08	0.55 ± 0.01
	Optimal Coding	2.96	2.29	$< 10^{-5}$*	1.96	1

(b) Variations of maximum length `max_len`. By default, the input distribution is a powerlaw and `voc_size` = 40.

Distribution	System	L_{type}	L_{token}	p_{ZLA}	L_{eff}	ρ_{inf}
powerlaw	LazImpa	5.49 ± 0.67	3.78 ± 0.34	$< 10^{-5}$*	2.67 ± 0.07	0.60 ± 0.07
	Optimal Coding	2.96	2.29	$< 10^{-5}$*	1.96	1
uniform	LazImpa	4.27 ± 0.37	4.27 ± 0.37	/	2.53 ± 0.09	0.81 ± 0.08
	Optimal Coding	2.96	2.96	/	1.96	1

(c) Variations of input distribution. By default: `voc_size` = 40, `max_len` = 30.

Table 4: Efficiency analysis of LazImpa and Optimal Coding for different set of parameters. L_{type} is the mean message length, L_{token} is the mean weighted message length, p_{ZLA} the ZLA significance score, L_{eff} the effective length and ρ_{inf} the information density. '/' indicates that the metric is not relevant. For p_{ZLA}, '*' indicates that the p-value is significant (< 0.001).

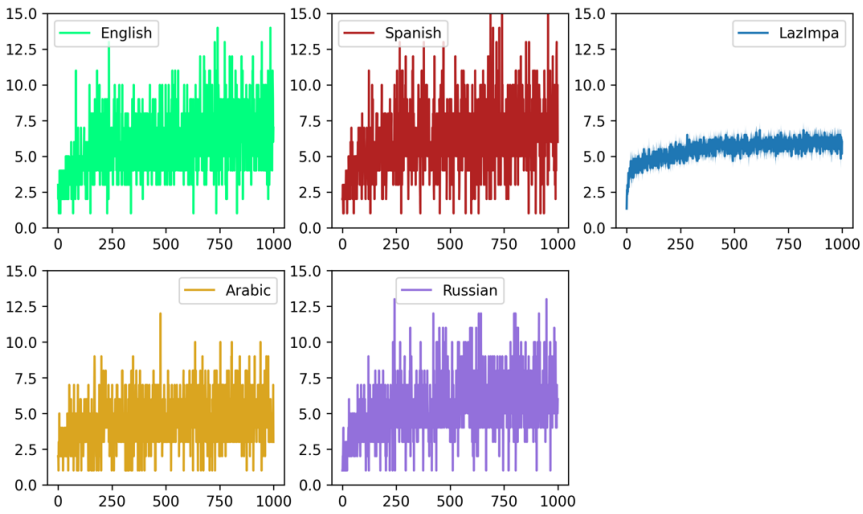


Figure 13: Comparison of the message length as a function of input frequency rank for LazImpa and natural languages.

Chapter 3

Word Order

The hypothesis that language is a consequence of an adaptive and evolutionary system dates back to Darwin and beyond [21]. According to this view, language is a complex "organism" that evolves to fit learners' general cognitive innate biases. For example, our limited memory can explain the regularity of non-frequent forms (e.g., laugh → laughed) and the irregularity of the frequent ones (e.g., go → went). High-frequency forms are easily learned, independently of whether they are regular or not, whereas low-frequency forms are harder to learn if they are not regular [76]. In this sense, language could be seen as a lens to investigate learners' innate cognitive biases.

In this chapter, we look at the relationship between NN learners' biases and their emergent languages. In particular, we study whether "natural" word-order regularities are found as "inductive" biases in recurrent NNs. We look at (1) temporal iconicity, (2) the trade-off between fixed-word order and case markers, and (3) the preference for local dependencies. To this end, we introduce several miniature languages that respect or violate these word-order regularities and study how they are acquired/transmitted by NN learners. We found that NN learners are biased toward short dependency-length, mirroring typological linguistic patterns. At the same time, they tend to prefer redundant encoding, in contrast to natural language. Importantly, this chapter introduces a framework that provides a simple paradigm to test NN inductive biases and thus contributes to more interpretable AI. Besides the interest for linguistics, this work might also be useful to develop better agents. For example, constraining NN

agents to use either case or fixed word order to mark grammatical functions would lead to a more efficient AI that avoids redundancy while being expressive. Also, as local dependencies and iconicity were shown to facilitate human sentence processing [38, 43, 53, 26], favoring these patterns in artificial models should be desirable for machine-human interaction.

Word-order biases in deep-agent emergent communication

Rahma Chaabouni[1,2], Eugene Kharitonov[1], Alessandro Lazaric[1],
Emmanuel Dupoux[1,2] and Marco Baroni[1,3]

[1]*Facebook A.I. Research*
[2]*Cognitive Machine Learning (ENS - EHESS - PSL Research University - CNRS - INRIA)*
[3]*ICREA*
{rchaabouni,kharitonov,lazaric,dpx,mbaroni}@fb.com

Abstract

Sequence-processing neural networks led to remarkable progress on many NLP tasks. As a consequence, there has been increasing interest in understanding to what extent they process language as humans do. We aim here to uncover which biases such models display with respect to "natural" word-order constraints. We train models to communicate about paths in a simple gridworld, using miniature languages that reflect or violate various natural language trends, such as the tendency to avoid redundancy or to minimize long-distance dependencies. We study how the controlled characteristics of our miniature languages affect individual learning and their stability across multiple network generations. The results draw a mixed picture. On the one hand, neural networks show a strong tendency to avoid long-distance dependencies. On the other hand, there is no clear preference for the efficient, non-redundant encoding of information that is widely attested in natural language. We thus suggest inoculating a notion of "effort" into neural networks, as a possible way to make their linguistic behavior more human-like.

1 Introduction

Deep neural networks, and in particular "sequence-to-sequence" (Seq2Seq, Sutskever et al., 2014) LSTM recurrent networks, attained astounding successes in many linguistic domains (Goldberg, 2017), but we still have a poor understanding of their language processing mechanisms (Lake and Baroni, 2018). We study here whether word-order constraints commonly observed in natural language are also found as "inductive" biases in recurrent networks. We consider three such constraints. The first is temporal *iconicity*, defined as the tendency of clauses denoting events to reflect the chronological order of the denoted events (as in Caesar's *veni, vidi, vici*; Greenberg, 1963; Haiman, 1980; Newmeyer, 1992; Radden and Dirven, 2007; Diessel, 2008; Marcus and Calude, 2010; de Ruiter et al., 2018). The second is the need to disambiguate the role of sentence constituents, that can be achieved either by means of fixed-word order (e.g., in an SVO language the first noun phrase denotes the subject), or by overting morphological markers (e.g., the subject is marked with nominative case). As the two mechanisms are redundant, a trade-off is generally observed, where languages preferentially adopt one or the other (Comrie, 1981; Blake, 2001). Finally, we consider the general tendency of languages to avoid or minimize long-distance dependencies (Hawkins, 1994; Gibson, 1998; Futrell et al., 2015). As Futrell et al. (2015) observe, "I checked [it] out", with one word intervening between the verb and the particle it composes with, *'is easier or more efficient to produce and comprehend'* than "I checked [the place you recommended] out", with four intervening words.

We test whether such constraints affect LSTM-based Seq2Seq models. To this end, we train them as agents in a simple $2D$ gridworld environment, in which they give and receive navigation instructions in hand-designed artificial languages satisfying or violating the constraints. We first study which languages are harder to learn for individual agents. Then, we look at the cultural transmission of language characteristics through multiple agent generations by means of the iterated learning paradigm (Kirby et al., 2014).[1]

Our results suggest a mixed picture. LSTM agents are partially affected by natural constraints, both in terms of learning difficulty and stability of patterns through evolution. For example, they

[1]Code link: https://github.com/facebookresearch/brica.

show a strong tendency to avoid long-distance dependencies. Still, some patterns are considerably different from those encountered in human language. In particular, LSTMs generally have a preference for the reverse version of an iconic language, and only show a weak tendency towards avoidance of redundant coding.

2 Related work

There is increasing interest in applying methods from linguistics and psychology to gain insights on the functioning of language processing networks, as witnessed by the recent BlackBoxNLP workshop at EMNLP 2018 (Linzen et al., 2018). In this context, researchers have looked at how *trained* models solve different NLP tasks characterizing their outputs and internal representation. We instead focus directly on uncovering their "innate" biases *while learning a task*.

We study whether LSTM-based Seq2Seq models deployed as communicating agents are subject to some of the natural pressures that characterize the typology and evolution of human languages. In this respect, we connect to the recent research line on language emergence in deep network agents that communicate to accomplish a task (e.g., Jorge et al., 2016; Havrylov and Titov, 2017; Kottur et al., 2017; Lazaridou et al., 2017; Choi et al., 2018; Evtimova et al., 2018; Lazaridou et al., 2018; Mordatch and Abbeel, 2018). Most of this work provides the agents with a basic communication channel, and evaluates task success and the emerging communication protocol in an entirely bottom-up fashion. We train instead our agents to communicate with simple languages possessing the properties we want to study, and look at whether such properties make the languages easier or harder to learn. Other studies (Lee et al., 2017b,a) had also seeded their agents with (real) languages, but for different purposes (letting them develop translation skills).

We introduce miniature artificial languages that respect or violate specific constraints. Other studies have used such languages with human subjects to test hypotheses about the origin of cross-linguistically frequent patterns (see Fedzechkina et al., 2016b, for a survey). We follow this approach to detect biases in *Seq2Seq models*. We specifically rely on two different measures. First, we evaluate the speed of learning a particular language, assuming that the faster it is, the easier its properties are for the agent (e.g., Tily et al., 2011; Hupp et al., 2009). Second, we look at the cultural evolution of a language by means of the *iterated language learning* paradigm (see Kirby et al., 2014, for a survey). That is, we investigate the changes that modern Seq2Seq networks exposed to a language through multiple generations introduce, checking which biases they expose.

3 Experimental setup

3.1 Languages

Our environment is characterized by trajectories of 4 oriented actions (LEFT, RIGHT, UP, DOWN). A trajectory contains from 1 to 5 segments, each composed of maximally 3 steps in the same direction. A possible 3-segment trajectory is: LEFT LEFT RIGHT UP UP UP, with (LEFT LEFT), (RIGHT), and (UP UP UP) being its segments.

Fixed- and free-order languages In a *fixed-order* language, a segment is denoted by a phrase made of a command (C) and a quantifier (Q). An utterance specifies an order for the phrases. For example, in the *forward-iconic* language, 3-phrase utterances are generated by the following rules:

(1)
$$\begin{aligned} &U \rightarrow P1\ P2\ P3 \\ &P(1|2|3) \rightarrow C\ Q \\ &C \rightarrow (\text{left}|\text{right}|\text{up}|\text{down}) \\ &Q \rightarrow (1|2|3) \end{aligned}$$

Shorter and longer utterances are generated analogously (a N-phrase utterance always has form P1 P2 ... PN). Importantly, the interpretation function associates PN to the N-th segment in a trajectory, hence the temporal iconicity of the grammar. For example, the utterance "left 2 right 1 up 3" denotes the 3-segment trajectory: LEFT LEFT RIGHT UP UP UP.

The *backward-iconic* language is analogous, but phrases are interpreted right-to-left. *Non-iconic* languages use the same interpretation function associating PN to the N-th segment, but now the grammar licenses phrases in a fixed order different from that of the trajectory. For example, 3-phrase utterances might be generated by U $\rightarrow$ P2 P3 P1 (the trajectory above would be expressed by: "right 1 up 3 left 2"). Relative phrase ordering is fixed across utterances irrespective of length. For example, 2-phrase utterances in the language we just illustrated must be generated by U$\rightarrow$P2 P1, to respect the fixed-relative-ordering constraint for

P2 and P1 with respect to the 3-phrase rule.

Fixed-order languages *with (temporal ordering) markers* use the same utterance rules, but now each phrase PN is also associated with an unambiguous marker. For example, the *iconic+markers* language obeys the first rule in (1), but the phrases are expanded by:

(2) P1 → first C Q
P2 → second C Q
P3 → third C Q

In the iconic+markers language, the trajectory above is expressed by "first left 2 second right 1 third up 3".

A *free-order* language licenses the same phrase structures as a fixed-order language and it uses the same interpretation function, but now there are rules expanding utterances with all possible phrase permutations (e.g., 3-phrase utterances are licensed by 6 rules: U → P1 P2 P3, U → P1 P3 P2, ...).[2] Both "second right 1 third up 3 first left 2" and "third up 3 second right 1 first left 2" are acceptable utterances in the free-order language with markers. Examples of trajectory-to-utterance mappings of these artificial languages are provided in Supplementary

Long-distance language We consider a long-distance language where any phrase can be split and wrapped around a single other phrase so that a long-distance dependency is created between the components of the outermost phrase.[3] We treat long-distance dependencies as optional, as in languages in which they are optionally triggered, e.g., by information structure factors. We compare the *long-distance* language to a *local* free-order language lacking the long-distance split construction. Since the long-distance option causes a combinatorial explosion of possible orders, we limit trajectories to 3 segments. At the same time, to have two languages partially comparable in terms of variety of allowed constructions, we extend the grammars of both to license free order within a phrase. Finally, markers are prefixed to both the command and the quantifier, to avoid ambiguities in the long-distance case. Summarizing, the local language is similar to the free-order+markers one above, but markers are repeated before each phrase element, and extra rules allow the quantifier to precede or go after the command, e.g., both of the following structures are permitted: P1 → first Q first C; P1 → first C first Q ("first left first 2"; "first 2 first left"). The long-distance grammar further includes rules where P1 has been split in two parts, such as:

(3) U → first C1 P2 first Q1 P3
U → first Q1 P2 first C1 P3

with C1 and Q1 expandable into the usual terminals (LEFT, RIGHT... and 1, 2, 3, respectively).[4] The interpretation function associates a discontinuous {CN, QN} phrase with the N-th segment in the trajectory. The first rule in (3) licenses the utterance "first left second right second 1 first 2 third up third 3", denoting the example trajectory at the beginning of this section. Similar rules are introduced for all possible splits of a phrase around another phrase (e.g., the elements of P2 around P1, those of P1 around P3, etc.). Only one split is allowed per-utterance. Examples of trajectory-to-utterance mappings in the long and local-distance languages are provided in Supplementary.

Datasets We generate sentences associated to all possible trajectories in the environment (88572 in the fixed- and free-order language environment, 972 in the local- and long-distance environment experiments). We randomly split all possible distinct trajectory-utterance pairs into training (80%) and test/validation sections (10% each).

3.2 Models

Architecture The agents are Encoder-Decoder Seq2Seq architectures (Cho et al., 2014; Sutskever et al., 2014) with single-layer LSTM recurrent units (Hochreiter and Schmidhuber, 1997). In light of the interactive nature of language, an agent is always trained to be both a *Speaker*, taking a trajectory as input and producing an utterance describing it, and as a *Listener*, executing the trajectory corresponding to an input utterance. Input and output vocabularies are identical, and contain all possible actions and words.[5] When an agent plays the Speaker role, it uses input action representations and output word representations, and conversely in the Listener role. We tie the embed-

[2] Equivalently, a free-order language is generated in two stages from a fixed-order one through a scrambling process.

[3] Note also that this language is projective, excluding cross-dependencies.

[4] Equivalently, long-distance constructions are derived by movement rules from canonical underlying structures.

[5] Word and action symbols are disjoint, e.g., the action symbol 'LEFT' is different from the word symbol 'left'.

dings of the encoder input and of the decoder output (Press and Wolf, 2016) making input and output representations of words and actions coincide. As a result, Speaker training affects the representations used in Listener mode and *vice versa*. Experiments without tying (not reported) show similar results with slower convergence. We additionally explore a standard attention mechanism (Bahdanau et al., 2014).

Training We consider two scenarios. In *individual learning*, an agent is taught a language by interacting with a hard-coded ground-truth "teacher", represented by the training corpus. In the *iterated learning* setup, a lineage of agents is trained to speak and listen by interacting with a "parent" agent. After convergence, an agent is fixed and used as a parent to train the next child.

Individual learning We synchronously train the agent to speak (from trajectory $\boldsymbol{t}$ to utterance $\boldsymbol{u}$) and listen (from utterance $\boldsymbol{u}$ to trajectory $\boldsymbol{t}$). Training the Listener is similar to standard Seq2Seq training with teacher forcing (Goodfellow et al., 2016, p. 376). We change the training procedure for the Speaker direction, as we must handle one-to-many trajectory-to-utterance mappings in free-order languages. We describe it below.

For each trajectory, we consider all corresponding utterances equally probable. Given a trajectory input, an agent must be able to produce, with equal probability, all utterances that correspond to the input. To achieve this, taking inspiration from the multi-label learning literature, we fit the agent's output distribution to minimize KL-divergence from the uniform over target utterances. We adopt the "Naïve" method proposed by Jin and Ghahramani (2003) (see Supplementary for how we derive the loss function in Eq. (4)).

Formally, our languages map trajectories $\boldsymbol{t}_j$ to one (fixed-order) or multiple (free-order) utterances $\{\boldsymbol{u}\}_j = \{\boldsymbol{u}_j^1, \boldsymbol{u}_j^2, \ldots\}$. The trajectory $\boldsymbol{t}$ is fed into the encoder, which produces a representation of the action sequence. Next, the latter is fed into the decoder along with the start-of-the-sequence element $u_0 = sos$. At each step, the decoder's output layer defines a categorical distribution $p_{\boldsymbol{\theta}}(u_k|u_{k-1}, \boldsymbol{h}_k)$ over the next output word u_k. This distribution is conditioned by the previous word u_{k-1} and the hidden state $\boldsymbol{h}_k$. As with the Listener, we use teacher forcing, so that the distribution of each word is conditioned by the ground-truth terms coming before it.

Overall, the model parameters $\boldsymbol{\theta}$ are optimized to minimize the loss $\mathcal{L}$ over $(\boldsymbol{t}_j, \{\boldsymbol{u}\}_j)$:

$$\mathcal{L} = -\sum_j \frac{1}{n_j} \sum_{\boldsymbol{u} \in \{\boldsymbol{u}\}_j} \sum_{k=1}^{|\boldsymbol{u}|} \log p_{\boldsymbol{\theta}}(u_k|u_{k-1}, \boldsymbol{h}_{j,k}) \tag{4}$$

In Eq. (4), n_j denotes the number of target utterances for the jth example, $n_j = |\{\boldsymbol{u}\}_j|$; $\boldsymbol{u}$ iterates over the utterances $\{\boldsymbol{u}\}_j$; and u_k enumerates words in the utterance $\boldsymbol{u}$ as k varies. As the number of ground-truth utterances $\{\boldsymbol{u}\}_j$ can be high, we sub-sample $n = 6$ when training free- and fixed-order languages.[6] This considerably speeds up training without significantly harming performance. We use all the possible utterances when training on long-distance languages (n equals the the number of all possible utterances).

For all studied languages, we perform a grid search over hidden layer [16,20] and batch sizes [16,32], and report test set results of the best validation configuration for each language re-initialized with 5 different seeds. We stop training if development set accuracy does not increase for 5 epochs or when 500 epochs are reached. In all scenarios, the optimization is performed with the *Amsgrad* (Reddi et al., 2018) which is an improved version of the standard Adam (Kingma and Ba, 2014); we did not experiment with other optimizers. We use the algorithm with its default parameters, as implemented in Pytorch (Paszke et al., 2017).

Iterated learning At "generation 0" agent $A_{\boldsymbol{\theta}_0}$ is trained individually as described above. Once $A_{\boldsymbol{\theta}_0}$ is trained, we fix its parameters and use it to train the next-generation agent, $A_{\boldsymbol{\theta}_1}$. $A_{\boldsymbol{\theta}_1}$, after training, is in its turn fixed and used to train the next agent $A_{\boldsymbol{\theta}_2}$, etc. At each iteration, the child agent $A_{\boldsymbol{\theta}_{i+1}}$ is trained to *imitate* its parent $A_{\boldsymbol{\theta}_i}$ as follows. Suppose that, given $\boldsymbol{t}$, the parent agent produces n[7] utterances $\{\hat{\boldsymbol{u}}\} = \{\hat{\boldsymbol{u}}^1, \hat{\boldsymbol{u}}^2, \ldots \hat{\boldsymbol{u}}^n\}$ (these utterances are obtained by sampling from the parent's decoder and can be identical). Then, we train the child agent to: (a) *listen*: map each utterance $\hat{\boldsymbol{u}}^j$ to the trajectory $\boldsymbol{t}$, and (b) *speak*: given

[6] Sampling is trivial in the latter case, since $\{\boldsymbol{u}\}_j$ contains a single utterance. Note that in this case the loss $\mathcal{L}$ reduces to the negative log-likelihood. This allows us to use the same loss function for free- and fixed-order languages.

[7] We use the same number n defined in individual learning section.

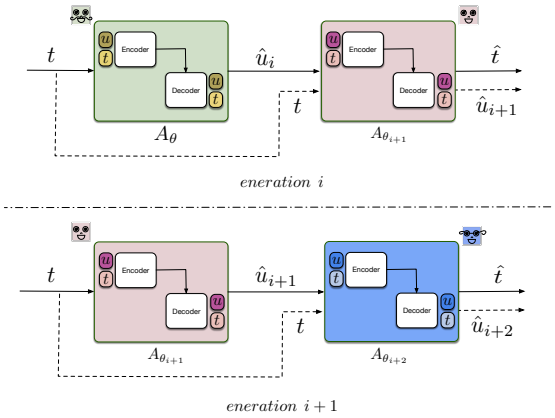


Figure 1: **Iterated learning**. Language is transmitted to a child agent $A_{\boldsymbol{\theta}_{i+1}}$ by teaching it to *speak* imitating the utterances of parent $A_{\boldsymbol{\theta}_i}$ given the same input trajectories (dashed lines) and to *listen* to the parent utterances, converting them to trajectories (continuous lines). After training, former child $A_{\boldsymbol{\theta}_{i+1}}$ becomes the parent of a new agent $A_{\boldsymbol{\theta}_{i+2}}$.

the trajectory $\boldsymbol{t}$, produce the utterance $\hat{\boldsymbol{u}}$ that is within $\{\hat{\boldsymbol{u}}\}$ (Fig. 1). Importantly, even if the parent's parameters are fixed at each generation, the child agent is allowed, while achieving perfect accuracy, to introduce changes into its' parent language, making the latter more closely aligned with its "innate" biases. [8]

Importantly, the language is not forced to remain stationary across generations.

Evaluation We evaluate agents both as Listeners and as Speakers. The former is standard, as each input $\boldsymbol{u}$ maps to a single output $\boldsymbol{t}$. Since the Speaker can be one-to-many, in order to obtain a single prediction $\boldsymbol{u}$ given trajectory $\boldsymbol{t}$, we predict at each time step k a word $u_k^* = \arg\max_{u_k}(p_{\boldsymbol{\theta}}(u_k|u_{k-1}^*, \boldsymbol{h}_k))$. This word is fed to the next unit of the decoder, and so on until $u_K^* = eos$. The final prediction $\hat{\boldsymbol{u}}^*$ is then defined as the sequence $[u_1^*, u_2^* ... u_K^*]$, and compared to M *samples* from the true distribution $P(\boldsymbol{u}|\boldsymbol{t})$. If $\hat{\boldsymbol{u}}^*$ matches *one* of the true samples, the agent succeeds, otherwise it fails (in iterated learning, $P(\boldsymbol{u}|\boldsymbol{t})$ corresponds to the parent's distribution). In other words, we are not evaluating the model on a perfect fit of the ground-truth (parent's, in case of iterated learning) distribution, but we score a hit for it as long as it outputs a combination in $P(\boldsymbol{u}|\boldsymbol{t})$. This mismatch between the training and evaluation criteria allows the emergence of interesting patterns (as we allow the agent to drift from the ground-truth distribution) while constituting a reasonable measure of actual communication success (as the agent produces an utterance that is associated to the input trajectory in the ground-truth).

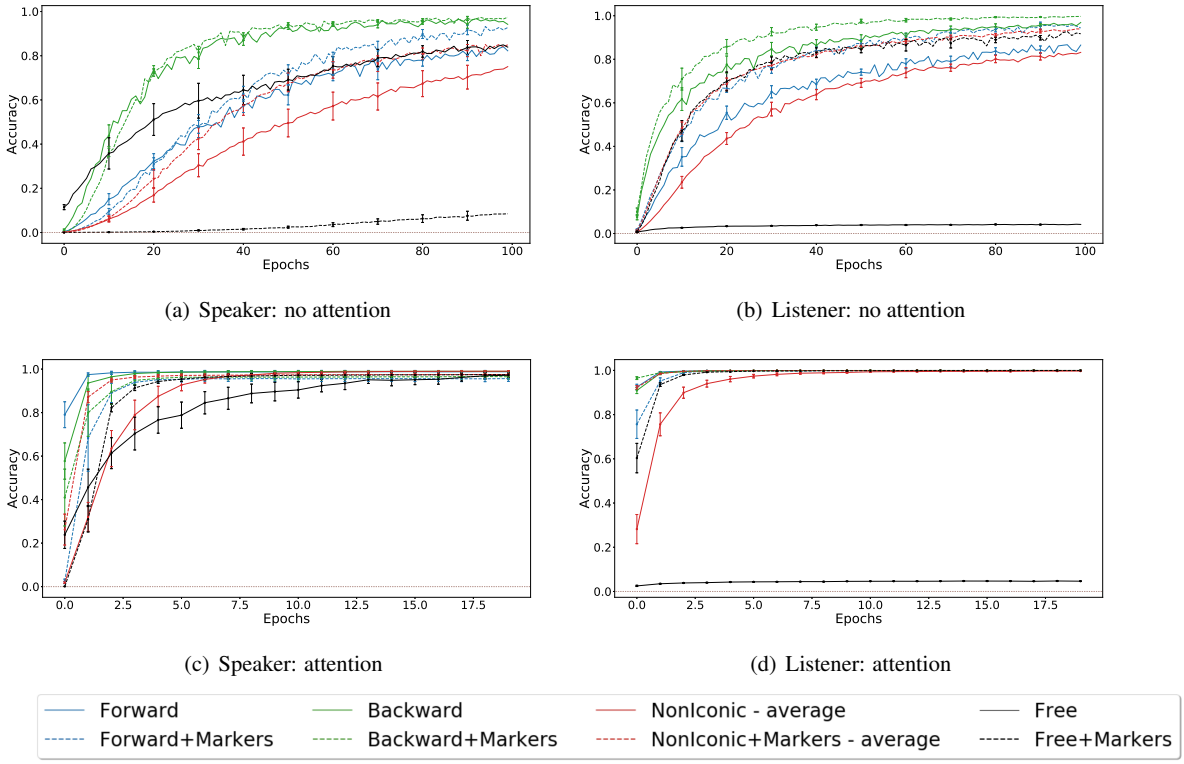
[8] as exemplified in the experiments below, the child can reach perfect accuracy while having a different distribution over the utterances than its parent.

4 Experiments

4.1 Iconicity, word order, and markers

We compare languages with fixed and free order, with and without markers. Experiments with humans have shown that, as listeners, children perform better with iconic sentences than non-iconic ones (de Ruiter et al., 2018). We check whether Seq2Seq networks show similar preferences in terms of learning speed and diachronic persistence. We compare in particular the forward-iconic order with the backward-iconic language, and three randomly selected non-iconic languages where the relation between segment and phrase order is fixed but arbitrary. Concerning the relation between fixed order and markers, typological studies show a trade-off between these cues. For example, languages with flexible word order (e.g., Japanese, and Russian) often use case to mark grammatical function, whereas languages with fixed word order (such as English and Mandarin) often lack case marking (Blake, 2001; Comrie, 1981). This might be explained by a universal preference for efficient and non-redundant grammatical coding (Fedzechkina et al., 2016a; Qian and Jaeger, 2012; Zipf, 1949). Seq2Seq agents might show similar preferences when tested as Speakers. That is, they might show a learning and preservation preference for either fixed no-marking languages or free marking languages.

Individual learning. Fig. 2 shows test accuracy during learning for each language type. The no-attention agent has a preference for backward-iconic both in speaking and listening. This is in line with the observation that Seq2Seq machine translation models work better when the source is presented in reverse order as it makes the optimization problem easier by introducing shorter-term dependencies (Sutskever et al., 2014). The (forward) iconic order is better than the non-iconic ones in the speaking direction only. The attention-enhanced model shows much faster convergence to near-perfect communication, with less room for clear biases to emerge. Still, we observe some interesting initial preferences. In speaking mode, the



(a) Speaker: no attention

(b) Listener: no attention

(c) Speaker: attention

(d) Listener: attention

Figure 2: **Iconicity / Fixed vs. free order**: Mean test set accuracy in function of training epoch. Error bars represent standard deviation over five random seeds. The NonIconic-average curve pools measurements for 3 non-iconic languages, each with five runs. Chance accuracy is represented by the horizontal dotted line. The continuous lines represent languages without markers, while the dashed lines represent languages with markers.

agent learns fastest with the forward iconic language, followed by the backward one. The non-iconic language without markers is the most difficult to learn, as expected. On the other hand, in listening mode we encounter again a preference for backward iconicity.

Only the attention agent in speaking mode shows a trade-off between order and markers coding, with a preference for markers-free fixed-order iconic languages over their counterparts with markers, and for the free-order language with markers over the marker-less one. Only the non-iconic languages violate the trend: arguably, though, non-iconic order coding is so sub-optimal that redundant markers are justified in this case. In listening mode, this agent shows the expected preference for markers in the free-order case (as the free-order language without markers is massively ambiguous, with most utterances mapping to multiple trajectories). However, among the fixed-order languages, both backward and non-iconic prefer redundant coding. The agent without attention also displays a preference for free-order+markers in listening mode (while it has serious difficulties to learn to speak this language), but no clear avoidance for redundant coding in either modes. In sum, we confirm a preference for iconic orders. Only the attention-enhanced agent in speaking mode displays avoidance of redundant coding.

Iterated learning. In iterated learning, we might expect the lineage of agents that starts with less natural non-iconic languages to either converge to speak more iconic ones, or possibly to drift into low communication accuracy. We moreover expect redundant coding to fade, with fixed-order+markers languages to either evolve free order or lose markers. Regarding the free-word order marked language, we expect it to either converge to a fixed order (possibly iconic) while losing its markers, as in the historical development from Old English (a language with flexible constituent order and rich case marking) to Modern English (a language with fixed constituent order and a rudimentary case system) (Traugott, 1972), or to remain stable maintaining good communica-

tion accuracy. We focus on the attention agent, as the no-attention one converges too slowly for multiple-generation experiments. We simulate 10 generations, repeating each experiment with 5 different initialization seeds. For non-iconic orders, we sample the same 3 languages sampled for individual learning.

For fixed-order languages, we do not observe any change in accuracy or behavior in the listener direction (the last-generation child is perfectly parsing the initial language). However, we observe in speaker mode a (relatively small) decrease in accuracy across generations, which, importantly, affects the most natural language (forward iconic without markers) the least, and the most difficult language (non-iconic without markers) the most (results are in Supplementary). Again, we observe a (weak) tendency for the attention agent to yield to the expected natural pressures.

We counted the overall number of markers produced by children in speaker mode after convergence, for all test trajectories in all languages with redundant coding. It was always constant, showing no trend towards losing markers to avoid redundant coding. Similarly, there was no tendency, across generations, to start producing multiple utterances in response to the same test trajectory.

