Categorial Grammar Induction as a Compositionality Measure for Understanding the Structure of Emergent Languages

Anonymous ACL submission

Abstract

This paper proposes a method for investigating the syntactic structure of emergent languages using categorial grammar induction. Although the structural property of emergent languages is an important topic, little has been done on syntax and its relation to semantics. Inspired by previous work on CCG induction for natural languages, we propose to induce categorial grammars from the sentence-meaning pairs of emergent languages. Since an emergent language born in a common environment called signaling game is represented as pairs of a message and a meaning, it is straightforward to extract sentence-meaning pairs to feed to categorial grammar induction. We also propose two compositionality measures that are based on the information obtained from induced grammars. Our experimental results reveal that our measures can recognize compositionality. While correlating with existing measure TopSim, our measures can gain more insights on the compositional structure of emergent languages from induced grammars.

1 Introduction

Communication among artificial agents born in an environment is called *emergent communication* and its protocols are *emergent languages* (Lazaridou and Baroni, 2020). Major motivations in this area are (1) to develop interactive AI (Foerster et al., 2016; Mordatch and Abbeel, 2018; Lazaridou et al., 2020), (2) to study language evolution (Kirby, 2001; Graesser et al., 2019; Dagan et al., 2021), and (3) to understand emergent languages or compare them with humans' (Kottur et al., 2017; Chaabouni et al., 2019a; Kharitonov et al., 2020). (1) and (2) are important from the engineering or scientific points of view. In fact, (3) is fundamental since the first two are not achievable without recognizing and filling the gap between emergent and human languages. Despite its importance, few methods have been established to evaluate the structure of emergent languages with respect to *syntax* and *semantics*. Previous work frequently employs a *signaling game* (Lewis, 1969) or its variant, where agents are either a function from *a meaning space* to *a message space* or its inverse. The problem is that little has been analyzed on how syntax combines messages to yield semantics or meanings. Such a structural property is known as *compositionality*.

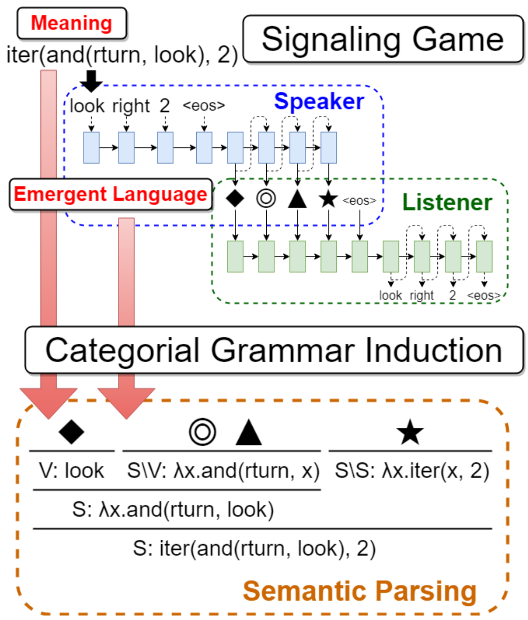

Figure 1: Illustration of a signaling game and categorial grammar induction (CGI). We first generate message-meaning pairs in the game, and then feed them to CGI.

To analyze the syntax of emergent languages, we focus on categorial grammar induction (CGI, e.g., Zettlemoyer and Collins, 2005) and propose to apply it to emergent languages. Figure 1 illustrates the relationship between a signaling game and CGI. Since CGI derives an explicit lexicon and a semantic parser given sentence-meaning pairs, it is suitable for the syntactic analysis of a language emerging as message-meaning pairs in a signaling game. We also propose compositionality measures built on the F1-score for unseen data and the lexicon size of CGI parsers. It is based on the intuition that a compositional language is expected to be generalized and described by a minimal lexicon.

Compositionality measures for emergent languages have been proposed, such as topographic similarity (TopSim, Brighton and Kirby, 2006), tree reconstruction error (TRE, Andreas, 2019), positional disentanglement (PosDis, Chaabouni et al., 2020), and bag-of-symbols disentanglement (BosDis, Chaabouni et al., 2020). We choose TopSim and TRE to compare with ours, since TopSim is most popular (e.g., Lazaridou et al., 2018) and TRE is similar to ours in the sense that it assumes structured meaning representations. Note that they do not consider the structure between a message and a meaning space, whereas our approach is aware of it with an explicit lexicon and a parser.

Pioneering and suggestive work by van der Wal et al. (2020) on the syntax of emergent languages proposes to apply unsupervised grammar induction (UGI) originally developed for natural languages: CCL (Seginer, 2007) and DIORA (Drozdov et al., 2019). UGI is reasonable if neither gold derivations nor meanings are available[1]. Note that UGI estimates the structure of emergent languages given only messages, whereas ours is intended to derive not only the structure but also the systematic composition of messages to meanings given message-meaning pairs.

Our contributions are (1) to propose to apply categorial grammar induction (CGI) to emergent languages for understanding their structure, (2) to propose two CGI-based compositionality measures that are more syntax-aware than existing compositionality measures, and (3) to show they can indeed measure compositionality.

2 Signaling Game in General

Most studies on emergent communication employ *Lewis signaling game* (Lewis, 1969) or its variant as an environment for agents to communicate. A signaling game contains a tuple (I, M, S, L), where I is an *input space*, M is a *message space*, a mapping $S : I \to M$ is a *speaker*, and a mapping $L : M \to I$ is a *listener*. The goal is $i = L(S(i))$ for a sampled input $i \in I$. Agents S, L are trained to achieve the goal given I, M. On the other hand