In the evolution of the free-order language with markers, accuracy was relatively stable in both speaking and listening (99.82% and 100%, respectively, for the last-generation agent, averaging across 25 runs).[9] However, we noticed that across generations, the language becomes more fixed with some preferred orders emerging. Fig. 3 quantifies this in terms of the entropy of the observed phrase order probabilities across all test set trajectories (the lower the entropy, the more skewed the distribution). There is already a clear decrease for the first agent with respect to the ground-truth distribution, and the trend continues across generations. We analyzed the distribution of Speaker utterances for the longest (5-segment) test trajectories in the last generation. We found that, out of 120 possible phrase orders, no last-generation agent used more than 10. This is in line with the typological observation that even non-configurational languages favor (at least statistically) certain orders (Hale, 1992; Mithun, 1992) and thus an equiprobable distribution of orders, as it is the case in our free word-order+markers language, is unlikely. The "survivor" orders of the last generation were not necessarily iconic but depended notably on the seed. The absence of clear preference for a specific order could be explained by the fact that attention-enhanced agents, as we saw, can learn any fixed-order language very fast. In this case, the seed of one generation, by randomly skewing the statistics in favor of one order or the other, can significantly impact the preference toward the favored order, that will then spread diachronically throughout the whole iteration.

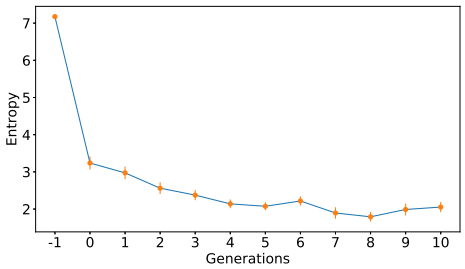


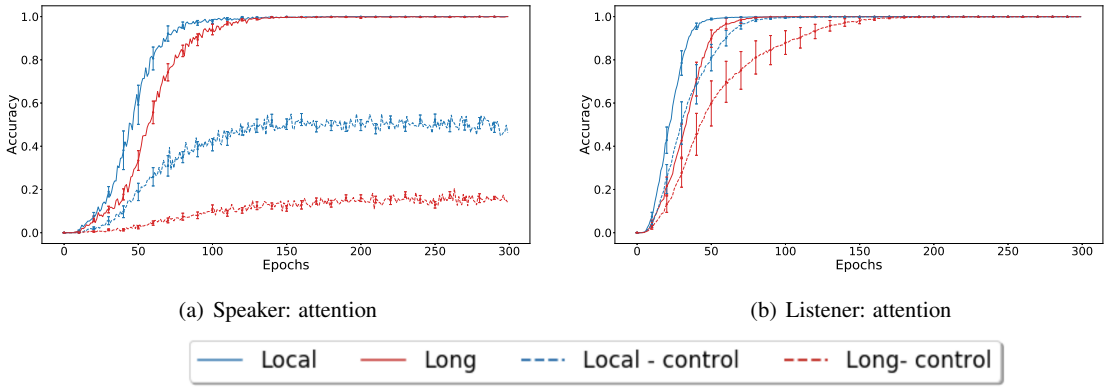
Figure 3: **Phrase-order entropy** in attention Speaker utterances given test set trajectories, in function of training generation (-1 represents the initial ground-truth distribution). Curve represents mean across 25 runs, with error bars for standard deviations.

4.2 Local vs. long-distance

We finally contrast the long-distance and local languages described in Section 3.1. In accordance with the linguistic literature (see Introduction), we predict that the long-distance language will be harder to learn, and it will tend to reduce long-distance constructions in diachrony. Although evidence for distance minimization is typically from production experiments (e.g., Futrell et al., 2015), we expect long-distance constructions to also be harder in perception, as they cannot be fully incrementally processed and require keeping material in memory for longer spans.

Individual learning. As the long-distance language includes all utterances from the local language, it might be trivially harder to learn. To account for this, we construct a set of *control* languages by randomly sampling, for each trajectory, the same number of possible utterances for the local and long-distance controls. We report averaged

[9] We run more simulations in this case as we noticed that the final language depends on the initial seed, and hence there is high variance with only 5 runs. Specifically, we start with 5 different parents and simulate 10 generations, repeating each experiment with 5 different seeds



(a) Speaker: attention

(b) Listener: attention

Figure 4: **Long vs. local distance**: Mean test set accuracy as a function of training epoch. The error bars correspond to the standard deviation, calculated over five random seeds.

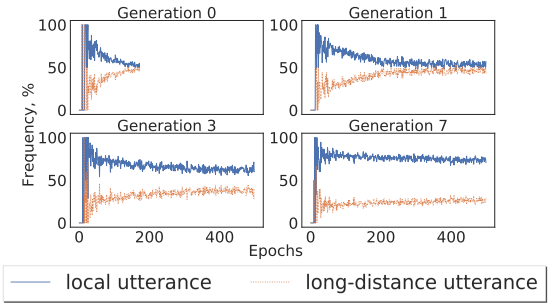


Figure 5: Frequency of the local and long-distance utterances produced by the attention Speaker in function of training epoch. The input trajectories are taken from the test set. Test set accuracies for the four generations shown: 99.99%, 87.62%, 84.54%, 79.38%. At Generation 0, less epochs were run due to early stopping.

results for 3 such languages of both kinds. Details on their construction are in Supplementary.

Fig. 4 shows test set accuracy across 300 training epochs for the attention model. The results, for speaking and listening, confirm the preference for the local language. The control languages are harder to learn, as they impose an arbitrary constraint on free word order, but they display the preference for the local language even more clearly. Overall, we see a tendency for listening to be easier than speaking, but this cuts across the local/long-distance division, and it seems to be a more general consequence of free-order languages with markers being easier in parsing than production (cf. the no-attention agent results in Fig. 2). Results without attention (not shown) are comparable in general, although the listener/speaker asymmetry is sharper, with no difference in difficulty among the 4 languages when listening.

Iterated learning. We study multiple-generation transmission of the long-distance language with the attention agent. To deal with the problem of skewed relative frequency of long-distance and entirely local utterances, the Speaker direction is trained by ensuring that the output utterance set $\{\boldsymbol{u}\}$ for each input trajectory $\boldsymbol{t}$ contains the same number of long-distance and local constructions. This is achieved by sub-sampling $n = 48$ long-distance utterances to match the number of possible local constructions. Fig. 5 shows the relative frequency across generations of local and long-distance utterances produced by the agent as a Speaker in function of training (one representative seed of 5). As predicted, a clear preference for local constructions emerges, confirming the presence of a distance minimization bias in Seq2Seq models.

5 Discussion

We studied whether word-order constraints widely attested in natural languages affect learning and diachronic transmission in Seq2Seq agents. We found that some trends follow natural patterns, such as the tendency to limit word order to few configurations, and long-distance dependency minimization. In other ways, our agents depart from typical human language patterns. For example, they exhibit a preference for a backward order, and there are only weak signs of a trade-off between different ways to encode constituent roles, with redundant solutions often being preferred.

The research direction we introduced might lead to a better understanding of the biases that affect the linguistic behaviour of LSTMs and simi-

lar models. This could help current efforts towards the development of artificial agents that communicate to solve a task, with the ultimate goal of developing AIs that can talk with humans. It has been observed that the communication protocol emerging in such simulations is very different from human language (e.g., Kottur et al., 2017; Lewis et al., 2017; Bouchacourt and Baroni, 2018). A better understanding of what are the "innate" biases of standard models in highly controlled setups, such as the one studied here, should complement large-scale simulations, as part of the effort to develop new methods to encourage the emergence of more human-like language. For example, our results suggest that current neural networks, as they are not subject to human-like least-effort constraints, might not display the same trend towards efficient communication that we encounter in natural languages. How to incorporate "effort"-based pressures in neural networks is an exciting direction for future work.

6 Acknowledgments

We would like to thank Roger Levy, Diane Bouchacourt, Alex Cristea, Kristina Gulordava and Armand Joulin for their very helpful feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barry Blake. 2001. *Case*. MIT Press, Cambridge, MA.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of EMNLP*, pages 981–985, Brussels, Belgium.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. 2018. Compositional obverter communication learning from raw visual input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada.

Benrard Comrie. 1981. *Language Universals and Linguistic Typology*. Blackwell, Malden, MA.

Laura de Ruiter, Anna Theakston, Silke Brandt, and Elena Lieven. 2018. Iconicity affects children's comprehension of complex sentences: The role of semantics, clause order, input and individual differences. *Cognition*, 171:202–224.

Holger Diessel. 2008. Iconicity of sequence: A corpus-based analysis of the positioning of temporal adverbial clauses in English. *Cognitive Linguistics*, 19(3):465–490.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. 2018. Emergent communication in a multi-modal, multi-step referential game. In *Proceedings of ICLR Conference Track*, Vancouver, Canada.

Maryia Fedzechkina, Elissa Newport, and T. Florian Jaeger. 2016a. Balancing effort and information transmission during language acquisition: Evidence from word order and case marking. *Cognitive Science*, 41:n/a–n/a.

Maryia Fedzechkina, Elissa Newport, and T. Florian Jaeger. 2016b. *Miniature artificial language learning as a complement to typological data*, pages 211–232.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 language. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Yoav Goldberg. 2017. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, San Francisco, CA.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Joseph Greenberg. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph Greenberg, editor, *Universals of Human Language*, pages 73–113. MIT Press, Cambridge, MA.

John Haiman. 1980. The iconicity of grammar: Isomorphism and motivation. *Language*, 56(3):515–540.

Kenneth Hale. 1992. Basic word order in two 'free word order' languages. In Doris Payne, editor, *Pragmatics of word order flexibility*, pages 63–82. John Benjamins, Amsterdam, the Netherlands.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Proceedings of NIPS*, pages 2149–2159, Long Beach, CA, USA.

John Hawkins. 1994. *A Performance Theory of Order and Constituency*. Cambridge University Press, Cambridge, UK.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Julie M. Hupp, Vladimir M. Sloutsky, and Peter W. Culicover. 2009. Evidence for a domain-general mechanism underlying the suffixation preference in language. *Language and Cognitive Processes*, 24(6):876–909.

Rong Jin and Zoubin Ghahramani. 2003. Learning with multiple labels. In *Advances in neural information processing systems*, pages 921–928.

Emilio Jorge, Mikael Kågebäck, and Emil Gustavsson. 2016. Learning to play Guess Who? and inventing a grounded language as a consequence. In *Proceedings of the NIPS Deep Reinforcement Learning Workshop*, Barcelona, Spain.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Simon Kirby, Tom Griffiths, and Kenny Smith. 2014. Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28:108–114.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of EMNLP*, pages 2962–2967, Copenhagen, Denmark.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of ICML*, pages 2879–2888, Stockholm, Sweden.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *Proceedings of ICLR Conference Track*, Toulon, France.

Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. 2017a. Emergent translation in multi-agent communication. *arXiv preprint arXiv:1710.06922*.

Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. 2017b. Answerer in questioner's mind for goal-oriented visual dialogue.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? End-to-end learning of negotiation dialogues. In *Proceedings of EMNLP*, pages 2443–2453, Copenhagen, Denmark.

Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors. 2018. *Proceedings of the EMNLP BlackboxNLP Workshop*. ACL, Brussels, Belgium.

Solomon Marcus and Andreea Calude. 2010. Syntactic iconicity, within and beyond its accepted principles. *Revue Roumaine de Linguistique*, 55(1):19–44.

Marianne Mithun. 1992. Is basic word order universal? In Doris Payne, editor, *Pragmatics of word order flexibility*, pages 15–61. John Benjamins, Amsterdam, the Netherlands.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Frederick Newmeyer. 1992. Iconicity and generative grammar. *Language*, 68(4):756–796.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*.

Ting Qian and T Florian Jaeger. 2012. Cue effectiveness in communicatively efficient discourse production. *Cognitive science*, 36(7):1312–1336.

Günter Radden and René Dirven. 2007. *Cognitive English Grammar*. John Benjamins, Amsterdam, the Netherlands.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *International Conference on Learning Representations*.

Laura E. de Ruiter, Anna L. Theakston, Silke Brandt, and Elena V.M. Lieven. 2018. Iconicity affects children's comprehension of complex sentences: The role of semantics, clause order, input and individual differences. *Cognition*, 171:202 – 224.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112, Montreal, Canada.

Harry Tily, Michael C Frank, and T. Florian Jaeger. 2011. The learnability of constructed languages reflects typological patterns. pages 1364–1369.

E. C. Traugott. 1972. *IA history of English syntax*. New York: Holt, Rinehart and Winston.

George Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Boston, MA.

Supplementary Material: Word-order biases in deep-agent emergent communication

Rahma Chaabouni[1,2], Eugene Kharitonov[1], Alessandro Lazaric[1],
Emmanuel Dupoux[1,2] and Marco Baroni[1,3]

[1]*Facebook A.I. Research*
[2]*Cognitive Machine Learning (ENS - EHESS - PSL Research University - CNRS - INRIA)*
[3]*ICREA*
{rchaabouni,kharitonov,lazaric,dpx,mbaroni}@fb.com

1 Deriving the training loss for the Speaker role

In the Speaker role, each input trajectory $\boldsymbol{t}_j$ maps onto a set of utterances $\{\boldsymbol{u}\}_j$. We want to train an agent such that, given $\boldsymbol{t}_j$, it generates all the corresponding utterances $\{\boldsymbol{u}\}_j$ uniformly. To do that, we follow the "Naïve" approach from Jin and Ghahramani (2003).

Given an input $\boldsymbol{t}$, the Seq2Seq model defines a distribution over the output sequences, $p_\theta(\boldsymbol{u}|\boldsymbol{t}_j)$. The KL-divergence (Kullback, 1997) $\mathcal{D}(P||p_\theta)$ between the uniform distribution $P(\boldsymbol{u}|\boldsymbol{t}_j)$ over the target utterances $\{\boldsymbol{u}\}_j$ and the output distribution of the agent, $p_\theta(\boldsymbol{u}|\boldsymbol{t}_j)$, is:

$$\begin{aligned}\mathcal{D}(P||p_\theta) =& \mathbb{E}_{\boldsymbol{u}\sim P(\boldsymbol{u}|\boldsymbol{t}_j)}\left[\log \frac{P(\boldsymbol{u}|\boldsymbol{t}_j)}{p_\theta(\boldsymbol{u}|\boldsymbol{t}_j)}\right] \\ =& E - \mathbb{E}_{\boldsymbol{u}\sim P(\boldsymbol{u}|\boldsymbol{t}_j)} \log p_\theta(\boldsymbol{u}|\boldsymbol{t}_j)\end{aligned} \tag{1}$$

with E independent from θ. Hence, finding θ that minimizes $\mathcal{D}(P||p_\theta)$ is equivalent to minimization of $\mathcal{L}'$:

$$\mathcal{L}'(\boldsymbol{t}_j) = -\mathbb{E}_{\boldsymbol{u}\sim P(\boldsymbol{u}|\boldsymbol{t}_j)} \log p_\theta(\boldsymbol{u}|\boldsymbol{t}_j) \tag{2}$$

Next, assuming that the target set of utterances $\{\boldsymbol{u}_j\}$ has n_j elements,

$$\mathbb{E}_{\boldsymbol{u}\sim P(\boldsymbol{u}|\boldsymbol{t}_j)} \log p_\theta(\boldsymbol{u}|\boldsymbol{t}_j) = \frac{1}{n_j} \sum_{\boldsymbol{u}\in\{\boldsymbol{u}\}_j} \log p_\theta(\boldsymbol{u}|\boldsymbol{t}_j) \tag{3}$$

We expand $p_\theta(\boldsymbol{u}|\boldsymbol{t}_j)$ by iterating words u_k in $\boldsymbol{u}$, as in Section 3.2 of the main text:

$$\log p_\theta(\boldsymbol{u}|\boldsymbol{t}_j) = \sum_{k=1}^{|\boldsymbol{u}|} \log p_\theta(u_k|u_{k-1}, \boldsymbol{h}_{j,k}) \tag{4}$$

By combining Eq. (3) and Eq. (4), we obtain:

$$\mathcal{L}'(\boldsymbol{t}_j) = -\frac{1}{n_j} \sum_{\boldsymbol{u}\in\{\boldsymbol{u}\}_j} \sum_{k=1}^{|\boldsymbol{u}|} \log p_\theta(u_k|u_{k-1}, \boldsymbol{h}_{j,k}) \tag{5}$$

After aggregation over all trajectories in the dataset, we obtain the full loss that coincides with Eq. (4) in the main text:

$$\begin{aligned}&\sum_j \mathcal{L}'(\boldsymbol{t}_j) = \\ &\quad -\sum_j \frac{1}{n_j} \sum_{\boldsymbol{u}\in\{\boldsymbol{u}\}_j} \sum_{k=1}^{|\boldsymbol{u}|} \log p_\theta(u_k|u_{k-1}, \boldsymbol{h}_{j,k})\end{aligned} \tag{6}$$

This concludes our derivation of the loss used for the Speaker role. Finally, we note that this derivation provides grounding for the sub-sampling we use during the training, as it corresponds to getting a Monte-Carlo estimate of the expectation in Eq. (2) over n samples, instead of the full support of the distribution.

2 Examples of trajectories and utterances

In Table 1, we exemplify how trajectories with two, three, and five segments are represented by utterances in free- and fixed-order languages with and without markers. Note how the free-order language without markers is extremely ambiguous, as the utterances do not encode the execution order of the corresponding trajectories.

Tables 2 and 3 give examples of how trajectories are represented by utterances in the local and long-distance languages, as well as in example controls. The control languages are constructed to enable a fairer comparison between the local and long-distance setups. The full long-distance language has more possible utterances per trajectory than the local one (the latter is a subset of the former). Their controls, however, have the same number of utterances. Practically, to construct one local control language, we sample 24 distinct utterance templates (that is, phrase orders) out of 48 from the full language. We use

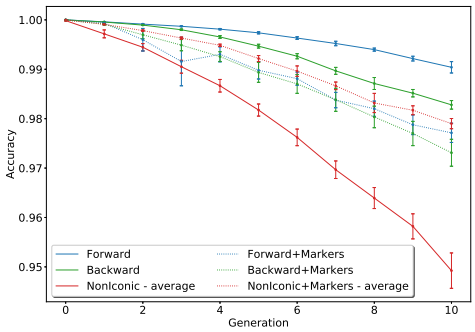


Figure 1: **Iterated learning with fixed-order languages.** Mean test set attention Speaker accuracy at the end of training over 10 generations. Error bars represent standard deviation over 5 random seeds. The NonIconic-average curve pools measurements for 3 non-iconic languages, each with 5 runs.

3 different local control languages by sampling a different subset each time. Table 2 exemplifies one of these control languages. To construct one long-distance control language, we also sample 24 distinct utterance templates from the full long-distance language (out of 144 possible utterances). The latter sampling maintains the proportion of local and long-distance constructions of the full long-distance language ($1/3$ vs. $2/3$). Again, we sample 3 different long-distance controls. One of them is exemplified in Table 3.

3 Iterated Learning of fixed word order languages

In this section, we use the iterated learning paradigm to analyze Seq2Seq networks biases toward iconic languages. We expect agents with less natural non-iconic languages to either converge to more iconic ones or diverge with low communication accuracy. We simulate 10 generations repeating the process with 5 different initialization seed and report the average of communication accuracy of each generation in Fig. 1. We observe in speaker mode a (relatively small) decrease in accuracy across generations, which, importantly, affects the most natural language (forward iconic without markers) the least, and the most difficult language (non-iconic without markers) the most. Thus, we observe a (weak) tendency for the attention agent to yield to the expected natural pressures in terms of iconic order.

References

Rong Jin and Zoubin Ghahramani. 2003. Learning with multiple labels. In *Advances in neural information processing systems*, pages 921–928.

Solomon Kullback. 1997. *Information theory and statistics*. Courier Corporation.

Table 1: Example utterances associated to trajectories of different lengths in the fixed- and free-order languages we consider.

Trajectory (two segments):	LEFT DOWN DOWN		
With markers	**Forward-iconic** first left 1 second down 2 **Non-iconic 2** second down 2 first left 1	**Reverse-iconic** second down 2 first left 1 **Non-iconic 3** first left 1 second down 2	**Non-iconic 1** first left 1 second down 2 **Free-order** first left 1 second down 2 second down 2 first left 1
Without markers	**Forward-iconic** left 1 down 2 **Non-iconic 2** down 2 left 1	**Reverse-iconic** down 2 left 1 **Non-iconic 2** left 1 down 2	**Non-iconic 1** left 1 down 2 **Free-order** left 1 down 2 down 2 left 1
Trajectory (three segments):	LEFT LEFT LEFT DOWN DOWN UP UP UP		
With markers	**Forward-iconic** first left 3 second down 2 third up 3 **Non-iconic 2** second down 2 third up 3 first left 3	**Reverse-iconic** third up 3 second down 2 first left 3 **Non-iconic 3** first left 3 second down 2 third up 3	**Non-iconic 1** first left 3 third up 3 second down 2 **Free-order** first left 3 second down 2 third up 3 first left 3 third up 3 second down 2 second down 2 third up 3 first left 3 second down 2 first left 3 third up 3 third up 3 first left 3 second down 2 third up 3 second down 2 first left 3
Without markers	**Forward-iconic** left 3 down 2 up 3 **Non-iconic 2** down 2 up 3 left 3	**Reverse-iconic** up 3 down 2 left 3 **Non-iconic 3** left 3 down 2 up 3	**Non-iconic 1** left 3 up 3 down 2 **Free-order** left 3 down 2 up 3 left 3 up 3 down 2 down 2 up 3 left 3 down 2 left 3 up 3 up 3 left 3 down 2 up 3 down 2 left 3
Trajectory (five segments):	DOWN RIGHT RIGHT UP UP UP RIGHT LEFT LEFT		
With markers	**Forward-iconic** first down 1 second right 2 third up 3 fourth right 1 fifth left 2 **Non-iconic 2** second right 2 third up 3 fifth left 2 fourth right 1 first down 1	**Reverse-iconic** fifth left 2 fourth right 1 third up 3 second right 2 first down 1 **Non-iconic 3** fourth right 1 first down 1 second right 2 fifth left 2 third up 3	**Non-iconic 1** first down 1 fourth right 1 third up 3 second right 2 fifth left 2 **Free-order** first down 1 second right 2 third up 3 fourth right 1 fifth left 2 first down 1 second right 2 third up 3 fifth left 2 fourth right 1 ... fifth left 2 fourth right 1 third up 3 second right 2 first down 1
Without markers	**Forward-iconic** down 1 right 2 up 3 right 1 left 2 **Non-iconic 2** right 2 up 3 left 2 right 1 down 1	**Reverse-iconic** left 2 right 1 up 3 right 2 down 1 **Non-iconic 3** right 1 down 1 right 2 left 2 up 3	**Non-iconic 1** down 1 right 1 up 3 right 2 left 2 **Free-order** down 1 right 2 up 3 right 1 left 2 down 1 right 2 up 3 left 2 right 1 ... left 2 right 1 up 3 right 2 down 1

Table 2: Example utterances associated to one trajectory by the local language and one of its controls.

Trajectory (three segments): DOWN DOWN DOWN LEFT LEFT LEFT UP		
	Local	
first down first 3 second left second 3 third up third 1	first down first 3 second left second 3 third 1 third up	first down first 3 second 3 second left third up third 1
first down first 3 second 3 second left third 1 third up	first down first 3 third up third 1 second left second 3	first down first 3 third up third 1 second 3 second left
...	...	...
third 1 third up first 3 first down second left second 3	third 1 third up first 3 first down second 3 second left	third 1 third up second left second 3 first down first 3
third 1 third up second left second 3 first 3 first down	third 1 third up second 3 second left first down first 3	third 1 third up second 3 second left first 3 first down
	Local control (one of three)	
first down first 3 second left second 3 third up third 1	first down first 3 third up third 1 second left second 3	second 3 second left third up third 1 first down first 3
second 3 second left third 1 third up first down first 3	third up third 1 second left second 3 first down first 3	second left second 3 first 3 first down third 1 third up
second left second 3 first down first 3 third up third 1	first 3 first down third up third 1 second 3 second left	third up third 1 first 3 first down second 3 second left
third up third 1 first down first 3 second left second 3	third 1 third up second left second 3 first 3 first down	third 1 third up second 3 second left first 3 first down
second 3 second left first 3 first down third up third 1	second 3 second left third up third 1 first 3 first down	first 3 first down third up third 1 second left second 3
second 3 second left first down first 3 third 1 third up	third 1 third up first 3 first down second 3 second left	first down first 3 third 1 third up second left second 3
third up third 1 second 3 second left first 3 first down	third up third 1 first down first 3 second 3 second left	third up third 1 second left second 3 first 3 first down
first 3 first down second 3 second left third up third 1	first 3 first down second left second 3 third up third 1	second left second 3 first down first 3 third 1 third up

Table 3: Example utterances associated to one trajectory by the long-distance language and one of its controls.

Trajectory (three segments): DOWN DOWN DOWN LEFT LEFT LEFT UP		
Long-distance		
local utterances		
first down first 3 second left second 3 third up third 1	first down first 3 second left second 3 third 1 third up	first down first 3 second 3 second left third up third 1
first down first 3 second 3 second left third 1 third up	first down first 3 third up third 1 second left second 3	first down first 3 third up third 1 second 3 second left
...	...	...
third 1 third up first 3 first down second left second 3	third 1 third up first 3 first down second 3 second left	third 1 third up second left second 3 first down first 3
third 1 third up second left second 3 first 3 first down	third 1 third up second 3 second left first down first 3	third 1 third up second 3 second left first 3 first down
long-distance utterances		
first down first 3 second left third up third 1 second 3	first down first 3 second left third 1 third up second 3	first down first 3 second 3 third up third 1 second left
first down first 3 second 3 third 1 third up second left	first down first 3 third up second left second 3 third 1	first down first 3 third up second 3 second left third 1
...	...	...
third 1 third up first 3 second left second 3 first down	third 1 third up first 3 second 3 second left first down	third 1 third up second left first down first 3 second 3
third 1 third up second left first 3 first down second 3	third 1 third up second 3 first down first 3 second left	third 1 third up second 3 first 3 first down second left
Long-distance control (one of three)		
local utterances		
first down first 2 second left second 3 third up third 1	second 3 second left first 2 first down third up third 1	third up third 1 first down first 2 second left second 3
third 1 third up first down first 2 second 3 second left	second left second 3 first down first 2 third 1 third up	second 3 second left third 1 third up first down first 2
second 3 second left third up third 1 first 2 first down	second left second 3 first down first 2 third up third 1	
long-distance utterances		
first 2 first down third up second 3 second left third 1	third 1 second 3 second left third up first down first 2	first 2 third up third 1 first down second left second 3
first 2 second 3 second left first down third 1 third up	second 3 third up third 1 second left first 2 first down	second 3 first down first 2 second left third 1 third up
second 3 second left first down third 1 third up first 2	third 1 second left second 3 third up first 2 first down	second left second 3 third up first 2 first down third 1
third 1 third up first 2 second 3 second left first down	first down second left second 3 first 2 third 1 third up	third 1 first down first 2 third up second left second 3
third up third 1 second left first 2 first down second 3	third up third 1 second 3 first down first 2 second left	third 1 third up second left first 2 first down second 3
	first 2 first down third 1 second 3 second left third up	

Chapter 4

Semantic Categorization - Color Naming

Eleanor Rosch suggested in 1999 that systems of linguistic categories are those that "*provide maximum information with the least cognitive effort*" [98]. In other words, we partition our semantic space under the two competing pressures of (1) maximizing the transmission of information, i.e., accuracy and, (2) minimizing our cognitive load, i.e., complexity. Optimizing this complexity/accuracy trade-off is also referred to as *efficiency* [39].

This view had much influence on the development of semantic theories in linguistics and several studies introduced distinct theoretical frameworks to formalize it [96, 95, 58]. However, the definitions of language complexity and accuracy differ across studies and domains. For example, while Regier and colleagues define the complexity of a system as the number of distinct terms [96], Kemp and colleagues define it as the smallest number of rules needed to generate all terms in the system (which can be smaller or larger than the number of distinct terms as exemplified in the reference paper) [58]. More recently, Zaslavsky and colleagues proposed an explicit information-theoretic framework to formalize the complexity/accuracy trade-off [118]. Their study demonstrates that human languages are efficient in the sense of the information bottleneck [109].

Yet, while the generality of the latter framework is appealing, it does not address

why this specific information bottleneck optimization arises. In this chapter, we rely on the information bottleneck principle and use communicating NNs to investigate this question focusing on the well-studied color domain. We first show that NN color-naming systems strikingly resemble the human ones. Second, we exploit the flexibility afforded by NNs to explore which factors lead them to human-like color naming. Finally, we found that both NN systems' simplicity and efficiency crucially relate to the bottleneck effect that a *discrete* channel imposes on the information flow.

Communicating artificial neural networks develop efficient color-naming systems

Rahma Chaabouni[a,b,1], Eugene Kharitonov[a], Emmanuel Dupoux[a,b], and Marco Baroni[a,c]

[a]Facebook AI Research, 75002 Paris, France; [b]Cognitive Machine Learning, ENS - EHESS - PSL Research University - CNRS - INRIA, 75012 Paris, France; and [c]Institució Catalana de Recerca i Estudis Avançats, 08010 Barcelona, Spain

Edited by Melanie Mitchell, Santa Fe Institute, Santa Fe, NM, and accepted by Editorial Board Member Michael S. Gazzaniga February 4, 2021 (received for review August 5, 2020)

Words categorize the semantic fields they refer to in ways that maximize communication accuracy while minimizing complexity. Focusing on the well-studied color domain, we show that artificial neural networks trained with deep-learning techniques to play a discrimination game develop communication systems whose distribution on the accuracy/complexity plane closely matches that of human languages. The observed variation among emergent color-naming systems is explained by different degrees of discriminative need, of the sort that might also characterize different human communities. Like human languages, emergent systems show a preference for relatively low-complexity solutions, even at the cost of imperfect communication. We demonstrate next that the nature of the emergent systems crucially depends on communication being discrete (as is human word usage). When continuous message passing is allowed, emergent systems become more complex and eventually less efficient. Our study suggests that efficient semantic categorization is a general property of discrete communication systems, not limited to human language. It suggests moreover that it is exactly the discrete nature of such systems that, acting as a bottleneck, pushes them toward low complexity and optimal efficiency.

efficiency of human language | language emergence in artificial neural networks | color-naming systems

Words partition our world into semantic categories. Converging evidence indicates that, while these categories differ widely across languages, they are shaped by universal constraints (1–3). In particular, it has been suggested that semantic categorization evolves to support efficient communication (4). Humans develop naming systems to talk about their experience under two competing pressures: "accuracy maximization" (words should encode precise information about their referents) and "complexity avoidance" (preventing unwieldy languages). At an extreme, a maximally accurate system would have a different term for each perceptual or mental experience. At the other, a maximally simple system would use only one term to refer to all experiences, completely hindering communication.

Actual human naming systems are efficient in the sense that they optimize the accuracy/complexity trade-off. More generally, since the foundational work of Zipf (5), a similar trade-off between precision and simplicity has been observed in many areas of language (6).

Zaslavsky et al. (7) formalized the measurement of naming-system efficiency within the general information–theoretic framework of the Information Bottleneck (IB) (8) (see also the closely related rate-distortion theory framework in ref. 9). A system is deemed efficient if it reaches the maximum possible accuracy for a given complexity. In the IB framework, both accuracy and complexity are computed in a communication model where an idealized Speaker aims to communicate a meaning to an idealized Listener. Accuracy is then inversely related to the cost of a misinterpreted meaning, while complexity measures the quantity of information needed to convey the meaning. The IB efficiency of a system is effectively visualized in plots (see Fig. 3). The black curve in Fig. 3 represents the theoretical limit: no system of a certain complexity (horizontal axis) can have accuracy (vertical axis) above the curve. Hence, according to IB, a system is optimal if it lies on the curve. Equipped with this framework, Zaslavsky et al. (7) demonstrated that color-naming systems (4, 10, 11) are notably close to the theoretical limit and hence efficient in a quantifiable way.

IB theory is agnostic about where on the theoretical-limit curve a system should lie. Degenerate systems lying at the extremes of the curve, and expressing each referent with a different term or all referents with a unique term, are also efficient according to this theory. However, such systems are not attested. Instead, real color-naming systems approximate a small range of possible optimal solutions, avoiding the extremes, and in particular high-complexity trade-offs (7). This avoidance of complexity extremes has been observed more broadly in studies of categorization and naming across many semantic domains (4, 12–14).

We study the efficiency of color naming from a different perspective. We compare natural language systems with those emerging from the interaction of modern neural networks (NNs) faced with a color-communication task. Artificial NNs trained with deep-learning methods (15) have recently been used to study human (neuro)cognition in many fields (e.g., refs. 16–19), including color naming (20, 21). Traditional simulations in cognitive science are specifically designed to assess how certain factors of interest affect system behavior by developing ad hoc models, an approach illustrated by Baronchelli et al. (22) and Loreto et al. (23), in the domain of color naming, and

Significance

Color names in human languages are organized into efficient systems optimizing an accuracy/complexity trade-off. We show that artificial neural networks trained with generic deep-learning methods to play a color-discrimination game develop color-naming systems whose distribution on the accuracy/complexity plane is strikingly similar to that of human languages. We proceed to show that efficiency and narrow complexity crucially depend on the discrete nature of communication, acting as an information bottleneck on the emergent code. This suggests that efficient categorization of colors (and possibly other semantic domains) in natural languages does not depend on specific biological constraints of humans, but it is instead a general property of discrete communication systems.

Author contributions: R.C., E.K., E.D., and M.B. designed research; R.C. and M.B. performed research; R.C., E.K., and M.B. analyzed data; and R.C. and M.B. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. M.M. is a guest editor invited by the Editorial Board.

[1]To whom correspondence may be addressed. Email: chaabounirahma@gmail.com.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2016569118/-/DCSupplemental.

Published March 17, 2021.

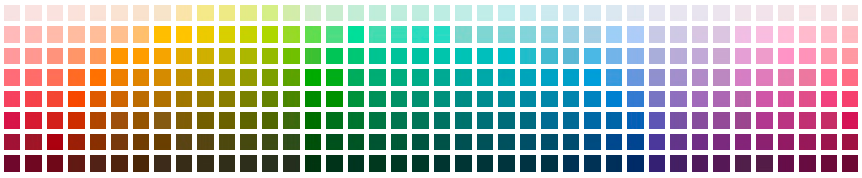


Fig. 1. The 330 WCS color chips. Rows correspond to equally spaced lightness values and columns to equally spaced Munsell hues. Each stimulus is at the maximum available saturation for that hue/lightness combination.

applied by Carr et al. (24) to the study of complexity/accuracy trade-offs in semantic categorization. Deep networks, however, are high-performance general-purpose learners, independently developed for engineering purposes, with no claims of cognitive plausibility concerning their architecture or learning process. In this respect, they might be best seen as complex "animal models" (25, 26). The main interest lies in whether the emergent behavior of these powerful mechanisms mirrors nontrivial properties of human behavior (27). If it does, we can entertain the intriguing hypothesis that the specific converging human and deep-network patterns we observe have common roots. We can moreover directly intervene on the artificial organisms (more easily so than we can on humans), in order to causally assess how different components affect their emergent behavior.

Specifically, we show that, when two deep learning-trained NNs play a simple color discrimination game, they develop naming systems that closely match the distribution of human languages on the IB plane, showing both efficiency maximization and complexity control (Fig. 3). The use of human-like artificial systems emerges without imposing ad hoc constraints favoring efficiency or limiting complexity on the training procedure. Having observed the systematic emergence of efficiency and complexity reduction in the NN systems, we proceed to test the hypothesis that these properties crucially depend on the bottleneck imposed by the discrete communication channel. Indeed, as we let NNs exchange messages that are increasingly more continuous, their naming systems become more complex, and, eventually, no longer efficient. Varying the degree of color-discrimination granularity required to play the game affects the complexity of the emergent systems, but not efficiency, and only within the range of attested human variation. NN capacity only affects the complexity of the system in function of discreteness of communication.

The emergence of efficient and reasonably simple semantic categorization is not specific to human language but might generally arise in cognitive devices exchanging discrete messages about their world. Discreteness of communication plays a central role in the emergence of efficient and low-complexity naming systems among our artificial agents, raising intriguing questions about the role of discreteness in human language.

Color-Naming Task

Stimuli. Following prior work (4, 7, 28), we use the World Color Survey (WCS). The WCS contains the names of 330 color chips (Fig. 1) in 110 languages of nonindustrialized societies (29). We represent each color stimulus as a three-dimensional vector in CIELAB space (a color space designed to approximate human vision). In particular, we measure color similarity based on Euclidean distance in CIELAB, as it correlates with human perceptual sensitivity (7).

Discrimination Game. We implement a classic discrimination game (30) played by 2 NN agents, Speaker and Listener. Speaker receives a target color c_t from the palette and sends one word w from its vocabulary V to Listener. Speaker chooses the word from a fixed vocabulary of size $|V| = 1{,}024$. As $|V|$ is larger than the number of colors (330), it is always possible, in principle, for Speaker to use a unique word to denote each distinct color. Given w and two distinct colors, c_t and a distractor c_d, Listener must predict the target. The agents succeed if Listener guesses the correct target (as in Fig. 2).

As in previous work (4, 28), we assume a uniform prior distribution $p(c)$ over target colors. In *SI Appendix, Supporting Information Text, 3. Salience-Weighted Source Distribution*, we test an alternative nonuniform prior (31), and the results still hold.

The game is implemented in EGG (32). Further details are in *Materials and Methods*.

Discriminative Need. Despite the presence of universal tendencies (10, 33, 34), color-naming variance is also observed (35, 36). Prior studies hypothesized that such variance depends on distinct frequencies of occurrence of colors across communities (31, 37). In lack of data capturing these differences, we explore a complementary source of variation, that is easier to model computationally. We hypothesize that different cultures have different discriminative needs. Intuitively, highly industrialized societies might need to distinguish between subtly different color shades characterizing different goods, whereas nonindustrialized societies can rely on coarser distinctions. As indirect evidence, Gibson et al. (31) reported that, in the nonindustrialized Tsimané community, color terms are "only used to discriminate between familiar artificial objects." Since in a nonindustrialized community, there is relatively low variety of artificial objects, discrimination need will be low. In English, instead, speakers systematically use color terms to discriminate between objects of all kinds (31).[†]

Concretely, we define discriminative need as the minimum allowed Euclidean distance between targets and distractors in CIELAB space. Agents trained with small minimum target-distractor distance, $dist_{min}$, simulate communities with high discriminative need; the ones trained with large $dist_{min}$ represent communities with low discriminative need. We quantify $dist_{min}$ in terms of the nth percentile in the list of pairwise distances between the 330 distinct color chips. For example, with percentile = 50, for a given target color c_i, a distractor c_j is sampled uniformly among candidate colors such that

$$dist(c_i, c_j) \geq med\left(\{dist(c_k, c_l); k, l \in \{1..330\}, k > l\}\right), \quad \textbf{[1]}$$

where med is the median function and $dist$ is the Euclidean distance in CIELAB space. Note that larger percentiles correspond to games requiring less granular discrimination. We provide examples in *SI Appendix, Supporting Information Text, 1. Example of Nearest Target-Distractors for Different Percentiles*.[‡]

Speaker Word Distributions. Just like in natural language, we allow fuzzy naming: the same color might be, in different occasions,

[†] Humans can also handle varying contextual discrimination needs by producing longer or shorter phrasal descriptions (38), a strategy we are not modeling here.

[‡] *SI Appendix, Supplementary Information Text, 2. Random Sampling of Distractors* demonstrates that highly efficient systems also emerge when no percentile is imposed, although the latter never reach our threshold for minimum game accuracy (95%).

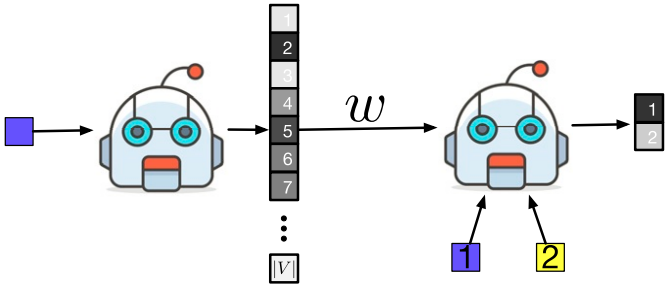


Fig. 2. A successful round of the discrimination game. A chip *c* is drawn from a uniform distribution and fed to Speaker. Speaker outputs a probability distribution $p(W|c)$ over its vocabulary of size $|V|$. Here, a probability is mapped to a color according to a gray gradient (with darker colors representing higher probabilities). A word *w* is sampled from $p(W|c)$ and fed to Listener. Finally, Listener—given *w*, the target chip (in position 1 in this illustration), and a distractor chip (in position 2 in this illustration)—assigns a probability to both positions, representing its guess about the position of the target (in this illustration, Listener correctly assigns a higher probability to the target position).[¶]

denoted by different words. To estimate the probability distribution $P(w|c)$ associated to a color chip c, we sample 25 words with replacement from Speaker after convergence.[§] For instance, since Speaker's outputs form a categorical distribution over V, if this distribution is a Dirac, the resulting set of 25 samples will correspond to a unique word. At the other extreme, if Speaker has no confidence about c's category, we might get 25 distinct words equiprobably naming c.

Evaluating the Accuracy/Complexity Trade-Off. To compareNN and human naming systems, we adopt the communication model of Zaslavsky et al. (7), keeping the same notation. U represents the set of world's objects, in our case, the set of colors; W represents the set of words; and M represents the set of Speaker's meanings. We assume that a NN Speaker, similarly to what is conjectured for humans (4, 7), internally represents each target color chip c as a Gaussian distribution $m \in M$ over U centered at c and defined upon CIELAB color similarity. That is, for a given target chip c, Speaker constructs an internal representation $m(c)$ reflecting its belief about the color chip it wishes to communicate to Listener. The Gaussian $m(c)$, of mean c, is then only parameterized by variance σ^2, that informs about the Speaker's (un)certainty about its belief. Concretely, an $m(c)$ with low variance, reflecting a certain belief, would only cover c and few neighboring chips according to the CIELAB space (e.g., slightly darker and lighter chips). Similarly to Zaslavsky et al. (7), we set $\sigma^2 = 64$ for all target chips. Note that M is only introduced to compute the accuracy and complexity measures below, and it plays no direct role in the discrimination game.

In the framework by Zaslavsky et al. (7), the complexity of a naming system is quantified by the number of bits of information required for expressing the intended meanings. As shown by Zaslavsky et al. (7), this is measured by the mutual information, $I(M; W)$ between M and W.

Also following Zaslavsky et al. (7), we use $I(U; W)$ to measure the accuracy of a naming system. The latter measure is inversely related to the Kullback–Leibler divergence between Speaker and Listener meanings. That is, the better Listener is at reconstructing Speaker's meaning, the larger $I(U; W)$ is.

The theoretically optimal trade-offs between complexity and accuracies are approximated by minimizing the IB objective function:

$$I(M; W) - \beta I(U; W) \;\; s.t.\; \beta \geq 1, \qquad [2]$$

where β is the trade-off parameter determining the relative weight a system will attribute to complexity avoidance vs. accuracy maximization. Both complexity and accuracy are quantified by mutual information terms. However, the IB objective minimizes the first term (lowering complexity) and maximizes the second (increasing accuracy; note the minus sign preceding the second term in Eq. **2**), two constraints that will be in contrast.

To minimize Eq. **2** for a fixed β, we look for the set $\{P(w_i|c_j)\}_{i,j}$, where $j \in [1, 330]$ and $i \in [1, K]$, with K a variable to optimize. To get the theoretical-limit curve shown in Fig. 3, we repeat this procedure for each β, as described in *Materials and Methods*.[#] Refer to Zaslavsky et al. (in particular, *Bounds on Semantic Efficiency* in the main text of ref. 7 and *SI Appendix*, section S1.3 in ref. 7) for more details about definitions and derivations.

The farther a system is to the theoretical-limit curve, the less efficient it is. To quantify the inefficiency of a system s, characterized as a point on the accuracy/complexity plane (Fig. 3), we introduce the *Inef* score:

$$Inef(s) = \min_{\beta}\{\|s - s^*_\beta\|^2 \;\; s.t.\; \beta \geq 1\}, \qquad [3]$$

where s^*_β are the coordinates of the optimal naming system (on the theoretical-limit curve) for a fixed β.

Experiments and Results

Human vs. NN Naming Systems. To simulate communities with different needs, we run the discrimination game varying minimum target-distractor distance, defined in terms of percentile of nearest distractor (see *Discriminative Need*). With percentile < 20, target-distractor pairs are too close, and agents fail to converge. Above percentile = 80, there are no distractors sufficiently distant to any given target. We hence played the game with $percentile \in \{20, 30, 40, 50, 60, 70, 80\}$, resulting in 60 successful games in total. Control experiments are in *SI Appendix, Supplementary Information Text, 5. Encouraging the Emergence of a Two-Word System during Training*.

Looking at how human and NN naming systems spread along the IB line in Fig. 3, we can make two striking observations. First, NN systems lie near the theoretical IB limit just like human languages do. *SI Appendix, Supplementary Information Text, 4. Efficiency: Comparing Human vs. NN Systems, and Actual vs. Rotated Systems* shows that NN system inefficiency (Eq. **3**) falls within the human range. Second, both human and NN systems lie on a narrow segment of the curve. While this segment does not include the minimal values, it is still clearly tilted toward the low-complexity end of the curve. Note that minimum complexity would be achieved by a system with a single color term. As this makes no sense, we do not expect minimum-complexity systems to emerge. Intriguingly, neither WCS nor NN systems include two-word codes (which are exceedingly rare in natural languages in general) (39). *SI Appendix, Supplementary Information Text, 5. Encouraging the Emergence of a Two-Word System during Training* shows that, even when we manipulate the game so that agents could achieve perfect discrimination with two words only, they minimally converge to a three-word system.

[§]A majority of WCS languages contains names elicited from 25 speakers, leading to comparable a sample size for $P(w|c)$ estimation.

[¶]Target and distractor positions are randomly shuffled at each round to prevent Listener from relying on position to succeed at the game.

[#]We only optimize Eq. **2** to calculate the IB bound. NN training is completely distinct and independent from this calculation.

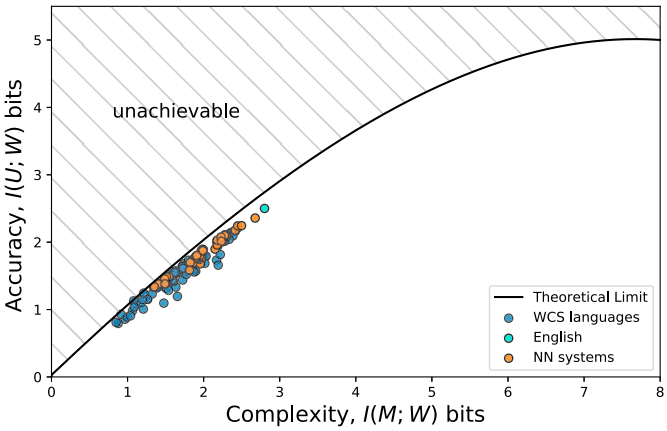


Fig. 3. Human (blue circles) and NN (orange circles) color-naming systems on the information plane. English (light blue circle) is not in WCS, but it is approximated relying on Zaslavsky et al. (*SI Appendix*, figure S7 in ref. 7). The IB curve (black line) defines the theoretical limit on accuracy given complexity. All color-naming systems achieve near-optimal efficiency.

We have no explanation for why two-word systems are avoided. Still, both WCS and NN systems are clearly coming much closer to the lower end of the complexity scale than to the upper bound.[‖]

In sum, standard NNs trained on the discrimination game develop systems that support efficient communication (i.e., are close to the IB curve) while preferring low complexity, similarly to human color-naming systems. Our focus here is on the IB trade-off. However, the way in which NN systems accomplish this trade-off is not radically different from that of human languages. *SI Appendix*, *Supplementary Information Text*, *10. Direct Comparison of Color Space Partitions* presents a detailed comparison of color partitioning in human and NN naming systems, highlighting partial differences but also important commonalities, in particular, in terms of the convexity of regions corresponding to distinct color names (see also *SI Appendix*, Fig. S12 for qualitative comparison between both systems).

Effect of discriminative need. Fig. 3 shows the NN systems resulting from exploring the full range of possible percentile values (the parameter controlling discriminative need). While all systems are efficient, we observe some variability in complexity (within the $[0.84, 2.8]$ range), that might be due to different discriminative needs. This is confirmed by Fig. 4, which shows NN naming system complexity in function of percentile. Smaller percentile values (requiring more granular discrimination) make systems more complex. Still, this trend is gradual with no significant pairwise differences, suggesting the need for distant discriminative needs to observe a significant difference in systems' complexity. Furthermore, NN systems' complexity remains within human-range complexity when exploring the full range of percentile values. Interestingly, Fan et al. (14) showed, in the context of visual communication, that humans are also sensitive to discriminative need and adapt the complexity of their communicative system accordingly.

Thus, discriminative need (or related environmental/societal pressures to make more/less granular distinctions) could account for the range of complexity variation we observe in NN and human naming systems (and that might be somewhat underestimated by the WCS sample). However, alone, it does not explain why the range of observed systems is so narrow.

Preference for low complexity. Both human and NN systems show much lower complexity than what could be found in an optimal naming system by systematically varying the trade-off parameter $\beta \in [1, +\infty[$.[**] The attested systems all occur within a small segment corresponding to $\beta \in]1, 1.14]$.

One might conjecture that more complex codes do not evolve simply because the attested ones are sufficiently granular to support all required discriminations. For our NN agents at least, this is not the case, as they systematically fail to achieve 100% success in the discrimination game, which would instead be possible with more complex systems. To illustrate the latter, we generate additional naming systems by partitioning the color space using the "fuzzy c-means" (FCM) soft clustering algorithm (40), treating cluster labels as color names. We obtain different systems by varying the number-of-clusters hyperparameter. We then play the discrimination game with Speakers and Bayesian Listeners that use $p(w|c)$ distributions derived from the soft clustering solutions.

The FCM-based agents can reach 100% communication success at all percentiles. However, this comes at the cost of higher complexity. Table 1 compares, for each percentile, the 100% successful FCM system with lowest complexity to the NN system with highest success rate. In all cases, NNs came up with systems that are considerably less complex but that also fail to reach perfect discrimination success.[††] We conclude that the low complexity of NN systems cannot be explained by lack of sufficient communicative pressure toward more complex solutions. We explore next other possible sources of low-complexity-preference.

Roots of Efficiency and Complexity Avoidance. Building on recent work (41), we explore the idea that the discrete nature of the communication channel acts as bottleneck on the amount of information that the agents are able to transmit, leading them to establish efficient and low-complexity naming systems. Another natural bottleneck could be agents' capacity. Perhaps, the "neural power" of our NNs does not suffice to develop more complex languages. We show next that channel discreteness plays a fundamental role in complexity reduction, whereas NN capacity only matters insofar as it allows the agents to further simplify the code in presence of a discrete channel.

Effect of channel discreteness. We fix percentile = 50 and explore different training regimes ranging from a fully discrete setup to a virtually continuous one, relying on two commonly used methods to train deep networks in language emergence scenarios (e.g., refs. 42 and 43; also see *Materials and Methods*). The REINFORCE (RF) algorithm uses fully discrete symbol transmission during both training and evaluation. The Gumbel-Softmax (GS) method is fully discrete at evaluation time, but it estimates symbol probabilities through a smooth approximation during training. At training time, discrete symbols are approximated by continuous vectors with most of the mass concentrated around a single value. The "peakiness" (and thus discreteness) of this approximation is controlled by the temperature parameter τ. The lower the τ, the peakier the vector (practically converging to a discrete "1-hot" encoding for low τs). We

[‖] NN systems are tilted toward the top of the human complexity range. This is probably an artifact of WCS' focus on nonindustrialized societies. English, the only industrialized-society language in Fig. 3, is more complex than any NN language.

[**] In practice, distinct optimal systems are only obtained for $\beta \in [1, 2^{13}]$, as all optimal systems with $\beta > 2^{13}$ are identical and assign a unique word to each color.

[††] In the few cases in which FCM systems converged to success rates comparable to those of NN systems, FCM systems were on average less complex, suggesting that the relatively high FCM complexity we observe in Table 1 is not due to an inherent tendency of the latter to converge to high-complexity solutions.

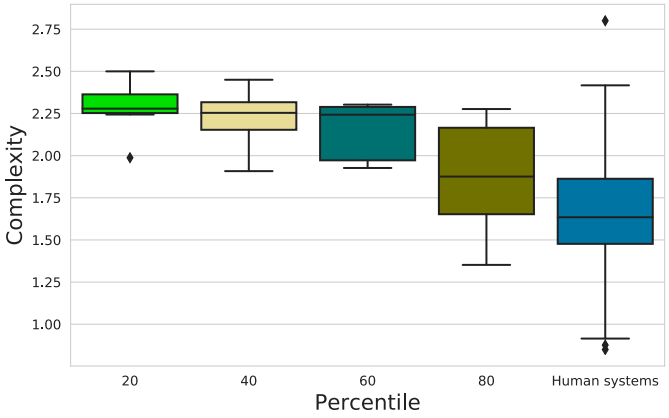


Fig. 4. Complexity distributions of NN systems across different discriminative needs (human distribution included for comparison). There is a decreasing trend in complexity when increasing percentile (P = 0.004; Kruskal–Wallis). Pairwise differences are not significant when evaluated with Bonferroni-corrected Mann–Whitney–Wilcoxon.

Table 1. Complexity and success rate (game accuracy after training) of FCM-based and NN systems in function of the game percentile parameter

Percentile	min complexity FCM	Complexity Best NN	Success rate Best NN
20	5.39	2.50	95.45%
30	4.34	2.28	96.97%
40	4.01	2.23	95.76%
50	3.75	2.68	98.79%
60	3.44	2.17	96.97%
70	3.39	2.30	97.56%
80	3.12	2.24	98.78%

FCM success rate is always 100%. For FCM, we report minimal complexity among fully successful solutions. For NN, we report complexity and success rate of the system achieving highest success rate.

explore $\tau \in \{1, 5, 10\}$, corresponding to increasingly smoother communication channels.

Settings with less smooth channels, and in particular fully discrete RF, are harder to train. Hence, we launch 60 runs for each GS setting and 180 for RF. In *SI Appendix*, *Supplementary Information Text*, *6. Discreteness and Success Rate*, we discuss the relation between channel smoothness and successful convergence, arguing that the high failure rate of more discrete settings is due to a higher complexity-reduction pressure.

Fig. 5*A* shows that agents trained with RF (thus, in the completely discrete setting) develop significantly less complex systems compared to the ones trained with GS. Within GS, lower τ (more discreteness) leads to simpler codes. With more complexity, smoother systems also become less efficient, an effect that is clear with the highest $\tau = 10$ (Fig. 5*B*).[‡‡] In *SI Appendix*, *Supplementary Information Text*, *9. How Are Color-Naming Systems (In)efficient?*, we study one concrete way in which these systems are inefficient, comparing them with complex but still efficient NN systems resulting from high discriminative need.

Effect of agent capacity. Only Speaker capacity has a significant impact on complexity and only with a discrete communication channel. Interestingly, larger Speakers further reduce the complexity of the emerging naming system (*SI Appendix*, Fig. S9). As further discussed in *SI Appendix*, *Supplementary Information Text*, *8. Impact of Agent Capacity*, a reasonable interpretation for this pattern is that, when the channel is discrete, transmitting information is difficult. Consequently, a "smarter" Speaker will use its extra computational power to come up with an encoding that allows it to transmit even less bits through the channel while maintaining reasonable accuracy. Thus, the agents' capacity experiments further confirm the importance of the discrete-channel bottleneck for complexity minimization.

Discussion

We have shown that NNs trained to play a color discrimination game develop naming systems whose distribution on the accuracy/complexity trade-off plane strikingly resembles that of human languages. We obtained this result using game success as the sole training signal, without imposing any constraint on the emergent code, except that it had to consist of single discrete symbols. A very recent study by Kågebäck et al. (21) reports that deep NN agents trained with generic techniques to play a color-naming game strike a similarly human-like complexity/accuracy trade-off, despite important differences between their game and ours (in their setup, the Listener receives only the message as input, and it has to reconstruct the color chip seen by the Speaker), different methods to derive a discrete protocol, and different factors modulating the trade-off (the complexity cline, in their experiments, depends on different amounts of noise added to the communication channel). This constitutes important converging evidence that deep-network communication tends to naturally optimize the accuracy/complexity trade-off, independently of the specifics of the simulations.

We observed, in particular, that the networks developed "low-complexity" systems, again in accordance with natural language data. We then looked for the source of this low-complexity pressure in NN systems. Building up on a recent study reporting similar results in artificial tasks (41), we showed that the presence of a discrete communication bottleneck plays a crucial role. As we relax discreteness, the emergent naming systems become complex beyond what is attested in human language, and, eventually, significantly inefficient.[§§]

In the last few years, much evidence for the efficiency of human languages in general (6) and semantic categorization in particular (4) has been accumulated. Yet, we still lack a full scientific understanding of "why" language is efficient. Our results provide two contributions relevant to this question. First, since efficiency and complexity avoidance also characterize the code evolved by communicating NNs, these factors cannot be explained away by least-effort factors specific to biological agents. Second, the fact that NNs exchange a discrete signal is crucial. Discreteness is a striking, possibly unique characteristic of human language (44, 45), often adduced as a precondition for the combinatorial infinity of expression that characterizes it (46). Our finding suggests that it might also be responsible for the efficient nature of semantic categorization (and possibly language at large). We do not have direct evidence on how the language of our ancestors became discrete and on how this affected the structure of semantic categorization. However, our computational results pave the way for experiments with contemporary humans, exploring how a continuous/discrete transition in communication

[‡‡]The trend toward higher complexity and lower efficiency continues with larger τ. However, above τ = 10, agents rarely succeed at the game, making the interpretation of results difficult.

[§§]Intriguingly, while Kågebäck et al. (21) do not explicitly discuss it, their results also point to a correlation between complexity reduction and discreteness, despite the fact that they control discreteness in a different way, that is, by injecting noise into a continuous channel.

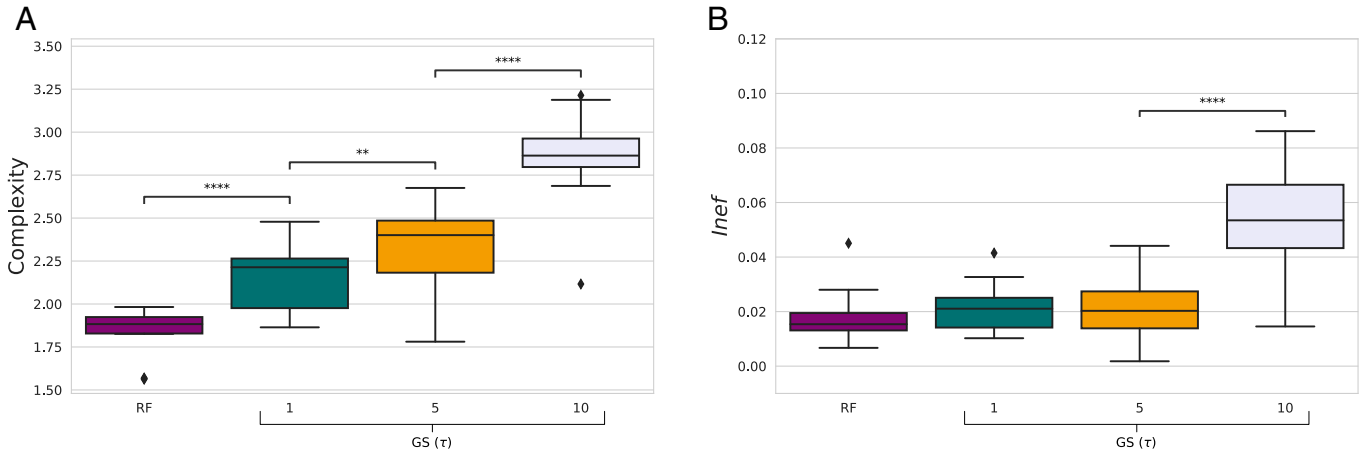


Fig. 5. Complexity and inefficiency of NN color-naming systems trained with REINFORCE or GS with different τs. Pairwise differences evaluated with Bonferroni-corrected Mann–Whitney–Wilcoxon. $^{*}P < 0.05$; $^{**}P < 0.01$; $^{***}P < 0.001$; $^{****}P < 0.0001$. Differences that are not significant are not marked.

systems affects the nature of information exchange. With human subjects, we might not have a direct equivalent of the GS temperature parameter. We can however build on a strong tradition of experimental semiotics studies using continuous signals, such as drawings, whistles, and nonconventionalized gestures, and sometimes reporting a tendency to discretize the signals as systematic communication strategies emerge (14, 47, 48). By using this framework, we should be able to design experiments that probe a causal relation between discreteness and communicative efficiency, ultimately strengthening our understanding of the roots of efficiency in language.

Materials and Methods

Human Languages. We used the WCS database (www1.icsi.berkeley.edu/wcs/). Two languages with extremely sparse information (judgments from 1 speaker only for at least some chips) were removed, resulting in 108 analyzed languages. English, which is not in WCS, was approximated based on the relevant figures from the study by Zaslavsky et al. (7).

Agent Architecture and Training. Both agents are feed-forward NNs. Speaker contains 3 hidden layers, each of size 1,000 and with leaky-ReLU (rectified linear unit) activations. For each color, the Speaker's output layer defines a Categorical distribution over its vocabulary V. Listener is modeled as a linear layer of hidden size 5. The impact of agents' capacity on results is discussed in *SI Appendix, Supplementary Information Text, 8. Impact of Agent Capacity*.

Training NNs to communicate through a discrete channel is nontrivial, as we cannot backpropagate into the Speaker through this bottleneck. We use two methods commonly employed in the deep agent language emergence literature: 1) GS relaxation (e.g., ref. 43) and 2) REINFORCE (e.g., ref. 42)) (in both cases, Listeners' gradients are obtained with standard backpropagation). We plug the obtained gradient estimates into Adam (49).

GS. Samples from the GS distribution (50, 51) approximate those from a Categorical distribution through a reparameterization trick, thus enabling gradient-based training. Let us denote $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ the standard softmax function. To get a sample that approximates an n-dimensional categorical distribution with probability $\boldsymbol{p}$, we draw $\boldsymbol{g} = [g_1, \ldots, g_n]$, where for each i, $g_i \sim \text{Gumbel}(0,1)$ and use it to calculate $\boldsymbol{y}$ such that:

$$\boldsymbol{y} = \sigma\left(\frac{\boldsymbol{g} + log\ \boldsymbol{p}}{\tau}\right), \qquad [4]$$

where τ is the temperature hyperparameter. As τ tends to 0, the samples get closer to one-hot, making communication more discrete; as $\tau \to +\infty$, the samples tend to uniform, resulting in smooth communication. At training time only, we use the relaxed samples as messages from Speaker, making the entire Speaker/Listener setup differentiable. We look at the impact of τ on Speakers' output distribution in *SI Appendix, Supplementary Information Text, 7. Effect of More/Less Discrete Training on Speakers' Output Distribution*.

Reinforce

Following Schulman et al. (52), we sample Speaker's words and estimate its gradients as follows:

$$\mathbb{E}_{\boldsymbol{i}_s, \boldsymbol{i}_l} \mathbb{E}_{\boldsymbol{w} \sim S(\boldsymbol{i}_s)} \left[\mathcal{L}(\boldsymbol{o}; \boldsymbol{t}) + sg\left(\mathcal{L}(\boldsymbol{o}; \boldsymbol{t}) - b\right) \log P_{\boldsymbol{\theta}}(\boldsymbol{w})\right], \qquad [5]$$

where $\boldsymbol{i}_a$ are agent's inputs with $a = s$ if agent is Speaker and $a = l$ if it is Listener. $\boldsymbol{o}$ denotes Listener's prediction, $\boldsymbol{t}$ denotes the ground-truth, and $\mathcal{L}$ denotes the cross-entropy loss function; sg refers to the "stop-gradient" operation. We use the standard running mean baseline b (53, 54) to reduce estimate variance. To achieve more robust convergence, we also adopt the common trick to add an entropy maximization term (55, 56) on Speaker's words. This could favor higher code complexity, which makes our low-complexity result even more striking.

When not stated otherwise, results are based on GS training with temperature τ = 1. Training consists in letting the agents play the game until their performance converge (this happens, on average, after about 6 million interactions). For each considered setting, we repeat experiments with 20 different random initializations and only focus the analysis on the successful runs. We consider a run successful if, after convergence, the agents have at least a 95% success rate. Following standard practice, success rates are computed in games in which the most likely word is deterministically sampled from the Speaker distribution.

IB Curve. We use the Agglomerative IB method (57) with $\beta_{init} = 2^{13}$. At each step of the annealing process, we evaluate the IB solution, i.e., $P(w|c)$ for each (w, c) using Iterative IB (57). The latter is an iterative method that alternates between evaluating $P(w|c)$ and $m(c)$ (Speaker's meaning for each c) until convergence. We refer readers to Zaslavsky et al. (*SI Appendix*, section 1.4 in ref. 7) for more details about this two-step process. When annealing β according to Agglomerative IB, the IB solution is initialized with the one found with the previous value of β. Optimization ends when $\beta = 1$.

Data Availability. The models reported in this paper have been deposited in GitHub (https://github.com/rahmacha/EGG).

ACKNOWLEDGMENTS. We thank Emmanuel Chemla, Thomas Brochhagen, Roger Levy, Louise McNally, Rachid Riad, Noga Zaslavsky, the PNAS reviewers, and, especially, Gemma Boleda and Diane Bouchacourt for feedback. Ted Gibson and Bevil Conway generously shared their data with us. This research was funding by Agence Nationale pour la Recherche Grants ANR-17-EURE-0017 Frontcog, ANR-10-IDEX-0001-02 PSL*, and ANR-19-P3IA-0001 PRAIRIE 3IA.

1. A. A. Goldenweiser, The principle of limited possibilities in the development of culture. *J. Am. Folklore* **26**, 259–290 (1913).

2. D. E. Brown, Human universals, human nature & human culture. *Daedalus* **133**, 47–54 (2004).

PNAS
www.pnas.org

Supplementary Information for

Communicating artificial neural networks develop efficient color-naming systems

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux and Marco Baroni

Corresponding Author name: Rahma Chaabouni
E-mail: rchaabouni@fb.com

This PDF file includes:

Supporting Information Text

1. Example of nearest target-distractors for different percentiles

Figure S1 shows examples of nearest target-distractors in games with different percentile values (recall that percentile is the parameter controlling how close target and distractor can be in CIELAB space). It clearly shows how at lower percentiles agents must discriminate between pairs that are visually close. Percentile 5 is below the level at which NNs successfully converge on a naming system. Indeed, Figure S1a shows that playing the game at this level requires distinguishing between color shades for which even a color-name-rich language such as English would have to resort to phrases. Figure S1b illustrates the percentile 20 game, which is the hardest one at which NNs succeed. We observe that some of the distinctions that need to be made (such as those between the first 3 pairs in the figure) are still quite subtle, for a single-word system at least. Figures S1c and S1d illustrate games requiring mid and low discrimination granularity, respectively.

(a) 5 **(b)** 20 **(c)** 50 **(d)** 80

Fig. S1. 5 closest target-distractor pairs for representative percentile values.

2. Random sampling of distractors

As we control discriminative need, we might worry that the efficiency of emergent systems is due to the constraints we impose, favoring solutions where color chips that are close in CIELAB space are named in the same way, rather than being a general property of discrimination-trained NNs. To address this concern, we considered an extreme version of the game, where there is *no* restriction on discriminative need, as both targets and distractors are sampled uniformly and *independently*. This is equivalent to setting discrimination need to the strongest possible pressure ($percentile = 0$).

As it requires discriminating extremely similar colors, for which no language could possibly have distinct words or even phrases, the game is very difficult, and indeed none of 20 runs met our criteria for success ($> 95\%$ discrimination accuracy after convergence). However, 6 runs did reach discrimination success above 90%. Figure S2 compares the corresponding systems to the ones emerging in the standard percentile-constrained games we study in the main text. It is clear that, in terms of efficiency/complexity, the systems emerging from the unconstrained games are comparable to those emerging when controlling discriminative need.

We conclude that our results do not depend on controlling discrimination need through the percentile parameter.

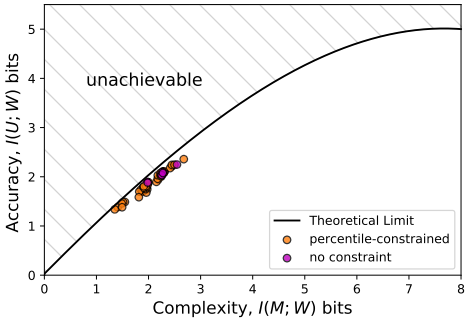


Fig. S2. NN systems emerging from *percentile-constrained* and *no constraint* games plotted on the information plane. The former are those further studied in the main text. The IB curve (black line) defines the theoretical limit on accuracy given complexity.

3. Salience-weighted source distribution

In the main article we assumed a uniform distribution $p(c)$ for chips sampling. To assess the impact of this assumption on the results, we replicate here our main experiment using the salience-weighted (SW) distribution introduced by Gibson et al. (1). This distribution, based on color frequencies in natural images, estimates the probability of a given color c considering the ratio between the frequency with which c appears in objects and its overall frequency (the sum of times c appears both in objects and as background). Gibson et al. (1) originally computed this probability for 80 colors only. To construct a SW prior for the whole WCS palette, we follow ZKRT (2), and use an RBF interpolation with the same parameters. The estimation of this prior is shown in Figure S3.

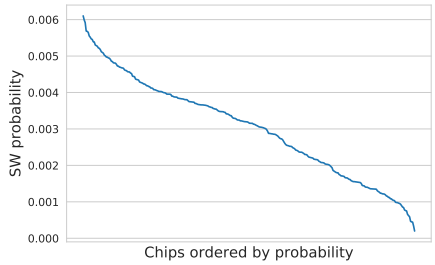
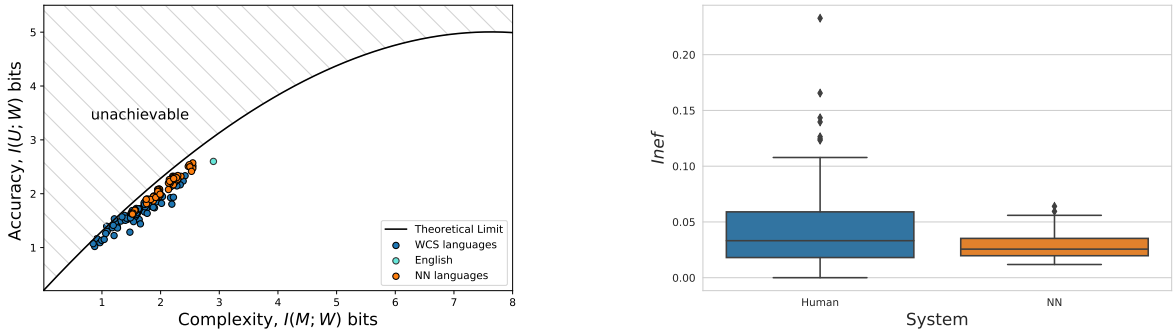


Fig. S3. The estimated salience-weighted (SW) prior over the 330 chips. Chips are ordered by decreasing probability.

We re-run the analysis described in the main paper, but now sampling both targets and distractors according to the SW distribution. Figure S4 confirms that our results do not depend on the uniform assumption made in the main paper. With this alternative skewed input distribution as well (see Figure S3), NN systems are as efficient as the humans ones, and lying just below the same segment of the IB curve.



(a) NN and human naming systems in the information plane. The theoretical limit is defined by the IB curve (black line).

(b) Comparing $Inef$ of emergent and human systems (WCS data and English). A t-test fails to detect a significant difference between the two types.

Fig. S4. IB efficiency of human and NN color-naming systems when considering SW input distribution.

4. Efficiency: comparing human vs. NN systems, and actual vs. rotated systems

Both human and NN systems are efficient. Figure S5 shows that the whole distribution of $Inef$ values of NN systems is well-contained within the range of variation attested in human languages.

ZKRT (2) presented a control study in which they compared each human naming system with a set of 39 hypothetical variants obtained by rotating the system along the hue dimension. They showed that the real systems are more efficient than the control set. In this section, we replicate the analysis with human systems and extend it to NN ones.

First, as shown in Figures S6a and S6b, and similar to ZKRT (2), we find that actual human naming systems are on average closer to the theoretical optimal limit: 98% of human languages attain a better trade-off than their rotated counterparts. Interestingly, this pattern is even stronger in NN naming systems, as they *all* achieve a better trade-off compared to any of their hypothetical variants (cf. Figures S6c and S6d).

5. Encouraging the emergence of a two-word system during training

We observed in the main text that the simplest NN systems include 3 terms. To test whether NNs could in principle develop a simpler system, we designed two variants of the discrimination game where we are positive that 100% performance could be

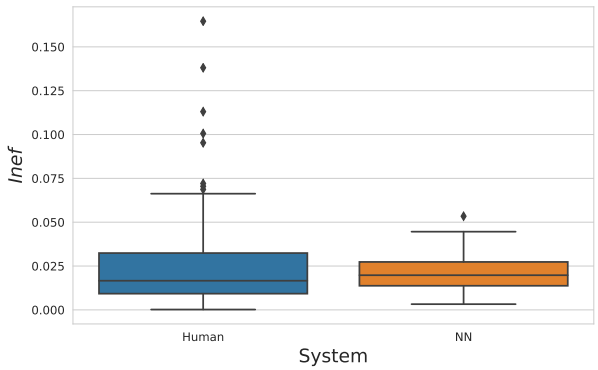
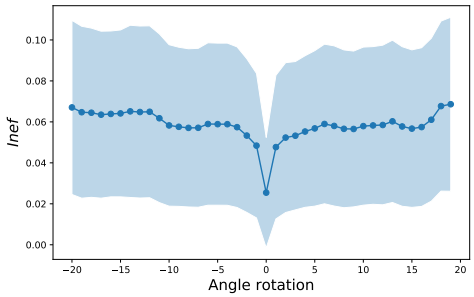
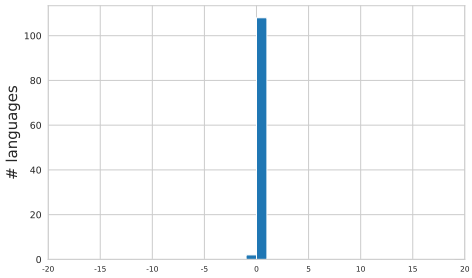


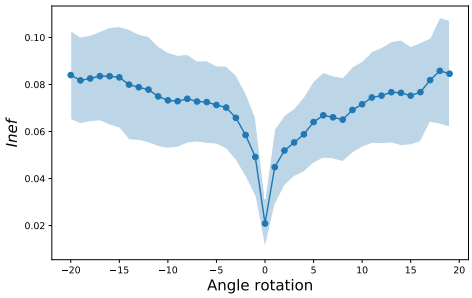
Fig. S5. Comparing $Inef$ of NN and human systems (WCS data and English). A t-test fails to detect a significant difference between the two types.



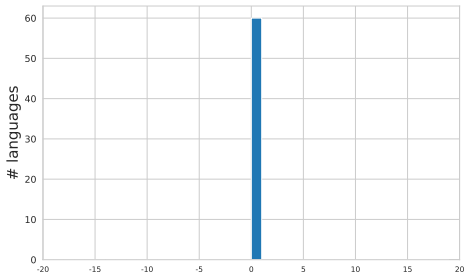
(a) Human systems: $Inef$ of actual WCS naming systems (at rotation$= 0$) vs. their rotated hypothetical variants. The continuous curve represents averages across systems for each rotation degree, and the colored region marks standard deviation across systems.



(b) Human systems: Histogram of distribution of most efficient rotation across WCS data and their hypothetical variants. As can be seen, in nearly all cases the highest-efficiency rotation is the 0-rotation, that is, the actual language



(c) NN systems: $Inef$ of actual NN naming systems (at rotation$= 0$) vs. their rotated hypothetical variants. The continuous curve represents averages across systems for each rotation degree, and the colored region marks standard deviation across systems.



(d) NN systems: Histogram of distribution of most efficient rotation across NN systems and their hypothetical variants. The 0-rotation system (that is, the actual NN system) is the most efficent one in all cases.

Fig. S6. Comparing WCS and NN naming systems with their rotated variants. Rotation 0 corresponds to actual systems.

attained using two color words only.

To construct the first game, we use the FCM clustering algorithm (see main text) to partition the color space into two clusters optimized for minimal intra-cluster distance (FCM is a fuzzy clustering algorithm, but we discretize its outcome to obtain a hard partition). As shown in Figure S7a (left), this leads roughly to a yellow/other distinction. For the second game, we partition the color space into dark and light regions (see Figure S7b (left)). This partition is in line with the basic distinction found in human languages with two color-terms (such as Dani) (3).

In both settings, we ensure that target and distractor always come from the two distinct clusters. To do so, we sample a target color uniformly from the 330 candidate colors. Then, knowing the cluster of the target color, we sample a distractor, also uniformly, from the other cluster. Thus, a system could reach 100% performance by relying on two names denoting the two clusters.

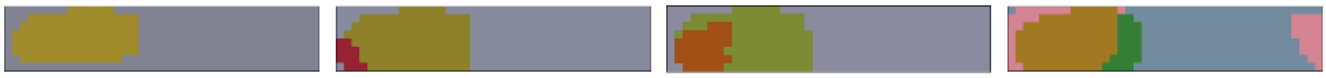
We find that this setup is relatively hard for NNs: only 3/20 runs succeed in each game. Average success rate across the latter runs is at 97%. The corresponding NN systems minimally feature 3 terms, and are more complex than necessary. Concretely, a 2-term system could have complexity 0.87 and 0.99 in the FCM-based and dark/light games, respectively. NNs develop systems with an average (std) complexity of 1.32 (0.22) and 2.13 (0.15), for the respective games. Though NNs use more words than needed, we observe in Figure S7 that their systems stay close to the ground-truth partitions (yellow/other with a supplementary reddish term in Figure S7a, and dark/light, but referring to dark with multiple terms in Figure S7b).

6. Discreteness and success rate

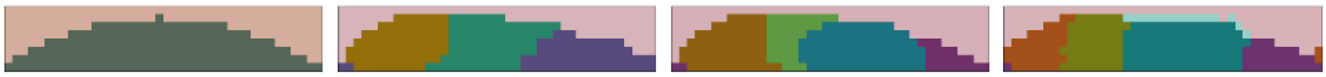
Coherently with the fundamental observation that communicating across a discrete channel is more challenging, which leads to the complexity avoidance effect we discuss in the main text, we also observe that fewer simulation runs are successful when the channel is more discrete.*

Recall that REINFORCE (RF) messages are purely discrete both at training and at testing time, and that, for Gumbel-Softmax (GS), the lower the τ, the closer the approximation to discreteness at training time is. Table S1 shows that success rate is clearly correlated with channel smoothness, with the exception of GS with $\tau = 10$, which is the "smoothest" setting, but also one of the most difficult ones.

*As usual, we consider a run successful if, after convergence, the NNs can correctly communicate about at least 95% of the possible distinct targets.



(a) Color space partition obtained with FCM clustering with 2 clusters (left panels), and the 3 successful NN systems trained on a game where targets and distractors are always sampled from the two distinct clusters (next 3 panels).



(b) Dark/light partition of the color space (left), and the 3 successful NN systems trained on a game where targets and distractors are always sampled from the two distinct regions (next 3 panels).

Fig. S7. Ground-truth partitions of the space used to design the discrimination games (left) and the 3 corresponding successful NN systems in each game. Cluster colors are obtained by averaging the RGB values of all chips in the cluster.

We conjecture that the low success rate of GS with $\tau = 10$ stems from different reasons than failures in the more discrete settings. In particular, we expect that, in the more discrete settings, complexity of the emergent system after training is systematically *lower* in failed runs, because failures stem from the difficulty of establishing a sufficiently complex protocol through the discrete channel. However, this should not be the case for $\tau = 10$, where complexity should be comparable in failed and successful runs. We verify this hypothesis quantitatively in Table S1 by comparing, for each setting, the complexity of failed and successful naming-systems. We observe that, if successful systems have systematically larger complexity, this difference is only significant when communicating with a discrete(-like) channel (RF and GS with $\tau = 1$). For the remaining settings, there is no significant difference in complexity between successful and failed systems. This supports the claim that failure correlates with lower complexity only when communicating through a more discrete channel. The low success rate of GS with $\tau = 10$ may be explained by the noise introduced in that setting when approximating Categorical samples. Indeed, as mentioned in the main paper, larger τ leads to more continuous, thus noisier, estimation of messages.

Setting	τ	success rate	avg. complexity	
			successful systems	failed systems
RF	-	7.8%	1.60	0.75*
GS	1	46.7%	2.13	1.42*
	5	46.7%	3.06	2.90
	10	20%	3.00	2.81

Table S1. Relation between channel smoothness, success rate and complexity for successful and failed systems. * marks significant differences (t-test, $p < 0.001$).

7. Effect of more/less discrete training on Speakers' output distribution

Recall that, following the human data modeling of ZKRT (2), we let Speaker define an output word probability distribution given each input color, $P(w|c)$. Here, we ask whether this distribution becomes flatter as we train it in an increasingly smoother (less dicrete) setup.

Figure S8 shows the average $P(w|c)$ entropy after training in settings ranging from purely discrete REINFORCE (RF) to Gumbel Softmax (GS) with increasing τ (corresponding to more smoothness during training). Here, entropy measures Speaker uncertainty about which term to use for a certain color input. In the more discrete settings, entropy is approximating 0, which would correspond to a categorical distribution (only one word is produced for each input). As smoothness increases, entropy also increases. However, Speaker is still, on average, quite confident about which term to pick, as entropy is still far from its uniform-probability level.

8. Impact of agent capacity

Informal experimentation showed that, provided that Speaker is powerful enough, the color discrimination task is easily solved by simple Listener NNs. As larger Listeners are harder to train, the main experiments use the simplest possible Listener succeeding at the game, that is, a 1-layer NN with 5 hidden units. In contrast, large Speakers were necessary for game success. We experiment with 3-layer NN Speakers of hidden size 1000. Although simpler Speakers are occasionally able to learn the game, our preliminary experiments showed that the chosen combination led to significantly more successful runs.

We now systematically vary these hyperparameters to study the effect of agent capacity, training with Gumbel-Softmax with different temperatures. In particular, we study how agents' hidden sizes influence the complexity of NN systems. We note

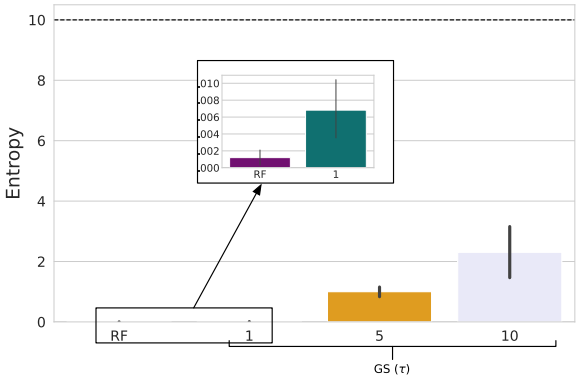


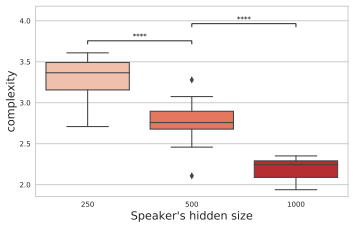
Fig. S8. Average entropy (in bits) of Speakers' outputs for different training regimes. The entropy is first averaged across the 330 outputs corresponding to all input color chips, then across Speakers in the considered training setting. Vertical bars represent standard deviation across Speakers. The horizontal line represents the entropy of a uniform distribution over 1024 categories (equal to $|V|$). The zoom box shows the low-entropy values (corresponding to (near-)discrete communication at training time).

h_a the hidden size of agent a, where $a = s$ if the agent plays the role of Speaker and $a = l$ if the agent is Listener. We vary h_s in $\{250, 500, 1000\}$ and h_l in $\{5, 10, 100, 500\}$. Across these experiments, we keep the same number of layers used in the main analysis. However, we observe the same pattern when varying the number of layers (not reported here). Finally, we only include successful runs in the analysis (discrimination accuracy strictly above 95%).

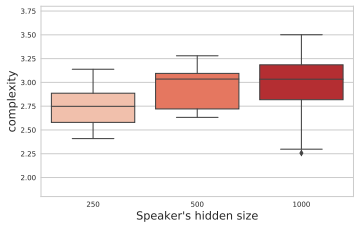
We start with the default Gumbel-Softmax temperature value $\tau = 1$, corresponding to an essentially discrete setup. Each experiment is repeated 20 times. Figure S9c shows that varying Listener's parameters does not impact the complexity of the emergent systems. On the other hand, in Figure S9a we observe that, with the same training regime, increasing h_s makes the systems significantly less complex ($p < 10^{-7}$, Kruskal-Wallis test). In short, when considering emergent system complexity, agent capacity only matters when it concerns the Speaker agent, that is, when the extra capacity occurs *before* the *discrete* communication channel bottleneck.

If h_s matters when communicating with a discrete channel, we look now at its effect when training with a smoother one, using $\tau = 10$. As noted in Supplementary 6, this large τ leads to fewer successful runs. We thus repeat this experiment 60 times. When varying Listener's capacity, agents were successful in the game only for $h_l \in \{5, 10\}$. For these successful settings, the average (standard deviation) complexities are 2.97 (0.31) and 2.87 (0.29) for h_l equal to 5 and 10 respectively, indicating no significant difference between naming-systems' complexities in this setting. The same observation holds when considering different Speaker's capacity with $h_s \in \{250, 500, 1000\}$. As shown in Figure S9b, in this more continuous setup there is no significant difference across Speakers with different capacities ($p = 0.07$, Kruskal-Wallis test).

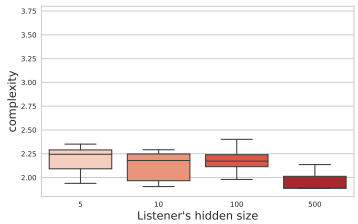
A natural interpretation of these results is that, when the channel is virtually discrete (low τ), the complexity minimization pressure is so high that Speaker tries to compress the input as much as it can, passing only the information that is strictly necessary for communication success. The more capacity Speaker has, the better it succeeds at compactly encapsulating useful information about its inputs, resulting in *simpler* systems for *larger* Speakers. However, this effect disappears for the more continuous setup with high τ, as the minimization pressure due to discreteness is no longer at play.



(a) Effect of Speaker's hidden size when τ=1: Systems' complexity with respect to Speaker hidden size. Results show a significant difference between the different hidden sizes with $p < 10^{-7}$ when using the Kruskal-Wallis test.



(b) Effect of Speaker's hidden size when τ=10: Systems' complexity with respect to Speaker hidden size. Results show *no* significant difference between the different hidden sizes with $p = 0.07$ when using the Kruskal-Wallis test.



(c) Effect of Listener's hidden size when τ=1: Systems' complexity with respect to Listener hidden size. Results show *no* significant difference between the different hidden sizes with $p = 0.10$ when using the Kruskal-Wallis test.

Fig. S9. Systems' complexity for different agents' hidden sizes and different training regimes. Pairwise differences evaluated with Bonferroni-corrected Mann-Whitney-Wilcoxon (****p<0.0001); 'ns' differences are not labeled.

Kharitonov and colleagues (4) present further evidence of the interaction between discreteness and Speaker/Listener capacity in the context of a toy experiment. They show that, with a more continuous channel, both a larger Speaker and a larger

Listener can memorize a data-set with random input-label associations. With a discrete channel, however, an extra-capacity Speaker will memorize the labels and only transmit a compressed summary of the needed information. Increasing Listener's capacity, on the other hand, has no effect, as the agents are simply incapable to learn to transmit the high-complexity "raw data" for the Listener to process through the discrete channel.

9. How are color naming systems (in)efficient?

We saw in the main text that systems with high discriminative need tend to be more complex, but not inefficient, whereas, when we let the channel be more continuous, we see the emergence of systems that are more complex *and* inefficient. In this section, we explore one concrete way in which color-naming systems might be (in)efficient (without claiming that it is the only one). To this end, we introduce a measure that quantifies, for a word in a given system, the degree to which its denotation is also covered by another word, or how *separate* the meaning of a word is from that of the others. We refer to this measure as *sep* (for *separation*). Intuitively, in efficient naming systems all words should have very high *sep*. Lower-*sep* words are redundant leading to inefficient partition of the color space. Indeed, their presence increases the system's complexity ($I(M;W)$) with no notable increase in accuracy ($\propto KL[M||\hat{M}]$ with KL the Kullback–Leibler divergence).

A. Estimating sep. We aim to quantify how redundant/separate each word is. For example, in English, the word "scarlet" is in a sense redundant, as its meaning is included in that of "red". Using both words makes the color-naming system less efficient. Indeed, "red" is a fine word to refer to scarlet tonalities, and adding "scarlet" only slightly increases communication accuracy at the cost of an increase in complexity ("scarlet" might still be useful as a specialized word, of course). Formally, to measure if a word w is redundant, we need to find a w' that covers the same reference. To do so, we define $C_w = \{c, \text{s.t. } c \text{ denoted by } w\}$ the set of colors/references denoted by the word w. Moreover, for two given words w and w', $p(C_w \not\subset C'_w)$, is the probability that the denotation of w is separate from that of w', such that:

$$\begin{aligned} p(C_w \not\subset C'_w) &= p(w) \times p(\overline{w'}|w) \\ &= p(w) \times (1 - p(w'|w)) \\ &= p(w) - \sum_c p(w, w'|c) \times p(c) \end{aligned} \quad [1]$$

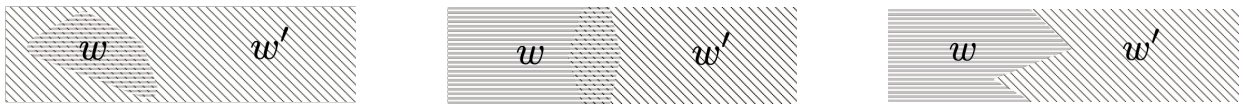
Since, by construction, for any given color c, we sample words independently from Speaker, we have:

$$p(C_w \not\subset C'_w) = p(w) - \sum_c p(w|c) \times p(w'|c) \times p(c) \quad [2]$$

Finally, we search, among all possible words distinct from w, the word from which w is *least* likely to be separated (in terms of denotation). To this end, we define:

$$sep(w) = min_{w' \neq w} p(C_w \not\subset C'_w) \quad [3]$$

Low $sep(w)$ indicates that w is redundant, that is, it exists a w' that is likely to cover w's denotation. Figure S10 presents 3 scenarios with different *sep* values.[†]



(a) No separation: $sep(w)$=0 **(b)** Partial separation: 0<$sep(w)$<1 **(c)** Perfect separation: $sep(w)$=1

Fig. S10. Three hypothetical two-word systems. The regions represent the extension of items (color chips) denoted by each word. Each system partitions the extension differently leading to different $sep(w)$ values.

B. Compared systems. We measure *sep* of the words in emergent NN systems when varying both discriminative need and channel smoothness. Based on the main paper results, we expect that, while high discriminative need complexifies the system, it should not impact the nature of its words, while a smooth channel might lead to the emergence of redundant words. In practice, we compare 3 different settings:

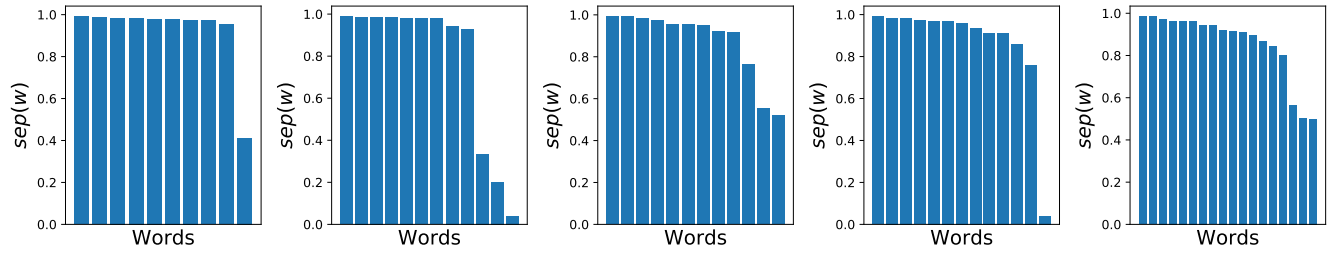
- `smooth`: composed of 5 different successful NN systems obtained with an intermediate discriminative need (*percentile*=50) and a largely smooth channel (τ=10).
- `need`: composed of 5 different successful NN systems arising when agents have an extreme discriminative need (*percentile*=20), and a discrete-like channel (τ=1).

[†] We verified that sep does not depend on the number of used words. In particular, FCM partitions till $K = 20$ comprise only separate words with $sep(w) \approx 1$, confirming the validity of our measure.

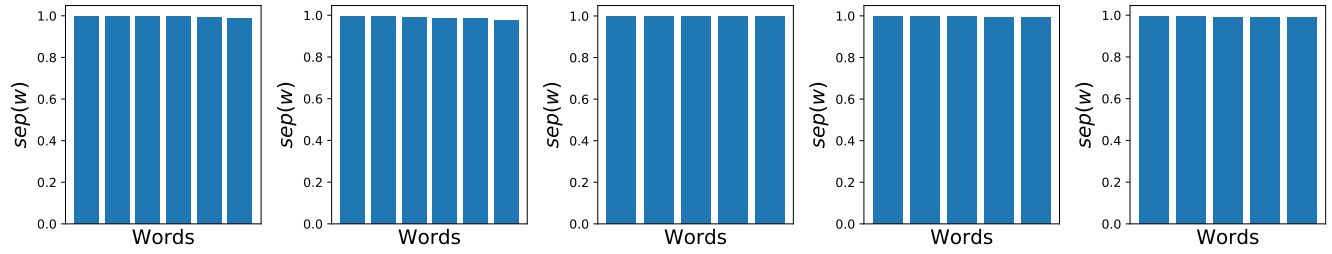
- `control`: a control setup containing 5 successful NN systems obtained with *percentile*=50 (similar to `smooth`) and τ=1 (similar to `need`).

Note that all studied systems allow successful communication ($> 95\%$ success rate in the discrimination game).

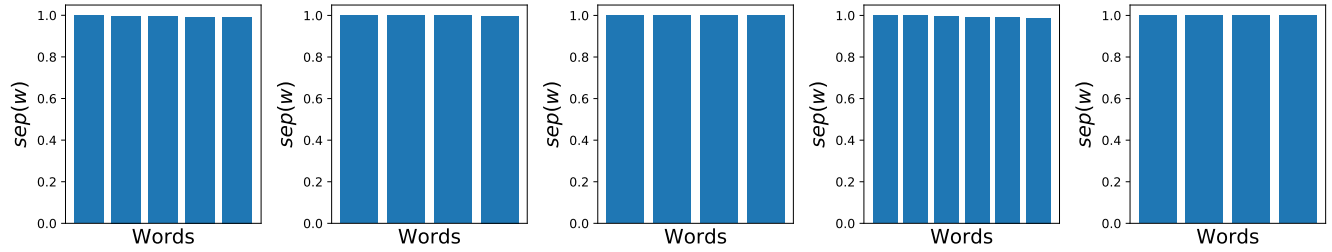
C. Results. Both the `need` and `control` settings avoid redundant words. In fact, Figures S11c and S11b show that, for any considered system, $sep(w) \approx 1$ for all w. On the other hand, `smooth` displays a different trend. We observe in Figure S11a that all systems in `smooth` (i.e., trained with high $\tau = 10$) include some words with low *sep*, confirming that a smooth channel leads to the emergence of redundant words, at the cost of efficiency.



(a) The 5 systems of the *smooth* setting



(b) The 5 systems of the *need* setting



(c) The 5 systems of the *control* setting

Fig. S11. sep across different systems. The histograms denote *all* unique words of a system, sorted by decreasing sep. Each row represents one setting, and each sub-figure one system (out of 5) in the corresponding setting.

Our results emphasize the difference between the effect of complexification due to higher discriminative need vs. smoothness. A high discriminative need will complexify the naming system by introducing separate words. However, varying channel smoothness changes both system complexity and how redundant/separate its words are. That is, with a discrete channel (`need` and `control` settings), we observe the emergence of systems containing only separate words, whereas when agents communicate through a smooth channel (`smooth` setting), NN systems start to develop several redundant words.

Finally, *sep* can be related to the notion of *basic terms* introduced by Berlin and Kay (5). Berlin and Kay state that one condition for a term/word to be basic is to have *"its signification [...] not included in that of any other color term"*. Words meeting this condition will have high *sep*. Hence, we can relate the idea of a basic term to the notion of efficiency, in that systems where most words meet this condition will lead to an efficient color-naming system.

10. Direct comparison of color space partitions

The distribution of emergent NN naming systems along the accuracy and complexity axes is strikingly similar to that of the WCS natural languages. The relation between the specific way in which NN systems cluster the color space and the partitions created by human languages is more nuanced. NN systems share with human systems the fundamental property of partitioning

the color space into convex regions, but they do not rely on the dark/light dimension as the core axis along which to partition colors. Also, they appear to stay closer to a purely perception-based partition of color space than human languages do, which actually makes them *more* convex than human languages.

To quantify the similarity between NN and human color partitioning, we frame it as a clustering problem. The sets of colors denoted by the same name in a NN naming system are treated as clusters, and compared against ground-truth partitions provided by the WCS languages (we also discuss an experiment in which FCM clustering solutions are used as gold standards to compare both NN and natural languages against). We adopt the standard F_1 clustering quality evaluation measure (6, Ch. 16). F_1 takes its highest possible value of 1 when a clustering solution (determined, in our case, by how colors are grouped by name) is identical to the ground truth (in terms of how it partitions the color space). We discretize the NN naming distributions by labeling each color with the name maximizing $P(w|c)$. For human systems, we use majority names across subjects as color labels.

As there is a lot of variation in human language, no NN system could be similar to all WCS systems. Thus, for each NN system, we pick the largest F_1 score it attains when compared to all human systems of the same cardinality (that is, with the same number of color names after discretization). We refer to this score as *best* F_1. Best F_1 thus measures the degree to which a NN naming system is similar to at least one human system.

Comparison \ *Cardinality*	3	4	5	6	7
NN vs. WCS	0.559 (0.039)	0.472 (0.036)	0.521 (0.051)	0.547 (0.030)	0.531 (NA)
WCS vs. WCS	0.869 (0.031)	0.717 (0.036)	0.685 (0.060)	0.765 (0.081)	0.719 (0.090)
informed baseline	0.412 (0.023)	0.319 (0.044)	0.280 (0.034)	0.262 (0.053)	0.225 (0.044)
NN vs. FCM	0.675 (0.049)	0.610 (0.102)	0.604 (0.052)	0.611 (0.025)	0.602 (NA)
WCS vs. FCM	0.466 (0.022)	0.510 (0.056)	0.481 (0.043)	0.464 (0.063)	0.485 (0.064)

Table S2. Average best F_1 by cardinality (standard deviation in parenthesis; the latter is NA when there is only one tested naming system of the corresponding cardinality). *NN vs. WCS*: averages across NN naming systems compared to WCS languages as ground truth. *WCS vs. WCS*: averages across WCS languages, using nearest WCS language as ground truth. *Informed baseline*: for each WCS language, generate 100 pseudo-naming-systems by shuffling its names; pick best F_1 with original naming scheme across pseudo-naming systems; average best F_1s across languages with same cardinality. *NN vs. FCM*: averages across NN naming systems when using discretized FCM clustering solutions as ground truth. *WCS vs. FCM*: averages across natural language naming systems when using FCM solutions as ground truth. Since there is only one FCM solution per cardinality, in the last two comparisons average best F_1 equals average F_1.

The first row of Table S2 reports averaged best F_1 for the NN systems when using the WCS names as ground-truth labels.[‡] To make sense of these numbers, we compare them to an upper bound and a baseline in the next two rows. The upper bound is given by averaging the same score across WCS languages, when using the nearest language to each as ground truth. NN naming schemes are clearly farther away from those of the nearest natural languages than natural languages are from each other. The baseline is obtained by generating, for each WCS language, 100 pseudo-naming-systems with the same label frequencies. The F_1 score with respect to the reference WCS language is computed for each of the 100 pseudo-naming-systems, and the best one is retained for each WCS language. This is an *informed* baseline because it has access to the ground-truth label distribution. Across all cardinalities, NN systems are much closer to actual WCS languages than the informed baseline is.

Where does the difference between natural and NN naming schemes come from? A partial answer is provided by the next two rows of Table S2, where we evaluate to what degree natural and NN systems match the partitions obtained through fuzzy c-means (FCM) clustering (7) in CIELAB space. The latter should approximate color space partitions that are optimal on purely perceptual grounds (FCM returns partitions that minimize within-cluster distance in color space). We compare each NN/human naming system to the (discretized) FCM solution with K equal to the naming system cardinality.

The *NN vs. FCM* comparison reveals that NN systems are closer to the purely perception-based FCM partitioning than to natural languages (*NN vs. WCS* in the first row of the table). Natural languages in turn, as shown in the *WCS vs. FCM* row, are further away from FCM solutions than NN naming systems are. So, the difference between WCS and NN systems stems, at least in part, from the fact that human naming systems drifted further apart from purely perceptual pressures than NN systems did. This makes sense: human language evolution is subject to many external pressures (social, environmental, etc.) that are not part of our language emergence simulations.

Figure S12 provides a more qualitative insight into how NN and natural language color partitions differ, by visualizing emergent naming systems of cardinality 3 and 5 together with their nearest WCS languages. To avoid cherry-picking, we chose, for each cardinality, the naming systems with median best F_1. The results are generally representative, although for higher cardinalities a qualitative comparison becomes problematic due to considerable noise in the WCS data.

Wobé, the reference 3-color-term human system (top right of Figure S12), illustrates the near-universal 3-way split into "light", "dark" and red (5). The corresponding NN system (top left) does not encode the dark/light split. While unnatural in this respect, the partitioning is, like those found in human languages, clearly convex (8, 9). The NN system clusters correspond, moreover, to the other basic colors attested in low-complexity human languages, once we exclude the dark/light distinction: red, green and a yellow/brown patch.

[‡] We include all successful NN systems also used in the main experiments.

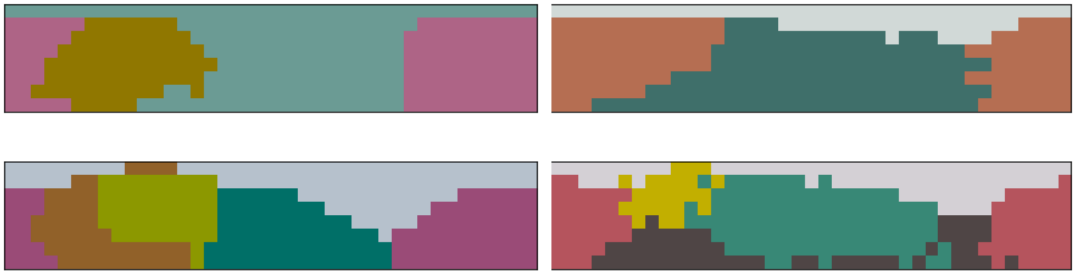


Fig. S12. Top: a 3-word NN naming system (left) compared to Wobé (Niger-Congo) (right), its closest WCS counterpart. Bottom: a 5-word NN system (left) compared to Bauzi (East Geelvink Bay) (right), its closest WCS counterpart. Same-color clusters represent colors denoted by the same word (after discretizing). Cluster colors are obtained by averaging the RGB values of all colors in the cluster. Refer to Figure 1 in the main article for a separate rendering of each chip, shown in the same arrangement.

Bauzi (bottom right of Figure S12) follows a typical 5-term naming scheme: white (light), black (dark), red, green and yellow. The corresponding NN system (bottom left) omits the black category, but does not radically depart otherwise from the human system, with the colors clustered into white, red, green and yellow areas. Again, the NN partition is clearly convex.

To quantitatively substantiate the qualitative claim about NN systems' convexity, we computed the *degree of convexity* of each NN system (and, for comparison, WCS and FCM systems) using the method recently proposed by Steinert-Threkeld and Szymanik (10, p. 5). For each (discretized) color name, we compute the ratio of number of points denoted by the name to the number of points in their convex hull, weighting by partition size and normalizing. Averaged results are given in Table S3.

Naming System \ *Cardinality*	3	4	5	6	7
NN	0.999 (0.003)	0.998 (0.004)	0.999 (0.002)	1.00 (0.000)	0.997 (NA)
WCS	0.964 (0.030)	0.936 (0.026)	0.935 (0.039)	0.949 (0.047)	0.926 (0.038)
FCM	0.991 (NA)	1.000 (NA)	1.000 (NA)	1.000 (NA)	1.000 (NA)

Table S3. Average degree of convexity by cardinality for different naming systems (standard deviation in parenthesis; the latter is NA when there is only one tested naming system of the corresponding cardinality).

The degree of convexity of NN systems is extremely high, and approaching that of FCM clustering (which naturally favors convexity because of its distance-minimizing objective). Remarkably, the degree of convexity of NN systems is *higher* than that of the natural languages in WCS. This might be due, again, to the fact that humans must optimize communicative constraints that are not entirely perception-driven, or, more simply, to the noise inherent in the WCS surveying methodology. We leave this intriguing question to further work.[§]

In sum, NN color-naming systems, like (and perhaps more than) human ones, show a clear tendency to partition the color space into convex regions. However, the latter regions depart to some extent from those typically defined by human color naming. NN systems might stay closer to a purely perceptual partitioning of the color space. Moreover, qualitatively, they do not seem to enforce the distinction between white (light) and black (dark), which is instead universally present in human languages. Note however that, as we report in Supplementary 5, NNs are in principle able to discover the dark/light distinction if we encourage it in the design of the game.

References

1. E Gibson, et al., Color naming across languages reflects color use. *Proc. Natl. Acad. Sci.* **114**, 10785–10790 (2017).
2. N Zaslavsky, C Kemp, T Regier, N Tishby, Efficient compression in color naming and its evolution. *Proc. Natl. Acad. Sci.* **115**, 7937–7942 (2018).
3. P Kay, CK McDaniel, The linguistic significance of the meanings of basic color terms. *Language*, 610–646 (1978).
4. E Kharitonov, R Chaabouni, D Bouchacourt, M Baroni, Entropy minimization in emergent languages in *Proceedings of ICML*. (virtual conference), pp. 2718–2728 (2020).
5. B Berlin, P Kay, *Basic color terms: Their universality and evolution*. (Univ of California Press), (1991).
6. C Manning, P Raghavan, H Schütze, *Introduction to Information Retrieval*. (Cambridge University Press, Cambridge, UK), (2008).
7. J Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. (Kluwer, Boston, MA), (1981).
8. P Gärdenfors, *Conceptual Spaces*. (MIT Press, Cambridge, MA), (2000).
9. G Jäger, Natural color categories are convex sets in *Logic, Language and Meaning*, eds. M Aloni, H Bastiaanse, T de Jager, K Schulz. (Springer, Berlin, Germany), pp. 11–20 (2010).
10. S Steinert-Threlkeld, J Szymanik, Ease of learning explains semantic universals. *Cognition* **195**, 104076 (2020).

[§] Could the convexity of NN systems be an artefact of how we pick distractors based on a discrimination need threshold, thus indirectly favoring systems where nearer color chips get the same name (as they are likely to be below this threshold)? This is not the case, as shown by the fact that all 6 successful systems discussed in Supplementary 2, where there is *no* control on target-distractor distance, are also fully convex (degree of convexity uniformly at 100%).

Chapter 5

Compositionality

Language varies across learners at every level of description (sound, lexicon, grammar). More intriguingly, language varies even within the span of a human life; new expressions, new words, are continuously introduced in the language. Such mutability has spurred the establishment of institutions devoted to the standardization and stabilization of languages (like the *French Academy*). Despite this continual variation, all children, at any time and place, can gain a good understanding of their native language. This observation spurred a lot of interest in language learning and evolution. One finding that arises from this interest is that human languages, despite their apparent differences, do share important regularities [51, 41, 39]. All languages are complex symbolic systems that combine the same units (phonemes, morphemes, words, sentences) to convey infinite meanings. This ability to combine already acquired units to express meanings is defined as *compositionality*. Such property is desirable for AI as it enables the generalization to new meanings [65].

In this chapter, we study if NNs can generalize to new objects through a compositional communication protocol. To this end, we introduce different measures to quantify intuitive forms of compositionality. We show that, in the presence of a rich and varied environment, NNs develop productive (i.e., referring to never-seen objects), though non-*intuitively*-compositional, languages. Our follow-up experiments suggest that, if compositionality emerges (possibly by chance), it will make a language easier to transmit. In other words, we propose a new view of the universality

of compositionality; If the latter emerges by chance, it will survive and spread fast across generations. From an AI perspective, our work suggests that, in order to build compositional representation, the latter should maximize ease of transmission across different learners.

Compositionality and Generalization in Emergent Languages

Rahma Chaabouni[1,2]*, Eugene Kharitonov[1]*, Diane Bouchacourt[1], Emmanuel Dupoux[1,2], and Marco Baroni[1,3]

[1]Facebook AI Research
[2]Cognitive Machine Learning (ENS - EHESS - PSL Research University - CNRS - INRIA)
[3]ICREA
{rchaabouni,kharitonov,dianeb,dpx,mbaroni}@fb.com

Abstract

Natural language allows us to refer to novel composite concepts by combining expressions denoting their parts according to systematic rules, a property known as *compositionality*. In this paper, we study whether the language emerging in deep multi-agent simulations possesses a similar ability to refer to novel primitive combinations, and whether it accomplishes this feat by strategies akin to human-language compositionality. Equipped with new ways to measure compositionality in emergent languages inspired by disentanglement in representation learning, we establish three main results. First, given sufficiently large input spaces, the emergent language will naturally develop the ability to refer to novel composite concepts. Second, there is no correlation between the degree of compositionality of an emergent language and its ability to generalize. Third, while compositionality is not necessary for generalization, it provides an advantage in terms of language transmission: The more compositional a language is, the more easily it will be picked up by new learners, even when the latter differ in architecture from the original agents. We conclude that compositionality does not arise from simple generalization pressure, but if an emergent language does chance upon it, it will be more likely to survive and thrive.

1 Introduction

Most concepts we need to express are composite in some way. Language gives us the prodigious ability to assemble messages referring to novel composite concepts by systematically combining expressions denoting their parts. As interest raises in developing deep neural agents evolving a communication code to better accomplish cooperative tasks, the question arises of how the emergent code can be endowed with the same desirable *compositionality* property (Kottur et al., 2017; Lazaridou et al., 2018; Mordatch and Abbeel, 2018; Cogswell et al., 2019; Li and Bowling, 2019). This in turn requires measures of how compositional an emergent language is (Andreas, 2019). Compositionality is a core notion in linguistics (Partee, 2004), but linguists' definitions assume full knowledge of primitive expressions and their combination rules, which we lack when analyzing emergent languages (Nefdt, 2020). Also, these definitions are categorical, whereas to compare emergent languages we need to quantify degrees of compositionality.

Some researchers equate compositionality with the ability to correctly refer to unseen composite inputs (e.g., Kottur et al., 2017; Cogswell et al., 2019). This approach measures the generalization ability of a language, but it does not provide any insights on *how* this ability comes about. Indeed, one of our main results below is that emergent languages can attain perfect generalization without abiding to intuitive notions of compositionality.

Topographic similarity has become the standard way to quantify the compositionality of emergent languages (e.g., Brighton and Kirby, 2006; Lazaridou et al., 2018; Li and Bowling, 2019). This metric measures whether the distance between two meanings correlates with the distance between the messages expressing them. While more informative than generalization, topographic similarity is still rather agnostic about the nature of composition. For example, when using, as is standard practice, Levenshtein distance to measure message distance, an emergent language transparently concatenating symbols in a fixed order and one mixing deletion and insertion operations on free-ordered symbols can have the same topographic similarity.

We introduce here two more "opinionated" measures of compositionality that capture some intuitive properties of what we would expect to hap-

*Contributed equally.

pen in a compositional emergent language. One possibility we consider is that order-independent juxtapositions of primitive forms could denote the corresponding union of meanings, as in English noun conjunctions: *cats and dogs*, *dogs and cats*. The second still relies on juxtaposition, but exploits order to denote different classes of meanings, as in English adjective-noun phrases: *red triangle*, *blue square*. Both strategies result in *disentangled* messages, where each primitive symbol (or symbol+position pair) univocally refers to a distinct primitive meaning independently of context. We consequently take inspiration from work on disentanglement in representation learning (Suter et al., 2019) to craft measures that quantify whether an emergent language follows one of the proposed composition strategies.

Equipped with these metrics, we proceed to ask the following questions. First, are neural agents able to generalize to unseen input combinations in a simple communication game? We find that generalizing languages reliably emerge when the input domain is sufficiently large. This somewhat expected result is important nevertheless, as failure-to-generalize claims in the recent literature are often based on very small input spaces. Second, we unveil a complex interplay between compositionality and generalization. On the one hand, there is no correlation between our compositionality metrics and the ability to generalize, as emergent languages successfully refer to novel composite concepts in inscrutablly entangled ways. (Order-dependent) compositionality, however, if not necessary, turns out to be a sufficient condition for generalization. Finally, more compositional languages are easier to learn for new agents, including agents that are architecturally different from the ones that evolved the language. This suggests that, while composition might not be a "natural" outcome of the need to generalize, it is a highly desirable one, as compositional languages will more easily be adopted by a large community of different agents. We return to the implications of our findings in the discussion.

2 Setup

2.1 The game

We designed a variant of Lewis' signaling game (Lewis, 1969). The game proceeds as follows:

1. Sender network receives one input $\boldsymbol{i}$ and chooses a sequence of symbols from its vocabulary $V = \{s_1, s_2..., s_{c_{voc}}\}$ of size c_{voc} to construct a message m of fixed length c_{len}.
2. Receiver network consumes m and outputs $\hat{\boldsymbol{i}}$.
3. Agents are successful if $\hat{\boldsymbol{i}} = \boldsymbol{i}$, that is, Receiver reconstructs Sender's input.

Each input $\boldsymbol{i}$ of the reconstruction game is comprised of i_{att} attributes, each with i_{val} possible values. We let i_{att} range from 2 to 4 and i_{val} from 4 to 100. We represent each attribute as a i_{val} one-hot vector. An input $\boldsymbol{i}$ is given by the concatenation of its attributes. For a given (i_{att}, i_{val}), the number of input samples $|I| = i_{val}^{i_{att}}$.

This environment, which can be seen as an extension of that of Kottur et al. (2017), is one of the simplest possible settings to study the emergence of reference to composite concepts (here, combinations of multiple attributes). Attributes can be seen as describing object properties such as color and shape, with their values specifying those properties for particular objects (*red*, *round*). Alternatively, they could be seen as slots in an abstract semantic tree (e.g., agent and action), with the values specifying their fillers (e.g., *dog*, *barking*). In the name of maximally simplifying the setup and easing interpretability, unlike Kottur et al. (2017), we consider a single-step game. We moreover focus on input reconstruction instead of discrimination of a target input among distractors as the latter option adds furtherx complications: for example, languages in that setup have been shown to be sensitive to the number and distribution of the distractors (Lazaridou et al., 2018).

For a fixed $|I|$, we endow Sender with large enough channel capacity $|C| = c_{voc}^{c_{len}}$ ($c_{voc} \in \{5, 10, 50, 100\}$ and $c_{len} \in \{3, 4, 6, 8\}$) to express the whole input space (i.e., $|C| \geq |I|$). Unless explicitly mentioned, we run 10 different initializations per setting. See Appendix 8.1 for details about the range of tested settings. The game is implemented in EGG (Kharitonov et al., 2019).[1]

2.2 Agent architecture

Both agents are implemented as single-layer GRU cells (Cho et al., 2014) with hidden states of size 500.[2] Sender encodes $\boldsymbol{i}$ in a message m of fixed

[1] Code can be found at https://github.com/facebookresearch/EGG/tree/master/egg/zoo/compo_vs_generalization.

[2] Experiments with GRUs of different capacity are reported in the Appendix. We also informally replicated our main

length c_{len} as follows. First, a linear layer maps the input vector into the initial hidden state of Sender. Next, the message is generated symbol-by-symbol by sampling from a Categorical distribution over the vocabulary c_{voc}, parameterized by a linear mapping from Sender's hidden state. The generated symbols are fed back to the cell. At test time, instead of sampling, symbols are selected greedily.

Receiver consumes the entire message m. Further, we pass its hidden state through a linear layer and consider the resulting vector as a concatenation of i_{att} probability vectors over i_{val} values each. As a loss, we use the average cross-entropy between these distributions and Sender's input.

2.3 Optimization

Popular approaches for training with discrete communication include Gumbel-Softmax (Maddison et al., 2016; Jang et al., 2016), REINFORCE (Williams, 1992), and a hybrid in which the Receiver gradients are calculated via backpropagation and those of Sender via REINFORCE (Schulman et al., 2015). We use the latter, as recent work (e.g., Chaabouni et al., 2019) found it to converge more robustly. We apply standard tricks to improve convergence: (a) running mean baseline to reduce the variance of the gradient estimates (Williams, 1992), and (b) a term in the loss that favors higher entropy of Sender's output, thus promoting exploration. The obtained gradients are passed to the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001.

3 Measurements

3.1 Compositionality

Topographic similarity (topsim) (Brighton and Kirby, 2006) is commonly used in language emergence studies as a quantitative proxy for compositionality (e.g., Lazaridou et al., 2018; Li and Bowling, 2019). Given a distance function in the input space (in our case, attribute value overlap, as attributes are unordered, and values categorical) and a distance function in message space (in our case, following standard practice, minimum edit distance between messages), *topsim* is the (Spearman) correlation between pairwise input distances and the corresponding message distances. The measure can detect a tendency for messages with similar meanings to be similar in form, but it is relatively agnostic about the type of similarity (as long as it is captured by minimum edit distance).

We complement *topsim* with two measures that probe for more specific types of compositionality, that we believe capture what deep-agent emergent-language researchers seek for, when interested in compositional languages. In most scenarios currently considered in this line of research, the composite inputs agents must refer to are sets or sequences of primitive elements: for example, the values of a set of attributes, as in our experiment. In this restricted setup, a compositional language is a language where symbols independently referring to primitive input elements can be juxtaposed to jointly refer to the input ensembles. Consider a language with a symbol *r* referring to input element *color:red* and another symbol *l* referring to *weight:light*, where *r* and *l* can be juxtaposed (possibly, in accordance with the syntactic rules of the language) to refer to the input set {*color:red*, *weight:light*}. This language is intuitively compositional. On the other hand, a language where both *r* and *l* refer to these two input elements, but only when used together, whereas other symbol combinations would refer to *color:red* and *weight:light* in other contexts, is intuitively not compositional. Natural languages support forms of compositionality beyond the simple juxtaposition of context-independent symbols to denote ensembles of input elements we are considering here (e.g., constructions that denote the application of functions to arguments). However, we believe that the proposed intuition is adequate for the current state of affairs in language emergence research.

The view of compositionality we just sketched is closely related to the idea of disentanglement in representation learning. Disentangled representations are expected to enable a consequent model to generalize on new domains and tasks (Bengio et al., 2013). Even if this claim has been challenged (Bozkurt et al., 2019; Locatello et al., 2019), several interesting metrics have been proposed to quantify disentanglement, as reviewed in Suter et al. (2019). We build in particular upon the *Information Gap* disentanglement measure of Chen et al. (2018), evaluating how well representations capture independence in the input sets.

Our **positional disentanglement (posdis)** metric measures whether symbols *in specific positions* tend to univocally refer to the values of a specific attribute. This order-dependent strategy is com-

results with LSTMs, that were slower to converge. We were unable to adapt Transformers to successfully play our game.

monly encountered in natural language structures (and it is a pre-condition for sophisticated syntactic structures to emerge). Consider English adjective-noun phrases with a fully intersective interpretation, such as *yellow triangle*. Here, the words in the first slot will refer to adjectival meanings, those in the second to nominal meanings. In our simple environment, it might be the case that the first symbol is used to discriminate among values of an attribute, and the second to discriminate among values of another attribute. Let's denote s_j the j-th symbol of a message and a_1^j the attribute that has the highest mutual information with s_j: $a_1^j = \arg\max_a \mathcal{I}(s_j; a)$. In turn, a_2^j is the second highest informative attribute, $a_2^j = \arg\max_{a \neq a_1^j} \mathcal{I}(s_j; a)$. Denoting $\mathcal{H}(s_j)$ the entropy of j-th position (used as a normalizing term), we define *posdis* as:

$$posdis = 1/c_{len} \sum_{j=1}^{c_{len}} \frac{\mathcal{I}(s_j; a_1^j) - \mathcal{I}(s_j; a_2^j)}{\mathcal{H}(s_j)} \quad (1)$$

We ignore positions with zero entropy. Eq. 1 captures the intuition that, for a language to be compositional given our inputs, each position of the message should only be informative about a single attribute. However, unlike the related measure proposed by Resnick et al. (2019), it does not require knowing which set of positions encodes a particular attribute, which makes it computationally simpler (only linear in c_{len}).

Posdis assumes that a language uses positional information to disambiguate symbols. However, we can easily imagine a language where symbols univocally refer to distinct input elements independently of where they occur, making order irrelevant.[3] Hence, we also introduce **bag-of-symbols disentanglement (bosdis)**. The latter maintains the requirement for symbols to univocally refer to distinct meanings, but captures the intuition of a permutation-invariant language, where only symbol counts are informative. Denoting by n_j a counter of the j-th symbol in a message, *bosdis* is given by:

$$bosdis = 1/c_{voc} \sum_{j=1}^{c_{voc}} \frac{\mathcal{I}(n_j; a_1^j) - \mathcal{I}(n_j; a_2^j)}{\mathcal{H}(n_j)} \quad (2)$$

In all experiments, the proposed measures *topsim*, *posdis* and *bosdis* are calculated on the train set.

[3] This is not unlike what happens in order-insensitive constructions such as English conjunctions: *dogs and cats*, *cats and dogs*.

In Appendix 8.2, we illustrate how the three metrics behave differently on three miniature languages. Across the languages of all converging runs in our simulations, their Spearman correlations are: *topsim*/*posdis*: 0.08; *topsim*/*bosdis*: 0.38; *posdis*/*bosdis*: 0.31. These correlations, while not extremely high, are statistically significant ($p < 0.01$), which is reassuring as all metrics attempt to capture compositionality. It is also in line with reasonable expectations that the most "opinionated" *posdis* measure is the one that behaves most differently from *topsim*.

3.2 Generalization

In our setup, generalization can be straightforwardly measured by splitting all possible distinct inputs so that the test set only contains inputs with attribute combinations that were not observed at training. Generalization is then simply quantified by test accuracy. In intuitive terms, at training time the agents are exposed to *blue triangles* and *red circles*, but *blue circles* only appear at test time. This requires Sender to generate new messages, *and* Receiver to correctly infer their meaning. If a *blue circle* is accurately reconstructed, then agents do generalize.

For all the considered settings, we split the possible distinct inputs into 90% train and 10% test items. This implies that the absolute training/test set sizes increase with input dimension (this issue is further discussed in Appendix 8.4).

Finally, we only evaluate generalization for runs that successfully converged, where convergence is operationalized as $> 99.9\%$ training-set accuracy.

4 Generalization emerges "naturally" if the input space is large

Fig. 1 shows that emergent languages are able to almost perfectly generalize to unseen combinations *as long as input size* $|I|$ is sufficiently large (input size/test accuracy Spearman $\rho = 0.86$, $p \approx 0$). The figure also shows that the *way* in which a large input space is obtained (manipulating i_{att} or i_{val}) does not matter (no significant accuracy difference between the bracketed runs, according to a set of t-tests with $p > 0.01$). Moreover, the correlation is robust to varying agents' capacity (Appendix 8.3; see Resnick et al. (2019) for a thorough study of how agent capacity impacts generalization and compositionality). Importantly, the effect is not simply a product of larger input sizes coming with

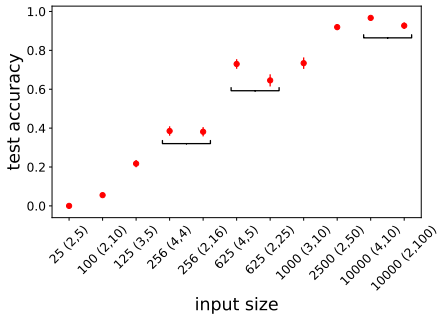


Figure 1: Average accuracy on unseen combinations as a function of input size of successful runs. The x-axis is ordered by increasing input size $|I|$. Brackets denote (i_{att}, i_{val}). Vertical bars represent the standard error of the mean (SEM). Horizontal brackets group settings with same $|I|$ but different (i_{att}, i_{val}).

larger training corpora, as we replicate it in Appendix 8.4 while keeping the number of distinct training examples fixed, but varying input *combinatorial variety*. What matters is that, in the training data, specific attribute values tend to occur with a large range of values from other attributes, providing a cue about the composite nature of the input.

That languages capable to generalize will only emerge when the input is varied enough might seem obvious, and it has been shown before in mathematical simulations of language emergence (Nowak et al., 2000), as well as in studies of deep network inductive biases (Zhao et al., 2018). However, our result suggests an important *caveat* when interpreting experiments based on small input environments that report failures in the generalization abilities of deep networks (e.g., Kottur et al., 2017; Lake and Baroni, 2018). Before assuming that special architectures or training methods are needed for generalization to emerge, such experiments should be repeated with much larger/varied input spaces, where it is harder for agents to develop ad-hoc strategies overfitting the training data and failing to generalize.

We also considered the relation between channel capacity $|C|$ and language emergence. Note that $|C| \geq |I|$ is a prerequisite for successful communication, and a perfectly compositional language could already generalize at the lower $|C| = |I|$ bound. Indeed, limiting channel capacity has been proposed as an important constraint for the emergence of compositionality (Nowak and Krakauer, 1999). However, we find that, when $|I|$ is sufficiently large to support generalization, our deep agents need $|C| > |I|$ in order to even converge at training time. The *minimum* $|C|/|I|$ ratio across all converging runs for each configuration with $|I| \geq 625$ (the settings where we witness generalizing languages) is on average 5.9 (s.d.: 4.4).

Concretely, this implies that none of our successful languages is as compact as a minimal fully-compositional solution would afford. Appendix 8.5 reports experiments focusing, more specifically, on the relation between channel capacity and generalization, showing that it is essential for $|C|$ to be above a large threshold to reach near-perfect accuracy, and further increasing $|C|$ beyond that does not hamper generalization.

5 Generalization does not require compositionality

Having established that emergent languages *can* generalize to new composite concepts, we test whether languages that generalize better are also more compositional. Since *bosdis* and *topsim* correlate with $|C|$ (Appendix 8.6), we compute Spearman correlations between test accuracy and compositionality metrics across all converging runs of each (i_{att}, i_{val}, c_{len}, c_{voc}) configuration separately. Surprisingly, in just 4 out of 141 distinct settings the correlation is significant ($p < 0.01$) for at least 1 measure.[4]

We further analyze the (i_{att}=2, i_{val}=100, c_{len}=3, c_{voc}=100) setting, as it has a large number of generalizing runs, and it is representative of the general absence of correlation we also observe elsewhere. Fig. 2 confirms that even non-compositional languages (w.r.t. any definition of compositionality) can generalize well. Indeed, for very high test accuracy ($> 98\%$), we witness a large spread of *posdis* (between 0.02 and 0.72), *bosdis* (between 0.03 and 0.4) and *topsim* (between 0.11 and 0.64). In other words, deep agents are able to communicate about new attribute combinations while using non-compositional languages. We note moreover that even the most compositional languages according to any metric are far from the theoretical maximum ($= 1$ for all metrics).

We observe however that the top-left quadrants of Fig. 2 panels are empty. In other words, it never happens that a highly compositional language has low accuracy. To verify this more thoroughly, for each compositionality measure μ, we select those languages, among *all converging runs in all con-*

[4] 3, 3 and 1 (different) significant settings for *topsim*, *posdis* and *bosdis*, respectively.

figurations, that have $\mu > 0.5$, and compute the proportion of them that reaches high test accuracy (> 0.80). We find that this ratio equates 0.90, 0.50, and 0.11 for *posdis*, *bosdis*, and *topsim* respectively. That is, while compositionality is not a necessary condition for generalization, it appears that the strongest form of compositionality, namely *posdis*, is at least sufficient for generalization. This provides some evidence that compositionality is still a desirable feature, as further discussed in Section 6.

We gain further insights on what it means to generalize without full compositionality by taking a deeper look at the language shown in red in Fig. 2, that has near-perfect generalization accuracy (>99%), and whose *posdis* score (0.70), while near the relative best, is still far from the theoretical maximum (we focus on *posdis* since it is the easiest compositional strategy to qualitatively characterize). As its behavior is partially interpretable, this "medium-*posdis*" language offered us clearer insights than more strongly entangled cases. We partially analyze one of the latter in Appendix 8.7.

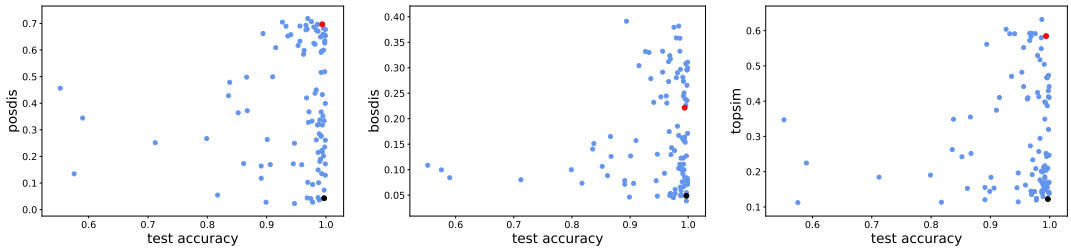
Note that, with (i_{att}=2, i_{val}=100), a (c_{len}=2, c_{voc}=100) channel should suffice for a perfectly positionally disentangled strategy. Why does the analyzed language use (c_{len}=3) instead? Looking at its mutual information profile (Appendix Table 5), we observe that positions 2 and 3 (*pos2* and *pos3*) are respectively denoting attributes 2 and 1 (*att2* and *att1*): *pos3* has high mutual information with *att1* and low mutual information with *att2*; the opposite holds for *pos2*. The remaining position, *pos1*, could then be simply redundant with respect to the others, or encode noise ignored by Receiver. However, this is not quite the case, as the language settled instead for a form of "leaky disentanglement". The two disentangled positions do most of the job, but the third, more entangled one, is still necessary for perfect communication.

To see this, consider the ablations in Table 1. Look first at the *top* block, where the trained Receiver of the relevant run is fed messages with the symbol in one original position preserved, the others shuffled. Confirming that communication is largely happening by disentangled means, preserving *pos2* alone suffices to have Receiver guessing a large majority of *att2* values, and keeping *pos3* unchanged is enough to guess almost 90% of *att1* values correctly. Conversely, preserving *pos1* alone causes a complete drop in accuracy for both attributes. However, neither *pos2* nor *pos3* are sufficient on their own to perfectly predict the corresponding attributes. Indeed, the results in the *bottom* block of the table (one symbol shuffled while the others stay in their original position) confirm that *pos1* carries useful complementary information: when fixing the latter *and either one of the other positions*, we achieve 100% accuracy for the relevant attribute (*att2* for *pos1+pos2* and *att1* for *pos1+pos3*), respectively.

In sum, *pos2* and *pos3* largely specialized as predictors of *att2* and *att1*, respectively. However, they both have a margin of ambiguity (in *pos2* and *pos3* there are 96 and 98 symbols effectively used, respectively, whereas a perfect 1-to-1 strategy would require 100). When the symbols in these positions do not suffice, *pos1*, that can refer to both attributes, serves a disambiguating role. We quantified this complementary function as follows. We define the cue validity of s_p (symbol in position p) w.r.t an attribute a as $CV(s_p, a) = \max_{\bar{a}} P(\bar{a}|s_p)$, where $\bar{a}$ iterates over all possible values of a. $CV(s_{pos1}, att2)$ is significantly higher in those (train/test) messages where $CV(s_{pos2}, att2)$ is below average. Similarly, $CV(s_{pos1}, att1)$ is significantly higher in messages where $CV(s_{pos3}, att1)$ is below average ($p \approx 0$ in both cases). We might add that, while there is a huge difference between our simple emergent codes and natural languages, the latter are not perfectly disentangled either, as they feature extensive lexical ambiguity, typically resolved in a phrasal context (Piantadosi et al., 2012).

		att1	*att2*	*both atts*
fixing	*pos1*	1	3	0
1 position	*pos2*	1	68	0
	pos3	89	1	1
shuffling	*pos1*	89	69	61
1 position	*pos2*	100	3	3
	pos3	1	100	1

Table 1: Feeding shuffled messages from the analyzed language to the corresponding trained Receiver. Average percentage accuracy across 10 random shufflings (s.d. always ≈ 0) when: *top*: symbols in all positions but one are shuffled across the data-set; *bottom*: symbols in a single position are shuffled across the data-set. The data-set includes all training and test messages produced by the trained Sender and correctly decoded in their original form by Receiver (>99% of total messages).



(a) *posdis* (b) *bosdis* (c) *topsim*

Figure 2: Compositionality in function of generalization. Each point represents a successful run in the (i_{att}=2, i_{val}=100, c_{len}=3, c_{voc}=100) setting. Red and black points correspond respectively to the medium- and low-disentanglement languages analyzed in Section 5 and Appendix 8.7.

6 Compositionality and ease of transmission

The need to generalize to new composite inputs does not appear to constitute a sufficient pressure to develop a compositional language. Given that compositionality is ubiquitous in natural language, we conjecture that it has other beneficial properties, making it advantageous once agents chanced upon it. Compositional codes are certainly easier to read out by humans (as shown by our own difficulty in qualitatively analyzing highly entangled languages), and we might hypothesize that this ease-of-decoding is shared by computational agents. A long tradition of subject studies and computational simulations has shown that the need to transmit a language across multiple generations or to populations of new learners results in the language being more compositional (e.g., Kirby, 2001; Kirby et al., 2015; Verhoef et al., 2016; Cornish et al., 2017; Cogswell et al., 2019; Guo et al., 2019; Li and Bowling, 2019). Our next experiments are closely related to this earlier work, but we adopt the opposite perspective. Instead of asking whether the pressure to transmit a language will make it more compositional, we test whether languages that have already emerged as compositional, being easier to decode, are more readily transmitted to new learners.[5]

Specifically, we run 30 games in the largest input setting (i_{att}=2, i_{val}=100), varying the channel parameters. We select the pairs of agents that achieved a high level of generalization accuracy ($\geq$0.80). Next, following the paradigm of Li and Bowling (2019), we freeze Sender, and train a new Receiver from scratch. We repeat this process 3 times per game, initializing new Receivers with different random seeds. Once the newly formed pair of agents is successful on the training set, we measure its test accuracy. We also report speed of learning, measured by area under the epochs vs. training accuracy curve. We experiment with three Receiver architectures. The first two, GRU (500) and GRU (50), are GRUs with hidden layer sizes of 500 (identical to the original Receiver) and 50, respectively. The third is a two-layer Feed-Forward Network (FFN) with a ReLu non-linearity and hidden size 500. The latter Receiver takes the flattened one-hot representation of the message as its input. This setup allows probing ease of language transmission across models of different complexity. We leave the study of language propagation across multiple generations of speakers to future work.

Results in the same setting studied in Section 5 are presented in Table 2 (experiments with other setups are in Appendix 8.8). Both learning speed and generalization accuracy of new Receivers are *strongly positively correlated with degree of compositionality*. The observed correlations reach values almost as high as 0.90 for learning speed and 0.80 for generalization, supporting our hypothesis that, when emergent languages are compositional, they are simpler to understand for new agents, including smaller ones (GRU (50)), and those with a different architecture (FFN).

7 Discussion

The natural emergence of generalization There has been much discussion on the generalization capabilities of neural networks, particularly in linguistic tasks where humans rely on compositionality (e.g., Fodor and Lepore, 2002; Marcus,

[5] Li and Bowling (2019) established this for hand-crafted languages; we extend the result to spontaneously emerging ones.

	posdis			*bosdis*			*topsim*		
	GRU(500)	GRU(50)	FFN	GRU(500)	GRU(50)	FFN	GRU(500)	GRU(50)	FFN
Learning Speed	0.87	0.71	0.35	0.85	0.68	0.33	0.87	0.71	0.35
Generalization	0.80	0.55	0.50	0.81	0.55	0.51	0.79	0.54	0.48

Table 2: Spearman correlation between compositionality metrics and ease-of-transmission measures for (i_{att}=2, i_{val}=100, c_{len}=3, c_{voc}=100). All values are statistically significant ($p < 0.01$).

2003; van der Velde et al., 2004; Brakel and Frank, 2009; Kottur et al., 2017; Lake and Baroni, 2018; Andreas, 2019; Hupkes et al., 2019; Resnick et al., 2019). In our setting, the emergence of generalization is very strongly correlated with variety of the input environment. While this result should be replicated in different conditions, it suggests that it is dangerous to study the generalization abilities of neural networks in "thought experiment" setups where they are only exposed to a small pool of carefully-crafted examples. Before concluding that garden-variety neural networks do not generalize, the simple strategy of exposing them to a richer input should always be tried. Indeed, even studies of the origin of human language conjecture that the latter did not develop sophisticated generalization mechanisms until pressures from an increasingly complex environment forced it to evolve in that direction (Bickerton, 2014; Hurford, 2014).

Generalization without compositionality Our most important result is that *there is virtually no correlation* between whether emergent languages are able to generalize to novel composite inputs and the presence of compositionality in their messages (Andreas (2019) noted in passing the emergence of non-compositional generalizing languages, but did not explore this phenomenon systematically). Supporting generalization to new composite inputs is seen as one of the core purposes of compositionality in natural language (e.g., Pagin and Westerståhl, 2010). While there is no doubt that compositional languages do support generalization, we also found other systems spontaneously arising that generalize without being compositional, at least according to our intuitive measures of compositionality. This has implications for the ongoing debate on the origins of compositionality in natural language, (e.g., Townsend et al., 2018, and references there), as it suggests that the need to generalize alone might not constitute a sufficient pressure to develop a fully compositional language. Our result might also speak to those linguists who are exploring the non-fully-compositional corners of natural language (e.g., Goldberg, 2019). A thorough investigation of neural network codes that can generalize while being partially entangled might shed light on similar phenomena in human languages. Finally, and perhaps most importantly, recent interest in compositionality among AI researchers stems from the assumption that compositionality is crucial to achieve good generalization through language (e.g., Lake and Baroni, 2018; Lazaridou et al., 2018; Baan et al., 2019). Our results suggest that the pursuit of generalization might be separated from that of compositionality, a point also recently made by Kharitonov and Baroni (2020) through hand-crafted simulations.

What is compositionality good for? We observed that positional disentanglement, while not necessary, is sufficient for generalization. If agents develop a compositional language, they are then very likely to be able to use it correctly to refer to novel inputs. This supports the intuition that compositional languages are easier to fully understand. Indeed, when training new agents on emerged languages that generalize, it is much more likely that the new agents will learn them fast and thoroughly (i.e., they will be able to understand expressions referring to novel inputs) if the languages are already compositional according to our measures. That language transmission increases pressure for structured representations is an established fact (e.g., Kirby et al., 2015; Cornish et al., 2017). Here, we reversed the arrow of causality and showed that, if compositionality emerges (due to chance during initial language development), it will make a language easier to transmit to new agents. Compositionality might act like a "dominant" genetic feature: it might arise by a random mutation but, once present, it will survive and thrive, as it guarantees that languages possessing it will generalize and will be easier to learn. From an AI perspective, this suggests that trying to enforce compositionality during language emergence will increase the odds

of developing languages that are quickly usable by wide communities of artificial agents, that might be endowed with different architectures. From the linguistic perspective, our results suggest an alternative view of the relation between compositionality and language transmission–one in which the former might arise by chance or due to other factors, but then makes the resulting language much easier to be spread.

Compositionality and disentanglement Language is a way to *represent* meaning through discrete symbols. It is thus worth exploring the link between the area of language emergence and that of representation learning (Bengio et al., 2013). We took this route, borrowing ideas from research on disentangled representations to craft our compositionality measures. We focused in particular on the intuition that, if emergent languages must denote ensembles of primitive input elements, they are compositional when they use symbols to univocally denote input elements independently of each other.

While the new measures we proposed are not highly correlated with topographic similarity, in most of our experiments they did not behave significantly differently from the latter. On the one hand, given that topographic similarity is an established way to quantify compositionality, this serves as a sanity check on the new measures. On the other, we are disappointed that we did not find more significant differences between the three measures.

Interestingly one of the ways in which they did differ is that, when a language is positionally disentangled, (and, to a lesser extent, bag-of-symbols disentangled), it is very likely that the language will be able to generalize–a guarantee we don't have from less informative topographic similarity.

The representation learning literature is not only proposing disentanglement measures, but also ways to encourage emergence of disentanglement in learned representations. As we argued that compositionality has, after all, desirable properties, future work could adapt methods for learning disentangled representations (e.g., Higgins et al., 2017; Kim and Mnih, 2018) to let (more) compositional languages emerge.

Acknowledgments

We thank the reviewers for feedback that helped us to make the paper clearer.

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *Proceedings of ICLR*.

Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. 2019. On the realization of compositionality in neural networks. In *Proceedings of ACL BlackboxNLP Workshop*.

Y. Bengio, A. Courville, and P. Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8).

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

Alican Bozkurt, Babak Esmaeili, Dana H Brooks, Jennifer G Dy, and Jan-Willem van de Meent. 2019. Evaluating combinatorial generalization in variational autoencoders. *arXiv preprint arXiv:1911.04594*.

Philémon Brakel and Stefan Frank. 2009. Strong systematicity in sentence processing by simple recurrent networks. In *Proceedings of CogSci*.

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019. Anti-efficient encoding in emergent communication. In *Proceedings of NeurIPS*.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Proceedings of NeurIPS*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. 2019. Emergence of compositional language with deep generational transmission. *arXiv preprint arXiv:1904.09067*.

Hannah Cornish, Rick Dale, Simon Kirby, and Morten Christiansen. 2017. Sequence memory constraints give rise to language-like structure through iterated learning. *PLoS ONE*, 12(1):1–18.

Jerry Fodor and Ernest Lepore. 2002. *The Compositionality Papers*. Oxford University Press, Oxford, UK.

Adele Goldberg. 2019. *Explain Me This: Creativity, Competition, and the Partial Productivity of Constructions*. Princeton University Press, Priinceton, NJ.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. 2019. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of ICLR*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2019. The compositionality of neural networks: integrating symbolism and connectionism. *arXiv preprint arXiv:1908.08351*.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Eugene Kharitonov and Marco Baroni. 2020. Emergent language generalization and acquisition speed are not tied to compositionality. *arXiv preprint arXiv:2004.03420*.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on emergence of language in games. In *Proceedings of EMNLP (System Demonstrations)*.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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, 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 EMNLP*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of ICML*.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of ICLR*.

David Lewis. 1969. *Convention*. Harvard University Press, Cambridge, MA.

Fushan Li and Michael Bowling. 2019. Ease-of-teaching and language structure from emergent communication. *arXiv preprint arXiv:1906.02403*.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of ICML*.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Gary Marcus. 2003. *The Algebraic Mind*. MIT Press, Cambridge, MA.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *AAAI*.

Ryan Nefdt. 2020. A puzzle concerning compositionality in machines. *Mind and Machines*. In press.

Martin Nowak and David Krakauer. 1999. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033.

Martin A Nowak, Joshua B Plotkin, and Vincent AA Jansen. 2000. The evolution of syntactic communication. *Nature*, 404(6777):495.

Peter Pagin and Dag Westerståhl. 2010. Compositionality II: Arguments and problems. *Philosophy Compass*, 5(3):265–282.

Barbara Partee. 2004. *Compositionality in Formal Semantics*. Blackwell, Malden, MA.

Steven Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Cinjon Resnick, Abhinav Gupta, Jakob Foerster, Andrew M Dai, and Kyunghyun Cho. 2019. Capacity, bandwidth, and compositionality in emergent language learning. *arXiv preprint arXiv:1910.11424*.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. 2015. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. 2019. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *Proceedings of ICML*.

Simon Townsend, Sabrina Engesser, Sabine Stoll, Klaus Zuberbühler, and Balthasar Bickel. 2018. Compositionality in animals and humans. *PLOS Biology*, 16(8):1–7.

Frank van der Velde, Gwendid van der Voort van der Kleij, and Marc de Kamps. 2004. Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science*, 16(1):21–46.

Tessa Verhoef, Simon Kirby, and Bart de Boer. 2016. Iconicity and the emergence of combinatorial structure in language. *Cognitive Science*, 40(8):1969–1994.

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

Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. 2018. Bias and generalization in deep generative models: An empirical study. In *Advances in Neural Information Processing Systems*.

8 Appendix

8.1 Grid search over (i_{att}, i_{val}, c_{len}, c_{voc})

We report in Table 3 the different (i_{att}, i_{val}, c_{len}, c_{voc}) combinations we explored. They were picked according to the following criteria:

- $|C| \geq |I|$ so that agents are endowed with enough different messages to refer to all inputs;
- discard some $|C| >> |I|$ so that we have approximately the same number of settings per (i_{att}, i_{val}) (between 13 and 15 different (c_{voc}, c_{len}));
- include some (c_{voc}, c_{len}) that are large enough that they can be tested with all the considered (i_{att}, i_{val}).

Unless it is mentioned explicitly, we run 10 different initializations per setting.

Table 3 shows that, for large $|I|$, GRU-agents need $|C|$ *strictly* larger than $|I|$. This suggests that, for large $|I|$, the emergence of a perfectly non-ambiguous compositional languages, where each message symbol denotes only one attribute value *and* each value attribute is denoted by only one message symbol, is impossible.

8.2 Behavior of the compositionality measures on hand-crafted miniature languages

We construct 3 simple miniature languages to illustrate the different behaviors of *topsim*, *posdis* and *bosdis*: Lang1, Lang2 and Lang3. We fix $i_{att} = 2$, $i_{val} = 4$, $c_{len} = 3$ and $c_{voc} = 8$.[6] Table 4 shows the input-message mappings of each language and reports their degree of compositionality. Note that all languages respect a bijective mapping between inputs and messages.

Lang1 is perfectly *posdis*-compositional (*posdis*=1). However, *topsim* < 1, as 2 symbols encode one attribute (we need the first two symbols to recover the value of the first attribute). Lang1 is penalized by *topsim* because it does not have a one-to-one attribute-position mapping; a feature that arguably is orthogonal to compositionality.

Lang2 and Lang3 are equally *topsim*-compositional. Nonetheless, they differ fundamentally in terms of the type of compositionality they feature. If Lang2 is more *posdis*-compositional, Lang3 is perfectly *bosdis*-compositional.

8.3 Generalization for different agents' capacity

We demonstrated in the main paper that agent's generalization correlates with input size. In fact, agents can successfully reconstruct new attribute combinations if trained on large input spaces. This could be due to agents overfitting when presented with few training samples. To test this hypothesis, we repeat the training/evaluation experiments with GRU agents of different capacities in the following settings: (i_{att}=2, i_{val}=10), a small input space where agents do not generalize; and (i_{att}=2, i_{val}=100), a large input space where agents generalize.[7] Fig. 3 shows that, even for small-capacity agents (one-layer GRU with hidden state of size 100), test accuracy is 0 for (i_{att}=2, i_{val}=10). Moreover, agents do not overfit when trained on (i_{att}=2, i_{val}=100) even with two-layer GRUs with hidden state of size 500.

8.4 Input space density

We showed in the main paper that generalization positively correlates with $|I|$. We further investigate here whether it is simply the increasing abso-

[6] Only Lang3 uses the whole available c_{voc}

[7] We only report experiments with GRUs, but the same results were replicated with differently-sized LSTMs.

(i_{val}, i_{att}) \ c_{voc} / c_{len}	5				10				50				100			
	2	3	4	{6,8}	2	3	4	{6,8}	2	3	4	{6,8}	2	3	4	{6,8}
(4,4)			X	X		X	X	X	X			X	X			X
(5,2)	X	X	X	X	X	X	X	X	X			X	X			X
(5,3)		X	X	X		X	X	X	X			X	X			X
(5,4)			X	X		X	X	X	X			X	X			X
(10,2)		-	X	X	X	X	X	X	X			X	X			X
(10,3)				X		-	X	X	X	X	X	X	X			X
(10,4)				{-, X}			-	X		X	X	X	-	X	X	X
(16,2)			-	X		X	X	X	X			X	X			X
(25,2)			-	X		-	X	X	X			X	X			X
(50,2)				X			-	X	-	X	X	X	X	X	X	X
(100,2)				{-, X}			-	X		X	X	X	-	X	X	X

Table 3: Grid search. 'X' indicates tested settings with at least one successful run. '-' indicates tested settings without any successful run. Finally, blank cells correspond to settings that were not explored for the reasons indicated in the text.

Input	Lang1	Lang2	Lang3
0,0	0,0,0	0,0,0	0,0,4
0,1	0,0,1	0,0,1	0,0,5
0,2	0,0,2	0,0,2	0,0,6
0,3	0,0,3	0,0,3	0,0,7
1,0	0,1,0	1,2,0	1,4,1
1,1	0,1,1	1,2,1	1,5,1
1,2	0,1,2	1,2,2	1,6,1
1,3	0,1,3	1,2,3	1,7,1
2,0	2,0,0	2,3,0	2,4,2
2,1	2,0,1	2,3,1	2,5,2
2,2	2,0,2	2,3,2	2,6,2
2,3	2,0,3	2,3,3	2,7,2
3,0	2,1,0	3,1,0	3,4,3
3,1	2,1,1	3,1,1	3,3,5
3,2	2,1,2	3,2,1	3,3,6
3,3	2,1,3	3,3,1	3,3,7
topsim	0.82	0.75	0.75
posdis	1	0.79	0.43
bosdis	0.42	0.13	1

Table 4: Input-message mappings and compositionality measures for the miniature languages.

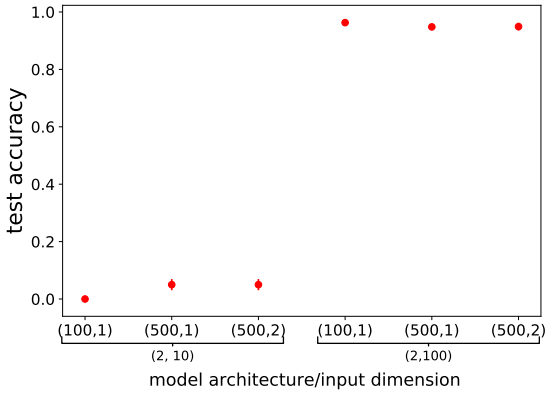


Figure 3: Average accuracy on unseen combinations as a function of agents capacity ((hidden size, number of layers)) for input sizes ($i_{att} = 2$, $i_{val} = 10$) and ($i_{att} = 2$, $i_{val} = 100$). Vertical bars represent SEM.

lute number of distinct training samples that is at the root of this phenomenon, or whether the variety of seen inputs also plays a role, independently of absolute input size.

To verify this, we design an experiment where we keep the absolute number of distinct input samples constant, but we change their *density*, defined as the proportion of sampled items over the the size of the space they are sampled from. When sampling points from a small space, on average each value of an attribute will occur with a larger range of values from other attributes, compared to a larger space, which might provide more evidence about the combinatorial nature of the underlying space.

In practice, we fix (c_{len}=3, c_{voc}=100, i_{att}=2) and sample 10000 points from spaces with i_{val}=100 (density=1), i_{val}=140 (density=0.51) and i_{val}=200 (density=0.25), respectively. As usual, we use 90% of the data for training, 10% for testing. In all cases, we make sure that all values are seen at least once during training (as visually illustrated in Fig. 4).

We obtain test accuracies of 92.7%, 66.7% and 22.8% for densities 1, 0.51 and 0.25 respectively. That is, the high generalization observed in the main paper is (also) a consequence of density, and hence combinatorial variety, of the inputs the agents are trained on, and not (only) of the number of training examples.

8.5 Impact of channel capacity on generalization

We showed in the main paper that generalization is very sensitive to input size. In this section, we focus on the relation between channel capacity $|C|$ and generalization.

First, when we aggregate across input sizes, Fig. 5 shows that $|C|$ has a just small effect on generalization, with a low Spearman correlation $\rho = 0.14$. Next, if we study this relation for specific large $|I|$ (where we observe generalization), we notice in Fig. 6 that agents need to be endowed with a $|C|$ above a certain threshold, with $\frac{|C|}{|I|} > 1$, in order to achieve almost perfect generalization. Moreover, contradicting previous claims (e.g., Kottur et al., 2017), having $|C| >> |I|$ does not harm generalization.

8.6 Impact of channel capacity on the compositionality measures

A good compositionality measure should describe the structure of the language independently of the used channel, so the corresponding score should not be greatly affected by $|C|$. However, Fig. 7 shows clear negative correlations of both *topsim* and *bosdis* with $|C|$.

8.7 Analysis of example medium- and low-posdis languages

We present more data about the *medium-posdis* language analyzed in the main article, and we provide comparable evidence for a language with similarly excellent generalization (>99%) but very low posdis (0.05), that we will call here *low-posdis*. The latter language is depicted in black in Fig. 2 of the main text. Both languages come from the training configuration with 2 100-valued input attributes and 3 100-symbol positions.

Mutual information profiles. Table 5 reports mutual information for the two languages. Note how the highly entangled *low-posdis* is almost uniform across the table cells.

	medium-posdis		*low-posdis*	
	att1	*att2*	*att1*	*att2*
pos1	1.10	2.01	1.72	1.95
pos2	0.19	4.16	1.74	1.71
pos3	4.44	0.13	2.16	1.77

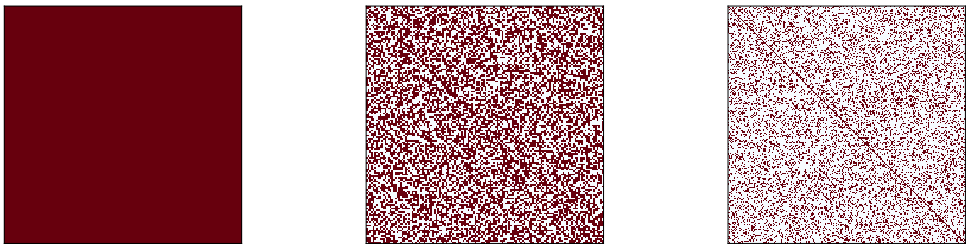
Table 5: Mutual information of each position with each attribute for the studied languages.

Vocabulary usage. Considering all messages produced after training for the full training and test set inputs, effective vocabulary usage for *pos1*, *pos2* and *pos3* are as follows (recall that 100 symbols are maximally available):

- *medium-posdis*: 91, 96, 98
- *low-posdis*: 99, 99, 100

Although vocabulary usage is high in both cases, *medium-posdis* is slightly more parsimonious than *low-posdis*.

Ablation studies. Table 6 reports ablation experiments with both languages. The results for *medium-posdis* are discussed in the main text. We observe here how virtually any ablation strongly impacts accuracy in denoting either attribute by the highly entangled *low-posdis* language. This points to another possible advantage of (partially) disentangled languages such as *medium-posdis*: since *pos2* and *pos3* are referring to *att2* and *att1* independently, in ablations in which they are untouched, Receiver can still retrieve partial information, by often successfully guessing the attribute they each refer to. We also report in the table the effect of shuffling *across the positions* of each message. This is very damaging not only for *medium-posdis*, but for *low-posdis* as well, showing that even the latter is exploiting positional information, albeit in an inscrutable, highly entangled way. Note in Fig. 2 of the main article that neither language has high *bos*.



(a) density=1 (b) density=0.51 (c) density=0.25

Figure 4: Sampling the same number of input instances (= 10000) with different densities. The axes of the shown matrices represent the values of two attributes, with the dark-red cells standing for inputs that were sampled. We ensure that each value of each attribute is picked at least once by always sampling the full diagonal.

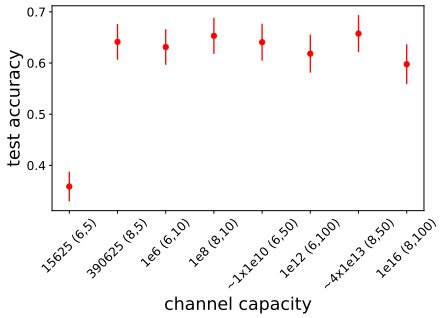


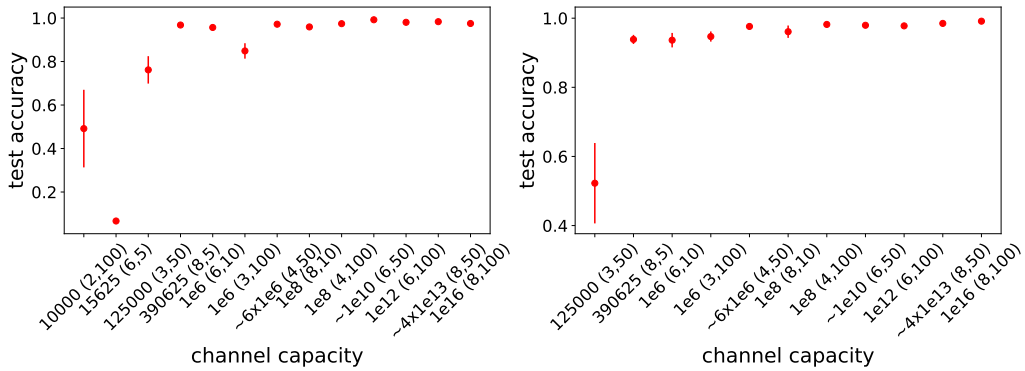
Figure 5: Average accuracy on unseen combinations as a function of channel capacity of the successful runs. The x-axis is ordered by increasing channel capacity. In the brackets we note (c_{len}, c_{voc}). Vertical bars represent SEM.

8.8 Effect of channel capacity on ease of transmission

Table 7 replicates the ease-of-transmission analysis presented in the main text across various channel capacities. We observe in most cases a significantly positive correlation, that is even higher (1) for larger Receivers and (2) for emergent languages with shorter messages (smaller c_{len}).

		medium-posdis			*low-posdis*		
		att1	*att2*	*both*	*att1*	*att2*	*both*
fixing	*pos1*	1	3	0	4	5	0
1 position	*pos2*	1	68	0	4	4	0
	pos3	89	1	1	8	5	0
shuffling	*pos1*	89	69	61	31	18	6
1 position	*pos2*	100	3	3	30	25	8
	pos3	1	100	1	15	20	3
shuffling	*msg*	1	2	0	2	4	0

Table 6: Feeding shuffled messages from the *medium-posdis* and *low-posdis* languages to the corresponding trained Receivers. Mean percentage accuracy across 10 random shufflings (standard deviation is always $\approx$ 0) when: *top*: symbols in all positions but one are shuffled across the data-set; *middle*: symbols in a single position are shuffled across the data-set; *bottom*: shuffling the symbols within each message (ensuring all symbols move). The data-set includes all training and test messages produced by the trained Sender and correctly decoded in their original form by Receiver (>99% of total messages).



(a) ($i_{att} = 2, i_{val} = 50$) (b) ($i_{att} = 2, i_{val} = 100$)

Figure 6: Average accuracy on unseen combinations as a function of channel capacity of the successful runs for two different (i_{att}, i_{val}). The x-axis is ordered by increasing channel capacity. In the brackets we note (c_{len}, c_{voc}). Vertical bars represent SEM.

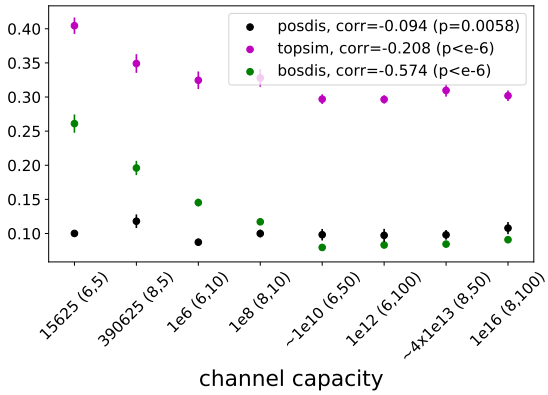


Figure 7: Average of different compositionality measures in function of channel capacity (c_{voc}, c_{len})). Vertical bars represent SEM.

(c_{len}, c_{voc})	measure	*posdis*			*bosdis*			*topsim*		
		GRU (500)	GRU (50)	FFN (500)	GRU (500)	GRU (50)	FFN (500)	GRU (500)	GRU (50)	FFN (500)
(3,50)	Learning Speed	0.82	0.78	0.74	0.71	0.67	0.62	0.72	0.74	0.66
	Generalization	0.77	0.77	0.75	0.61	0.62	0.66	0.75	0.76	0.74
(4,50)	Learning Speed	0.79	0.44	0.48	0.76	0.51	0.47	0.89	0.59	0.61
	Generalization	0.73	-	0.50	0.77	0.27	0.54	0.84	0.41	0.61
(6,50)	Learning Speed	0.82	0.77	0.79	0.79	0.76	0.77	0.89	0.85	0.87
	Generalization	0.78	0.56	0.69	0.76	0.55	0.67	0.85	0.65	0.77
(8,50)	Learning Speed	0.75	0.56	0.78	0.80	0.68	0.78	0.75	0.55	0.71
	Generalization	0.67	0.27	0.68	0.78	0.41	0.70	0.53	-	0.54
(10,50)	Learning Speed	0.51	0.29	0.60	0.42	0.31	0.48	0.72	0.49	0.73
	Generalization	0.39	-	0.44	0.47	-	0.36	0.41	0.27	0.57
(12,50)	Learning Speed	-	-	-	0.33	-	-	0.49	-	0.35
	Generalization	-	-0.28	-	-	-	-	-	-	-
(3,100)	Learning Speed	0.87	0.71	0.35	0.85	0.68	0.33	0.87	0.71	0.35
	Generalization	0.80	0.55	0.50	0.81	0.55	0.51	0.79	0.54	0.48
(4,100)	Learning Speed	0.84	0.54	0.43	0.82	0.54	0.46	0.86	0.57	0.49
	Generalization	0.82	0.38	0.47	0.80	0.39	0.47	0.82	0.41	0.48
(6,100)	Learning Speed	0.88	0.83	0.80	0.89	0.78	0.78	0.94	0.87	0.83
	Generalization	0.87	0.68	0.68	0.90	0.69	0.67	0.90	0.70	0.68
(10,100)	Learning Speed	0.85	0.58	0.62	0.82	0.59	0.64	0.72	0.74	0.66
	Generalization	0.86	0.39	0.47	0.81	0.50	0.37	0.72	0.35	0.46
(8,100)	Learning Speed	0.73	0.58	0.65	0.79	0.59	0.65	0.70	0.57	0.66
	Generalization	0.69	0.39	0.37	0.67	0.37	0.37	0.49	0.34	0.46
(12,100)	Learning Speed	0.39	-	0.27	0.69	-	0.40	0.67	-	0.51
	Generalization	0.38	-	0.34	0.52	-	0.38	0.36	-	-
Average	Learning Speed	0.75	0.62	0.61	0.72	0.63	0.59	0.79	0.67	0.62
	Generalization	0.71	0.42	0.54	0.72	0.49	0.51	0.70	0.51	0.63

Table 7: Statistically significant ($p < 0.01$) Spearman correlations between retraining performance (measured by new Receiver Learning Speed and Generalization) and compositionality measures (*posdis*, *bosdis* and *topsim*) for ($i_{att} = 2, i_{val} = 100$) and different channel capacity. '-' indicates no significant correlations.

Chapter 6

General Discussion

This manuscript investigates whether general-purpose NNs, when playing a simple communication game, develop some language universals. In this part, we first summarize the main conclusions of the different chapters and provide a unified framework that attempts to capture how the emergence of language universal properties could be explained by a trade-off between the communication success in a referential game and different constraints at various levels of communication. Second, we highlight how the emergent communication setting could be used to identify NNs' biases. Specifically, we argue that this setting offers an alternative way to standard "BlackBox" studies [73, 74, 3] to probe NNs' biases relying on interactive learning. Finally, we show some of the limitations of our studies and provide some directions to pursue in future research.

6.1 Universal language properties in emergent languages

In this work, we found that some human-like regularities arose spontaneously while others were missing in NN languages (see Table 6.1). However, in both cases, communicating NNs offered a framework to study the origin of the agreement or disagreement between their emergent language and human languages. Specifically, we related these

observations to the presence/absence of some constraints that are either on the side of learners (**Chapter 2** and **3**), the communication channel (**Chapter 4**), the environment (**Chapter 5**), or the language functionality (**Chapter 5**). In all these cases, the constraints act in the opposite direction than the communication success constraint imposed by the game, leading to a trade-off between them. Concretely, when NN and human languages share a property, we hypothesize that the latter depends on both learners' communicative constraints. By varying these communicative constraints, we can examine the origin of the shared property. On the opposite, if NN languages depart from the human ones on a specific property, we can append hypothetical human "cognitive" constraints to NNs and causally assess which ones are needed to induce the emergence of that property.

The former outcome, where NN agents develop a cross-linguistic property, was encountered in **Chapter 4**. We found that NNs produce efficient and low-complex color-naming systems. We connected this to the discrete nature of their communication channel. This finding suggests that the discrete nature of our communication could also be responsible for the efficient, low-complex nature of *our* naming systems. Further experiments showing how the discrete nature of communication leads to a robust transmission of information are reported in **Appendix B**. We hope that further testing of this hypothesis on human subjects, building on experimental semiotics studies [114, 86], would strengthen our understanding of the roots of efficiency in human naming-systems.

The second outcome where emergent languages depart from natural language was observed in **Chapter 2**. We found that communicating NNs do not develop a ZLA-obeying language. We related this to the absence of the least-effort pressure towards brevity on the speaker side, which is counter-balanced by the preference of long, more discriminable messages on the listener side. Such a preference for easy discriminability of messages is also found in humans when the least-effort pressure is lifted [57]. Hence, our results suggest that in the absence of the need to minimize messages length in human speakers, our language should follow an anti-ZLA distribution, similar to NN languages. **Chapter 2** provides a concrete example of non-human-like NN languages

and relates this divergence to the absence of general "innate" constraints on the NNs' side. Furthermore, we show that by introducing the right constraints on NNs, using the "LazImpa" communicative system, we see the development of ZLA-obeying emergent languages, as efficient as natural language.

Finally, **Chapters 3** and **5** show a more mixed picture. In **Chapter 5**, we found that NNs, similarly to humans, can develop productive languages that can be used to generalize to never-seen objects when provided with a rich environment. However, this productivity was achieved via non-intuitively-compositional strategies, different from the ones found in natural language [35]. In other words, **Chapter 5** demonstrates that the need for generalization cannot explain the emergence of intuitively-compositional languages. Interestingly, our follow-up experiments suggest that compositionality might still be beneficial instead for languages' ease of learning. Contrary to what we did in **Chapters 2** and **4**, we did not directly intervene on this constraint to determine its effect on emergent languages' compositionality. Though, concurrent and posterior work looked at the other direction and studied if the ease of learning can encourage the emergence of compositional languages. For example, Li and Bowling showed that making the NN speaker communicate sequentially with multiple NN listeners enforces the emergence of a more compositional language [71]. Other studies used the iterated learning framework [61] to show that compositionality emerges as a trade-off between learnability (ease of acquisition through generations) and expressivity (optimization of the communication success in a referential game) [45, 97]. In **Chapter 3**, we used the emergent language framework to investigate NN biases with respect to some word order constraints. Specifically, we looked at a popular NN model for language processing, that is, LSTM architectures [50]. Herewith we found that LSTMs, similarly to humans, have a preference for information locality. On the other hand, they depart from human preferences by adopting redundant languages that display both case and fixed-word order. The former preference was hypothesized by Futrell and Levy [36] to be a consequence of our imperfect memory. Interestingly, such imperfect memory could also be found in LSTMs. Indeed, if LSTMs were introduced to allow the processing of longer sequences compared to standard recurrent networks,

they still must compress all prior information into one hidden representation for each new input. The hidden representation could thus be seen as an imperfect memory akin to ours. One interesting line for future work is to compare LSTMs' preference for information locality with Transformer models [113]. The latter, unlike LSTMs, has access to all prior inputs when dealing with sequential inputs and hence do not suffer from imperfect memory. Therefore, based on Futrell and Levy's hypothesis, we should not encounter a preference for local dependencies when experimenting with Transformers. Lastly, the preference for redundant solutions could be due to the lack of least effort minimization akin to the one observed in **Chapter 2**. We leave both investigations to future work.

Table 6.1 summarizes the different cross-linguistic regularities that we looked at across the different chapters and the constraints we found to shape them in NN models. Most of our work is based on the EGG toolkit developed by our team. We provide a detailed description of EGG in **Appendix A**.

In sum, our work shows that there is a rich future for language research aiming at connecting linguistics/cognitive science to modeling with NNs employed in communication games. The latter provides a flexible framework to study the origin of language properties. In this manuscript, we only looked at a few of them using simple environments. In future work, we can extend this framework to investigate more complex regularities such as subject-verb-object order [41], that would require the use of a more complex input space, or generalize it to more than one Speaker/Listener agent pair to study the effect of agent communities on various language properties. Furthermore, this manuscript highlights the importance of introducing the right pressures in the communication system. Indeed, it shows that these general learners lack, in some cases, the right constraints to develop human-like languages. To construct automated agents that would eventually interact with humans, we need to introduce task-agnostic constraints, such as the ones applied in **Chapter 2** in "LazImpa", allowing the emergence of more natural communication.

Chapter	*Regularity*	*Hypothesized constraint*	*Constraint's level*
Chapter 2	word length/frequency inverse correlation	least effort minimization & online processing	learners
Chapter 3	case marking/fixed word ordercorrelation	least effort minimization	learners
Chapter 3	long distance dependency minimization (locality)	memory load	learners
Chapter 4	efficient and low-complex semantic categorization	discrete communication channel	channel
Chapter 5	productivity	rich environment	environment
Chapter 5	compositionality	ease of learning	language functionality

Table 6.1: Summary of the phenomena studied in the various chapters. Our findings suggest that each of the studied cross-linguistic regularities (column 2) originates from a trade-off between communication success and a specific constraint (column 3). Each constraint can lead to one or many regularities. We represent the constraints with two different colors. Green refers to the constraints that we specifically intervened on and tested their impact on NN languages. Orange refers to constraints hypothesized by prior work and that are coherent with our observations in NN languages. These constraints could be varied in NN communication games to assess their impact on the emergent language (see text for more details).

6.2 More interpretable AI

This manuscript also tackles the questions of *how* NN communicate and *why* they communicate in this way. For example, in **Chapter 5**, we studied *how* NN agents can generalize to never-seen objects without using intuitively-compositional languages. To this end, we introduced different measures, taking inspiration from the representation learning literature [17, 77], to assess different types of compositionality. We also intervened directly on their language to evaluate the role of different symbols/positions. Concerning the *why* question, in **Chapter 4** we varied different parameters in the communication game to causally assess which of them lead to human-like color-naming systems. In particular, we looked at target-distractor distance in the game, NNs' capacity, and channel discreetness. Hence our work contributes to the understanding of NN's behavior and thus to interpretability research and explainable AI in general. This domain has recently seen an increasing interest from linguistics and psychology [73, 74, 3]. In this context, researchers have looked at how trained

models solve different NLP tasks characterizing their outputs and internal representation. In **Chapter 3**, for example, we took a different route compared to the previous literature and focused directly on uncovering NNs' "innate" biases while learning a task. That is, emergent communication provides an alternative way to investigate the linguistic skills/biases of NNs, focusing on the communicative function of language. This complements the more standard approach of exposing NNs passively to a large amount of text.

6.3 Future directions

In this thesis, we build up on the emergent communication framework to shed light on the origin of cross-linguistic properties. Another exciting research direction is to consider the opposite viewpoint and use this framework to develop better artificial agents that can eventually interact with us. Indeed, if we understand the sources of our robust and efficient communication protocol (natural language), we can encourage its emergence in communicating NN agents. Preliminary work in this direction was provided in **Chapter 2** where we introduced "LazImpa", adding laziness and impatience constraints, to encourage the emergence of a ZLA-obeying language as efficient as human languages. An interesting line for future work is to extend "LazImpa" in other challenging setups, such as story generation. Indeed, the tendency of NNs to diverge towards long redundant generation, known as text degeneration, is an unsolved problem [52]. Adding another NN that impatiently processes the generated story could lead to the generation of a non-degenerate text. It would also highlight the importance of adopting a realistic interactive setup with the right constraints in text generation tasks.

Furthermore, emergent communication could be used as a pretraining step for standard natural language processing tasks. This is interesting as the natural language processing field has recently seen the tremendous success of general-domain pretrained language models such as BERT [27] or GPT-3 [15]. In this context, emergent communication offers a natural framework for learning representations using

unlabeled data. A similar idea has recently been explored in Li and collaborators' work [72] where they explored how pretrained NNs in a referential game reach better performances on standard machine translation tasks compared to non-pretrained models. Future work could extend this study to other natural language processing tasks, grounding agents in a rich environment close to human learner's ones, and using a better constrained pretraining setup. Indeed, I believe that having a more constrained setup, such as the one explored in **Chapter 2**, would encourage the emergence of the right inductive biases, shown to be powerful to transfer in real tasks [90].

Bibliography

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Judith Aissen. Differential object marking: Iconicity vs. economy. *Natural Language & Linguistic Theory*, 21(3):435–483, 2003.

[3] Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad. Proceedings of the third blackboxnlp workshop on analyzing and interpreting neural networks for nlp. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020.

[4] Marco Baroni, Armand Joulin, Allan Jabri, Germàn Kruszewski, Angeliki Lazaridou, Klemen Simonic, and Tomas Mikolov. CommAI: Evaluating the first steps towards a useful general AI. In *Proceedings of ICLR Workshop Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/workshop`.

[5] John Batali. Computational simulations of the emergence of grammar. In James Hurford, Michael Studdert-Kennedy, and Chris Knight, editors, *Approaches to the Evolution of Language: Social and Cognitive Bases*, pages 405–426. Cambridge University Press, Cambridge, UK, 1998.

[6] John Batali. The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In *Linguistic evolution through language acquisition: Formal and computational models*. Citeseer, 1999.

[7] Chris Bentz and Ramon Ferrer Cancho. Zipf's law of abbreviation as a language universal. In *Proceedings of the Leiden workshop on capturing phylogenetic algorithms for linguistics*, pages 1–4. University of Tübingen, 2016.

[8] Brent Berlin and Paul Key. *Basic Color Terms: Their Universality and Evolution*. University of California Press, Berkeley, CA, 1969.

[9] Robert Berwick, Gabriel Beckers, Kazuo Okanoya, and Johan Bolhuis. A bird's eye view of human language evolution. *Frontiers in evolutionary neuroscience*, 4:5, 2012.

[10] Barry Blake. *Case*. MIT Press, Cambridge, MA, 2001.

[11] Johan J Bolhuis, Gabriel JL Beckers, Marinus AC Huybregts, Robert C Berwick, and Martin BH Everaert. Meaningful syntactic structure in songbird vocalizations? *PLoS biology*, 16(6):e2005157, 2018.

[12] Henry Brighton, Simon Kirby, and Kenny Smith. Cultural selection for learnability: Three principles underlying the view that language adapts to be learnable. *Language origins: Perspectives on evolution, ed. M. Tallerman*, pages 291–309, 2005.

[13] Henry Brighton, Kenny Smith, and Simon Kirby. Language as an evolutionary system. *Physics of Life Reviews*, 2(3):177–226, 2005.

[14] Ted Briscoe, editor. *Linguistic evolution through language acquisition*. Cambridge University Press, Cambridge, UK, 2002.

[15] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[16] Joan Bybee. *Language, usage and cognition*. Cambridge University Press, 2010.

[17] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[18] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, Doha, Qatar, 2014.

[19] Noam Chomsky. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger, Wesport, CT, 1986.

[20] Morten Christiansen and Simon Kirby, editors. *Language Evolution*. Oxford University Press, Oxford, UK, 2003.

[21] Morten H Christiansen and Nick Chater. Language as shaped by the brain. *Behavioral and brain sciences*, 31(5):489–509, 2008.

[22] Morten H Christiansen and Simon Ed Kirby. *Language evolution*. Oxford University Press, 2003.

[23] Benrard Comrie. *Language Universals and Linguistic Typology*. Blackwell, Malden, MA, 1981.

[24] Bernard Comrie. Numeral bases. In *The world atlas of language structures*, pages 530–533. Oxford Univ. Press, 2005.

[25] William Croft and Alan Cruse. *Cognitive Linguistics*. Cambridge University Press, Cambridge, UK, 2004.

[26] Laura de Ruiter, Anna Theakston, Silke Brandt, and Elena Lieven. Iconicity affects children's comprehension of complex sentences: The role of semantics, clause order, input and individual differences. *Cognition*, 171:202–224, 2018.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[28] Emmanuel Dupoux. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59, 2018.

[29] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[30] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

[31] Maryia Fedzechkina, Elissa Newport, and Florian Jaeger. Miniature artificial language learning as a complement to typological data. In Lourdes Ortega, Andrea Tyler, Hae In Park, and Mariko Uno, editors, *The Usage-based Study of Language Learning and Multilingualism*, pages 211–232. Georgetown University Press, Washington, DC, 2016.

[32] Maryia Fedzechkina, Elissa Newport, and Florian Jaeger. Balancing effort and information transmission during language acquisition: Evidence from word order and case marking. *Cognitive Science*, 41:416–446, 2017.

[33] Janet Dean Fodor. Setting syntactic parameters. 2001.

[34] Jerry Fodor and Ernest Lepore. *The Compositionality Papers*. Oxford University Press, Oxford, UK, 2002.

[35] Jerry Fodor and Zenon Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71, 1988.

[36] Richard Futrell and Roger Levy. Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 1, long papers*, pages 688–698, 2017.

[37] Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341, 2015.

[38] Edward Gibson. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76, 1998.

[39] Edward Gibson, Richard Futrell Steven Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. How efficiency shapes human language. *Trends in Cognitive Science*, 2019. In press.

[40] Adele Goldberg. *Constructions at work: The nature of generalization in language*. Oxford University Press, Oxford, UK, 2005.

[41] Joseph Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph Greenberg, editor, *Universals of Human Language*, pages 73–113. MIT Press, Cambridge, MA, 1963.

[42] Joseph H Greenberg. Generalizations about numeral systems. *Universals of human language*, 3:249–295, 1978.

[43] Daniel Grodner and Edward Gibson. Consequences of the serial nature of linguistic input for sentenial complexity. *Cognitive science*, 29(2):261–290, 2005.

[44] Kristina Gulordava, Paola Merlo, and Benoît Crabbé. Dependency length minimisation effects in short spans: a large-scale analysis of adjective placement in complex noun phrases. In *ACL 2015 - the 53rd annual meeting of the Association for Computational Linguistics*, Beijing, China, July 2015.

[45] Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

[46] Martin Haspelmath. *Indefinite pronouns*. Oxford University Press, 1997.

[47] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Proceedings of NIPS*, pages 2149–2159, Long Beach, CA, 2017.

[48] John Hawkins. *A Performance Theory of Order and Constituency*. Cambridge University Press, Cambridge, UK, 1994.

[49] John A Hawkins. *Efficiency and complexity in grammars*. Oxford University Press on Demand, 2004.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[51] Charles Hockett. The origin of speech. *Scientific American*, 203:88–111, 1960.

[52] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[53] Samar Husain, Shravan Vasishth, and Narayanan Srinivasan. Strong expectations cancel locality effects: Evidence from hindi. *PloS one*, 9(7):e100986, 2014.

[54] Ray Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, Oxford, UK, 2002.

[55] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

[56] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR (Poster)*. OpenReview.net, 2017.

[57] Jasmeen Kanwal, Kenny Smith, Jennifer Culbertson, and Simon Kirby. Zipf's law of abbreviation and the principle of least effort: Language users optimise a miniature lexicon for efficient communication. *Cognition*, 165:45–52, 2017.

[58] Charles Kemp and Terry Regier. Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054, 2012.

[59] Charles Kemp and Terry Regier. Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054, 2012.

[60] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *Proceedings of ICML*, virtual conference, 2020. In press.

[61] Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28:108–114, 2014.

[62] Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, 2015.

[63] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of EMNLP*, pages 2962–2967, Copenhagen, Denmark, 2017.

[64] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of ICML*, pages 2879–2888, Stockholm, Sweden, 2018.

[65] Brenden Lake, Tomer Ullman, Joshua Tenenbaum, and Samuel Gershman. Building machines that learn and think like people. *Behavorial and Brain Sciences*, 40:1–72, 2017.

[66] Ronald W Langacker. *Foundations of Cognitive Grammar: descriptive application. Volume 2*, volume 2. Stanford university press, 1987.

[67] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[68] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

[69] Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. Emergent translation in multi-agent communication. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[70] David Lewis. *Convention*. Harvard University Press, Cambridge, MA, 1969.

[71] Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. In *Proceedings of NeurIPS*, Vancouver, Canada, 2019. Published online: `https://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019`.

[72] Yaoyiran Li, Edoardo M Ponti, Ivan Vulić, and Anna Korhonen. Emergent communication pretraining for few-shot machine translation. *arXiv preprint arXiv:2011.00890*, 2020.

[73] Tal Linzen, Grzegorz Chrupała, and Afra Alishahi. Proceedings of the 2018 emnlp workshop blackboxnlp: Analyzing and interpreting neural networks for nlp. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

[74] Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes. Proceedings of the 2019 acl workshop blackboxnlp: Analyzing and interpreting neural networks for nlp. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

[75] Haitao Liu. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9:159–191, 09 2008.

[76] Scott E Lively, David B Pisoni, and Stephen D Goldinger. Spoken word recognition: Research and theory. 1994.

[77] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions

in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[78] Chris Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

[79] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[80] Kyle Mahowald, Isabelle Dautriche, Edward Gibson, and Steven Piantadosi. Word forms are structured for efficient use. *Cognitive Science*, 42:3116–3134, 2018.

[81] Gary Marcus. *The Algebraic Mind*. MIT Press, Cambridge, MA, 2003.

[82] Michael McCloskey. Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, 2(6):387–395, 1991.

[83] Alex Mesoudi and Andrew Whiten. The hierarchical transformation of event knowledge in human cultural transmission. *Journal of cognition and culture*, 4(1):1–24, 2004.

[84] George Miller. Some effects of intermittent silence. *American Journal of Psychology*, 70(2):311–314, 1957.

[85] George Peter Murdock. Social structure. 1949.

[86] Savithry Namboodiripad, Daniel Lenzen, Ryan Lepic, and Tessa Verhoef. Measuring conventionalization in the manual modality. *Journal of Language Evolutio*, 1(2):109–118, 2016.

[87] Martin Nowak and David Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999.

[88] Peter Pagin and Dag Westerståhl. Compositionality II: Arguments and problems. *Philosophy Compass*, 5(3):265–282, 2010.

[89] Bozena Pajak. Perceptual advantage from generalized linguistic knowledge. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

[90] Isabel Papadimitriou and Dan Jurafsky. Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, 2020.

[91] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[92] Steven T Piantadosi, Harry Tily, and Edward Gibson. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529, 2011.

[93] Steven Pinker, Paul Bloom, et al. Natural language and natural selection. *The adapted mind: Evolutionary psychology and the generation of culture*, pages 451–493, 1992.

[94] Terry Regier and Susanne Gahl. Learning the unlearnable: The role of missing evidence. *Cognition*, 93(2):147–155, 2004.

[95] Terry Regier, Paul Kay, and Naveen Khetarpal. Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104(4):1436–1441, 2007.

[96] Terry Regier, Charles Kemp, and Paul Kay. Word meanings across languages support efficient communication. *The handbook of language emergence*, 87:237, 2015.

[97] Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*, 2020.

[98] Eleanor Rosch. Principles of categorization. *Concepts: core readings*, 189, 1999.

[99] Kaius Sinnemäki. Complexity trade-offs in core argument marking. *Language complexity: Typology, contact, change*, 67:88, 2008.

[100] Brian Skyrms. *Signals: Evolution, learning, and information*. Oxford University Press, Oxford, UK, 2010.

[101] Luc Steels. Self-organizing vocabularies. In *Artificial Life V*, pages 179–184, 1997.

[102] Luc Steels. What triggers the emergence of grammar? In *Proceedings of EELC*, pages 143–150, Hatfield, UK, 2005.

[103] Luc Steels, editor. *Experiments in Cultural Language Evolution*. John Benjamins, Amsterdam, the Netherlands, 2012.

[104] Udo Strauss, Peter Grzybek, and Gabriel Altmann. Word length and word frequency. In *Contributions to the science of text and language*, pages 277–294. Springer, 2007.

[105] Ilya Sutskever, Oriol Vinyals, and Quoc Le. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112, Montreal, Canada, 2014.

[106] William J Teahan, Yingying Wen, Rodger McNab, and Ian H Witten. A compression-based algorithm for chinese word segmentation. *Computational Linguistics*, 26(3):375–393, 2000.

[107] David Temperley and Daniel Gildea. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, 4(1):67–80, 2018.

[108] Joshua B Tenenbaum and Thomas L Griffiths. Generalization, similarity, and bayesian inference. *Behavioral and brain sciences*, 24(4):629, 2001.

[109] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[110] Michael Tomasello. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA, 2003.

[111] David J Townsend, Thomas G Bever, et al. *Sentence comprehension: The integration of habits and rules*. MIT Press, 2001.

[112] Simon Townsend, Sabrina Engesser, Sabine Stoll, Klaus Zuberbühler, and Balthasar Bickel. Compositionality in animals and humans. *PLOS Biology*, 16(8):1–7, 2018.

[113] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008, Long Beach, CA, 2017.

[114] Tessa Verhoef, Simon Kirby, and Bart de Boer. Iconicity and the emergence of combinatorial structure in language. *Cognitive Science*, 40(8):1969–1994, 2016.

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

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

[117] Yongqin Xian, Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning: A comprehensive evaluation of the good, the bad and the ugly. `https://arxiv.org/abs/1707.00600`, 2017.

[118] Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942, 2018.

[119] George Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley, Boston, MA, 1949.

[120] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*, volume 21. Psychology Press, 1999.

*E*mergence of lan*G*uage in *G*ames

For most of the described works, we used the EGG toolkit. The latter provides reliable building blocks to experiment with different game variants (reconstruction vs. discrimination) and optimization procedures (Reinforce [116] vs. Gumbel Softmax [79, 56]) irrespective of NN agents' architectures.

To encourage multi-disciplinary research in the neural language evolution domain, we made the EGG toolkit open-source. This effort was mainly lead by Eugene Kharitonov and is joint work with Diane Bouchacourt, Marco Baroni, and myself.

This appendix includes EGG's documentation describing the main features design principles and providing a concrete game example detailing how it operates.

EGG: a toolkit for research on Emergence of lanGuage in Games

Eugene Kharitonov
Facebook AI
kharitonov@fb.com

Rahma Chaabouni
Facebook AI Research / LSCP
rchaabouni@fb.com

Diane Bouchacourt
Facebook AI
dianeb@fb.com

Marco Baroni
Facebook AI / ICREA
mbaroni@fb.com

Abstract

There is renewed interest in simulating language emergence among deep neural agents that communicate to jointly solve a task, spurred by the practical aim to develop language-enabled interactive AIs, as well as by theoretical questions about the evolution of human language. However, optimizing deep architectures connected by a discrete communication channel (such as that in which language emerges) is technically challenging. We introduce EGG, a toolkit that greatly simplifies the implementation of emergent-language communication games. EGG's modular design provides a set of building blocks that the user can combine to create new games, easily navigating the optimization and architecture space. We hope that the tool will lower the technical barrier, and encourage researchers from various backgrounds to do original work in this exciting area.

1 Introduction

Studying the languages that emerge when neural agents interact with each other recently became a vibrant area of research (Havrylov and Titov, 2017; Lazaridou et al., 2016, 2018; Kottur et al., 2017; Bouchacourt and Baroni, 2018; Lowe et al., 2019). Interest in this scenario is fueled by the hypothesis that the ability to interact through a human-like language is a prerequisite for genuine AI (Mikolov et al., 2016; Chevalier-Boisvert et al., 2019). Furthermore, such simulations might lead to a better understanding of both standard NLP models (Chaabouni et al., 2019b) and the evolution of human language itself (Kirby, 2002).

For all its promise, research in this domain is technically very challenging, due to the discrete nature of communication. The latter prevents the use of conventional optimization methods, requiring either Reinforcement Learning algorithms (e.g., REINFORCE; Williams 1992) or the Gumbel-Softmax relaxation (Maddison et al., 2016; Jang et al., 2016). The technical challenge might be particularly daunting for researchers whose expertise is not in machine learning, but in fields such as linguistics and cognitive science, that could contribute to this interdisciplinary research area.

To lower the starting barrier and encourage high-level research in this domain, we introduce the EGG (**E**mergence of lan**G**uage in **G**ames) toolkit. EGG aims at

1. Providing reliable building bricks for quick prototyping;
2. Serving as a library of pre-implemented games;
3. Providing tools for analyzing the emergent languages.

EGG is implemented in PyTorch (Paszke et al., 2017) and it is licensed under the MIT license. EGG can be installed from https://github.com/facebookresearch/EGG.

Notable features of EGG include: (a) Primitives for implementing single-symbol or variable-length communication (with vanilla RNNs (Elman, 1990), GRUs (Cho et al., 2014), LSTMs (Hochreiter and Schmidhuber, 1997));[1] (b) Training with optimization of the communication channel through REINFORCE or Gumbel-Softmax relaxation via a common interface; (c) Simplified configuration of the general components, such as check-pointing, optimization, Tensorboard support,[2] etc.; (d)

A screencast demonstration of EGG is available at https://vimeo.com/345470060

[1] EGG also provides an experimental support of Transformers (Vaswani et al., 2017).

[2] https://www.tensorflow.org/tensorboard

A simple CUDA-aware command-line tool for hyperparameter grid-search.

2 EGG's architecture

In the first iteration of EGG, we concentrate on a simple class of games, involving a single, unidirectional (Sender → Receiver) message. In turn, messages can be either single-symbol or multi-symbol variable-length sequences. Our motivation for starting with this setup is two-fold. First, it corresponds to classic signaling games (Lewis, 1969), it already covers a large portion of the literature (e.g., 5 out of 6 relevant studies mentioned in Introduction) and it allows exploring many interesting research questions. Second, it constitutes a natural first step for further development; in particular, the majority of components should remain useful in multi-directional, multi-step setups.

2.1 Design principles

As different training methods and architectures are used in the literature, our primary goal is to provide EGG users with the ability to easily navigate the space of common design choices.

Building up on this idea, EGG makes switching between Gumbel-Softmax relaxation-based and REINFORCE-based training effortless, through the simple choice of a different wrapper. Similarly, one can switch between one-symbol communication and variable-length messages with little changes in the code.[3]

We aim to maintain EGG minimalist and "hackable" by encapsulating the user-implemented agent architectures, the Reinforce/GS agent wrappers and the game logic into PyTorch modules. The user can easily replace any part.

Finally, since virtually any machine-learning experiment has common pieces, such as setting the random seeds, configuring the optimizer, model check-pointing, etc., EGG pre-implements many of them, reducing the necessary amount of boilerplate code to the minimum.

2.2 EGG design

EGG, in its first iteration, operates over the following entities. Firstly, there are two distinct agent roles: **Sender** and **Receiver**. Sender and Receiver are connected via a one-directional communication channel from the former to the latter, that has to produce the game-specific output.

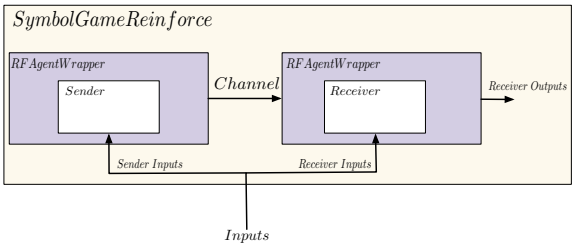


Figure 1: Example of EGG's game flow when using REINFORCE. White boxes (Sender and Receiver) represent the user-implemented agent architectures. The colored boxes are EGG-provided wrappers that implement a REINFORCE-based scenario. For example, *SymbolGameReinforce* is an instance of the Game block. It sets up single-symbol Sender/Receiver game optimized with REINFORCE. To use Gumbel-Softmax relaxation-based training instead, the user only has to change the EGG-provided wrappers.

The next crucial entity is **Game**. It encapsulates the agents and orchestrates the game scenario: feeding the data to the agents, transmitting the messages, and getting the output of Receiver. Figure 1 illustrates EGG's game flow in a specific example. Game applies a user-provided **loss** function, which might depend on the outputs of Receiver, the message transmitted, and the data. The value of the loss is minimized by a fourth entity, **Trainer**. Trainer also controls model checkpointing, early stopping, etc.

The Trainer and Game modules are pre-implemented in EGG. In a typical scenario, the communication method (single or multiple symbol messages) will be implemented by EGG-provided wrappers. As a result, what is left for the user to implement consists of: (a) the data stream, (b) core (non-communication-related) parts of the agents, (c) the loss. The data interface that is expected by Trainer is an instance of the standard PyTorch data loader `utils.data.DataLoader`.

To implement Sender, the user must define a module that consumes the data and outputs a tensor. On Receiver's side, the user has to implement a module that takes an input consisting of a message embedding and possibly further data, and generates Receiver's output.

Section 4 below provides examples of how to implement agents, choose communication and optimization type, and train a game.

[3] This also proved to be a convenient debugging mechanism, as single-symbol communication is typically simpler to train.

```
class Sender(nn.Module):
    def __init__(self, vision, output_size):
        super(Sender, self).__init__()
        self.fc = nn.Linear(500, output_size)
        self.vision = vision

    def forward(self, x):
        with torch.no_grad():
            x = self.vision(x)
        x = self.fc(x)
        return x


class Receiver(nn.Module):
    def __init__(self, input_size):
        super(Receiver, self).__init__()
        self.fc = nn.Linear(input_size, 784)

    def forward(self, channel_input, receiver_input=None):
        x = self.fc(channel_input)
        return torch.sigmoid(x)

sender = Sender(vision, output_size)
receiver = Receiver(input_size)
```

Figure 2: The MNIST game: Defining and instantiating the user-defined parts of the agents' architecture.

3 Optimizing the communication channel in EGG

EGG supports two widely adopted strategies for learning with a discrete channel, Gumbel-Softmax relaxation (used, e.g., by Havrylov and Titov (2017)) and REINFORCE (used, e.g., by Lazaridou et al. (2016)). Below, we briefly review both of them.

Gumbel-Softmax relaxation is based on the Gumbel-Softmax (GS) (aka Concrete) distribution (Maddison et al., 2016; Jang et al., 2016), that allows to approximate one-hot samples from a Categorical distribution. At the same time, GS admits reparametrization, hence allows backpropagation-based training. Suppose that Sender produces a distribution over the vocabulary, with ith symbol having probability $p_i = S(\boldsymbol{i}_s)$. To obtain a sample from a corresponding Gumbel-Softmax distribution, we take i.i.d. samples g_i from the `Gumbel(0, 1)` distribution and obtain the vector $\boldsymbol{y}$ with components y_i:

$$y_i = \frac{exp((\log p_i + g_i)/\tau)}{\sum_j exp((\log p_j + g_j)/\tau)} \quad (1)$$

where τ is the temperature hyperparameter, which controls the degree of relaxation. We treat $\boldsymbol{y}$ as a relaxed symbol representation. In the case of single-symbol communication, the embedding of $\boldsymbol{y}$ is passed to Receiver. In case of variable-length messages, the embedding is also fed into a RNN cell to generate the next symbol in the message.

As a result, if Receiver and the game loss are differentiable w.r.t. their inputs, we can get gradients of all game parameters, including those of Sender, via conventional backpropagation.

REINFORCE (Williams and Peng, 1991) is a standard Reinforcement Learning algorithm. Assume that both agents are stochastic: Sender samples a message m, and Receiver samples its output $\boldsymbol{o}$. Let us fix a pair of inputs, $\boldsymbol{i}_s$, $\boldsymbol{i}_r$, and the ground-truth output l. Then, using the log-gradient "trick", the gradient of the expectation of the loss L w.r.t. the vector of agents' parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_s \bigsqcup \boldsymbol{\theta}_r$ is:

$$\mathbb{E}_{m,\boldsymbol{o}}\left[L(\boldsymbol{o}, l)\nabla_\theta \log \mathbb{P}(m, \boldsymbol{o}|\boldsymbol{\theta})\right] \quad (2)$$

where $\mathbb{P}(m, \boldsymbol{o}|\boldsymbol{\theta})$ specifies the joint probability distribution over the agents' outputs.

The gradient estimate is found by sampling messages and outputs. A standard trick to reduce variance of the estimator in Eq. 2 is to subtract an action-independent baseline b from the optimized loss (Williams, 1992). EGG uses the running mean baseline.

Importantly, the estimator in Eq. 2 allows us to optimize agents even if the loss is not differentiable (e.g., 0/1 loss). However, if the loss is differentiable and Receiver is differentiable and deterministic, this can be leveraged by a "hybrid" approach: the gradient of Receiver's parameters can be found by backpropagation, while Sender is optimized with REINFORCE. This approach, a special case of gradient estimation using stochastic computation graphs as proposed by Schulman et al. (2015), is also supported in EGG.

4 Implementing a game

In this Section we walk through the main steps to build a communication game in EGG. We illustrate them through a MNIST (LeCun et al., 1998) communication-based autoencoding task: Sender observes an image and sends a message to Receiver. In turn, Receiver tries to reconstruct the image. We only cover here the core aspects of the implementation, ignoring standard pre- and post-processing steps, such as data loading. The full implementation can be found in an online tutorial.[4]

We start by implementing the agents' architectures, as shown in Figure 2. Sender is passed an input image to be processed by its pre-trained `vision` module, and returns its output after a linear transformation. The way Sender's output will be interpreted depends on the type of communication to be used (discussed below). Receiver gets

[4] https://colab.research.google.com/github/facebookresearch/EGG/blob/master/tutorials/EGG%20walkthrough%20with%20a%20MNIST%20autoencoder.ipynb

```
sender = core.GumbelSoftmaxWrapper(sender, temperature=1.0)

receiver = core.SymbolReceiverWrapper(receiver, vocab_size,
                                      agent_input_size=400)

game = core.SymbolGameGS(sender, receiver, loss)
```

(a) Single-symbol communication, Gumbel-Softmax relaxation.

```
sender = core.ReinforceWrapper(sender)

receiver = core.SymbolReceiverWrapper(receiver, vocab_size,
                                      agent_input_size=400)
receiver = core.ReinforceDeterministicWrapper(receiver)
game = core.SymbolGameReinforce(sender, receiver, loss, sender_entropy_coeff=0.05,
                                receiver_entropy_coeff=0.0)
```

(b) Single-symbol communication, REINFORCE.

```
sender_rnn = core.RnnSenderGS(sender, vocab_size, emb_size, hidden_size,
                              cell="rnn", max_len=2, temperature=1.0)
receiver_rnn = core.RnnReceiverGS(receiver, vocab_size, emb_size,
                                  hidden_size, cell="rnn")
game_rnn = core.SenderReceiverRnnReinforce(sender_rnn, receiver_rnn, loss,
                                           sender_entropy_coeff=0.025,
                                           receiver_entropy_coeff=0.0)
```

(c) Variable-length communication, Gumbel-Softmax relaxation.

```
sender_rnn = core.RnnSenderReinforce(sender, vocab_size, emb_size, hidden_size,
                                     cell="gru", max_len=2)
receiver_rnn = core.RnnReceiverDeterministic(receiver, vocab_size, emb_size,
                                             hidden_size, cell="gru")

game_rnn = core.SenderReceiverRnnGS(sender_rnn, receiver_rnn, loss)
```

(d) Variable-length communication, REINFORCE.

Figure 3: MNIST game: The user can choose different communication wrappers to switch between training regimes (Gumbel-Softmax or REINFORCE) and communication type (single-symbol or variable-length messages).

```
trainer = core.Trainer(game=game, optimizer=optimizer,
                       train_data=train_loader,
                       validation_data=test_loader,
                       epoch_callback=None)
trainer.train(n_epochs=15)
```

Figure 4: MNIST game: Once the agents and the game are instantiated, the user must pass them to a Trainer, which implements the training/validation loop, checkpointing, etc.

an input from Sender and returns an image-sized output with pixels valued in $[0; 1]$. Again, depending on the type of channel employed, the Receiver input will have a different semantics.

In the case of one-symbol communication, Sender's output is passed through a `softmax` layer and its output is interpreted as the probabilities of sending a particular symbol. Hence, the output dimensionality defines the size of the vocabulary. In the case of variable-length messages, Sender's output specifies the initial hidden state of an RNN cell. This cell is then "unrolled" to generate a message, until the end-of-sequence symbol (`eos`) is produced or maximum length is reached. Receiver's input is an embedding of the message: either the embedding of the single-symbol message, or the last hidden state of the RNN cell that corresponds to the `eos` symbol.

Once Sender and Receiver are defined, the user wraps them into EGG-implemented wrappers which determine the communication and optimization scenarios. Importantly, the actual user-specified Sender and Receiver architectures can be agnostic to whether single-symbol or variable-length communication is used; and to whether Gumbel-Softmax relaxation- or REINFORCE-based training is performed. In Figure 3 we illustrate different communication/training scenarios: (a) single-symbol communication, trained with Gumbel-Softmax relaxation, (b) single-symbol communication, trained with REINFORCE, (c) variable-length communication, trained with Gumbel-Softmax relaxation, (d) variable-length communication, trained with REINFORCE.

Once the Game instance is defined, everything is ready for training. That is, the user has to pass the game instance to `core.Trainer`, as shown in Figure 4.

We report some results obtained with the code we just described. We used the following parameters. The vision module is a pre-trained LeNet-1 (LeCun et al., 1990) instance, the maximal message length is set to 2, the communication between the agents is done through LSTM units with hidden-size 20, vocabulary size is 10. The agents are trained with REINFORCE for 15 epochs with batch size of 32, and the loss is per-pixel cross-entropy.

In Figure 5 we illustrate the language that emerges in this setup. To do this, we enumerate all possible 100 two-symbol messages x, y and input them to Receiver. We report all images that Receiver produces. The `eos` symbol is fixed to be 0, hence if the first symbol is 0 then the second symbol is ignored (top row of Figure 5).

Note that the first symbol x tends to denote digit identity: $x \in \{2, 4, 7, 8, 9\}$. In contrast, the second symbol y is either ignored ($x \in \{4, 8\}$) or specifies the style of the produced digit ($x \in \{3, 7\}$). The second symbol has the most striking effect with $x = 7$, where y encodes the rotation angle of the digit 1.

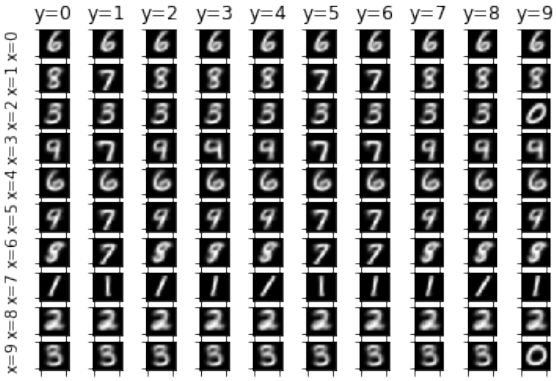


Figure 5: The emergent code-book in the MNIST autoencoder game. After training, we feed all 100 possible two-symbol messages xy from the size-10 vocabulary to Receiver and show the returned images. The rows iterate over the first symbol x, the columns enumerate the second symbol, y. The `eos` symbol has id 0.

5 Some pre-implemented games

EGG contains implementations of several games. They (a) illustrate how EGG can be used to explore interesting research questions, (b) provide reference usage patterns and building blocks, (c) serve as means to ensure reproducibility of studies reported in the literature. For example, EGG incorporates an implementation of the signaling game of Lazaridou et al. (2016) and Bouchacourt and Baroni (2018). It contains code that was recently used to study the communicative efficiency of artificial LSTM-based agents (Chaabouni et al., 2019a) and the information-minimization properties of emergent discrete codes (Kharitonov et al., 2019).[5] Finally, EGG provides a pre-implemented game that allows to train agents entirely via the command line and external input/output files, without having to write a single line of Python code. We hope this will lower the learning curve for those who want to experiment with language emergence without previous coding experience.

6 Conclusion and future work

We introduced EGG, a toolkit for research on emergence of language in games. We outlined its main features design principles. Next, we briefly reviewed how training with a discrete communication channel is performed. Finally, we walked through the main steps for implementing a MNIST autoencoding game using EGG.

We intend to extend EGG in the following directions. First, we want to provide support for multi-direction and multi-step communicative scenarios. Second, we want to add more advanced tooling for analyzing the properties of the emergent languages (such as compositionality; Andreas 2019). We will also continue to enlarge the set of pre-implemented games, to build a library of reference implementations.

[5] A small illustration can be run in Google Colaboratory: https://colab.research.google.com/github/facebookresearch/EGG/blob/master/egg/zoo/language_bottleneck/mnist-style-transfer-via-bottleneck.ipynb.

Acknowledgments

We are grateful to Roberto Dessì and Tomek Korbak for their contributions to the EGG codebase and to Serhii Havrylov for sharing his code with us.

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *ICLR*, New Orleans, LA.

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

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019a. Anti-efficient encoding in emergent communication. *arXiv preprint arXiv:1905.12561*.

Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. 2019b. Word-order biases in deep-agent emergent communication. In *ACL*.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. BabyAI: First steps towards grounded language learning with a human in the loop. In *International Conference on Learning Representations*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jeffrey Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. Information minimization in emergent languages. *arXiv preprint arXiv:1905.13687*.

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

Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge'naturally'in multi-agent dialog. *arXiv preprint arXiv:1706.08502*.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

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

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *NIPS*.

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

David Lewis. 1969. *Convention*. Harvard University Press, Cambridge, MA.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. 2019. On the pitfalls of measuring emergent communication. *arXiv preprint arXiv:1903.05168*.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Tomas Mikolov, Armand Joulin, and Marco Baroni. 2016. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 29–61. Springer.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. 2015. Gradient estimation using stochastic computation graphs. In *NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

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

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

Entropy Minimization in Emergent Languages

We found in **Chapter 4** of this manuscript that NNs develop an efficient and low-complex languages when communicating with a discrete communication channel. In this Appendix, we look only at the complexity of the emergent communication in a more controlled setup, where we can quantify the minimum needed complexity to solve the task. We found that NN minimize the complexity of their emergent languages within the range afforded by the need for successful communication. This minimization is amplified as we increase communication channel discreteness. Furthermore, the simpler languages are, the more robust NN agents are to over-fitting and adversarial attacks.

This study was mainly lead by my collaborator Eugene Kharitonov and is joint work with Diane Bouchacourt, Marco Baroni, and myself.

Entropy Minimization In Emergent Languages

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

Abstract

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

1. Introduction

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

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

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

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

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

2. General framework

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

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

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

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

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

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

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

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

3. Methodology

3.1. Games

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

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

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

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

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

a softmax activation returning the class probabilities.

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

3.2. Training with discrete channel

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

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

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

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

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

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

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

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

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

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

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

3.3. Experimental protocol

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

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

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

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

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

4. Experiments

4.1. Entropy minimization

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

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

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

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

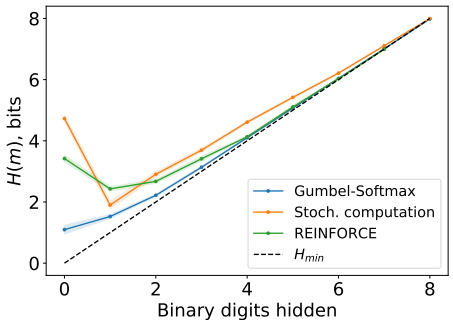
In Figures 1c & 1d, we report $H(\boldsymbol{m})$ when training with Stochastic Graph Optimization and REINFORCE across degrees of entropy regularization. We report curves corresponding to λ_s values which converged in more than three configurations. With REINFORCE, we see a weak tendency for a higher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

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

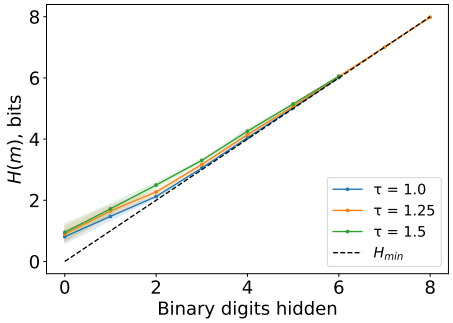
Summarizing, *when communicating through a discrete channel, there is consistent pressure for the emergent protocol to encode as little information as necessary*. This holds across games, training methods and hyperparameters. When training with Gumbel-Softmax, temperature controls the strength of this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

4.2. Evolution of message entropy during training

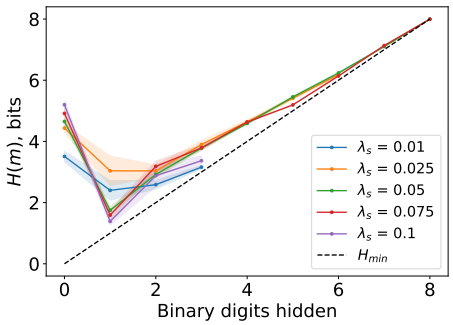
To gain further insights into the minimization trend, we studied the evolution of message entropy during training. We observed that the initial entropy of Sender can be both higher and lower than the minimum entropy H_{min} required for solving the task. Further, we measured how the entropy of the messages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



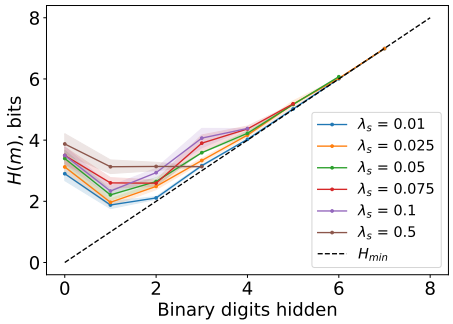
(a) All three training approaches.



(b) Training with Gumbel-Softmax relaxation.



(c) Training with Stochastic Computation Graph.



(d) Training with REINFORCE.

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

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

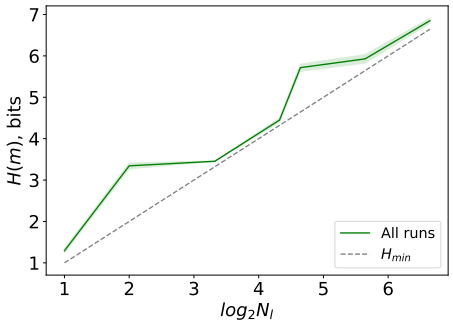
4.3. Representation discreteness and robustness

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

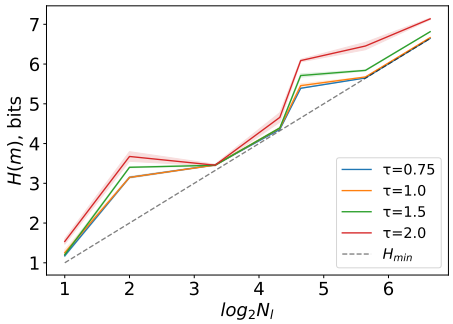
Further, we hypothesize that the emergent discrete bottleneck might have useful properties, since existing (continuous) architectures that explicitly impose a bottleneck pressure are more robust to overfitting (Fischer, 2019) and adversarial attacks (Alemi et al., 2016; Fischer, 2019). We test whether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

4.3.1. Robustness to over-fitting

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



(a) Successful runs pooled together.



(b) Successful runs grouped by temperature.

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

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

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

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

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

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

4.3.2. Robustness to adversarial examples

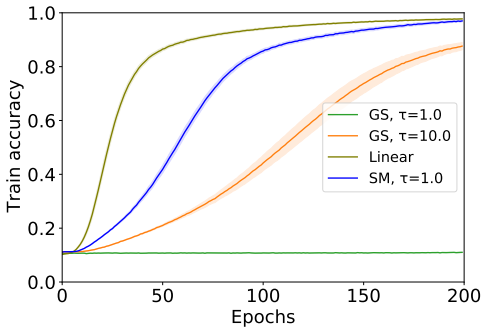
We study next robustness of agents equipped with a relaxed discrete channel against adversarial attacks. We use the same architectures as in the preceding experiment.

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

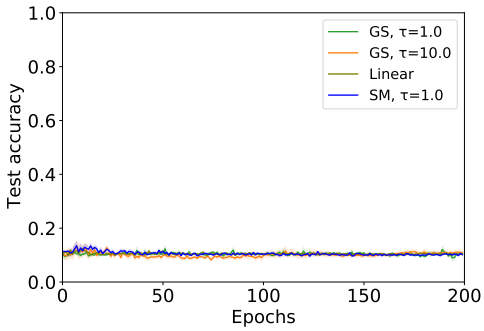
$$\boldsymbol{i}_s^* = clip\left[\boldsymbol{i}_s + \epsilon \cdot sign\left[\nabla_{\boldsymbol{i}_s}\mathcal{L}(\boldsymbol{o}, \boldsymbol{l})\right], 0, 1\right], \qquad (5)$$

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

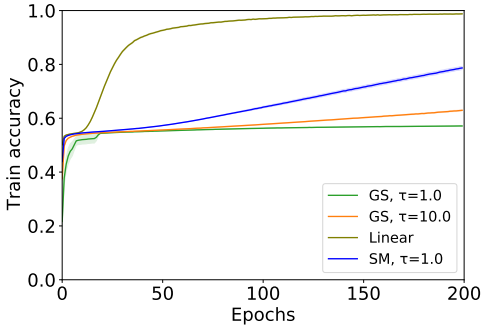
We experiment with two approaches of getting gradients for



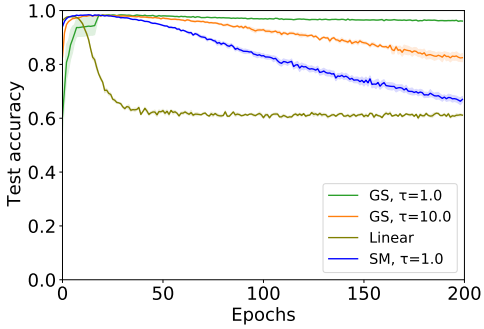
(a) All train labels are shuffled.



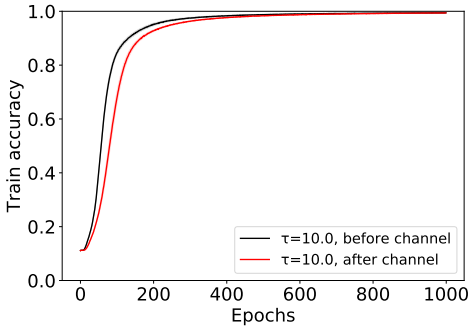
(b) All train labels are shuffled.



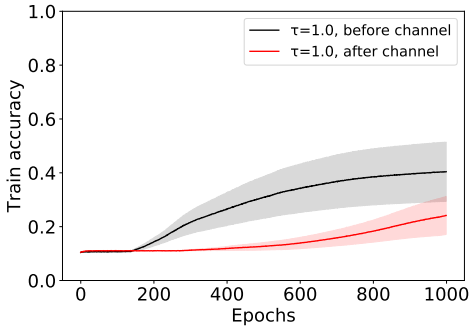
(c) Half of train labels are shuffled.



(d) Half of train labels are shuffled.

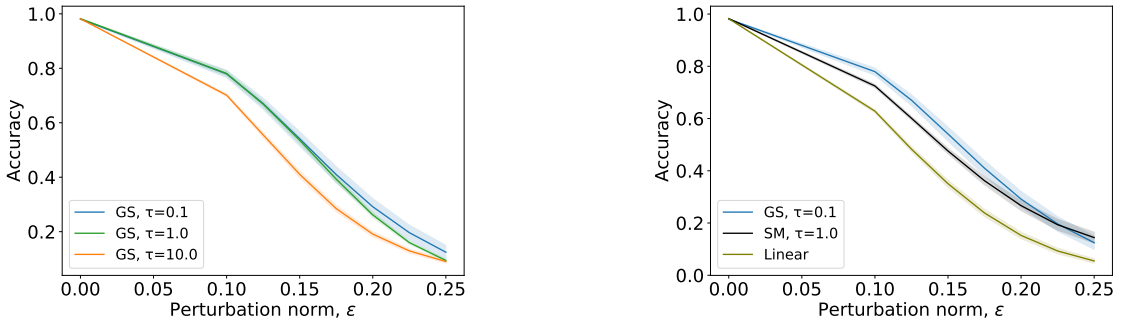


(e) All labels shuffled; Additional layer before channel vs. after channel



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

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



(a) Robustness vs. temp. τ.

(b) Comparison to the baselines.

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

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

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

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

5. Related Work

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

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

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

6. Discussion

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

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

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

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

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

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

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

References

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

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

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

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

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

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

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

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

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

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

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder

for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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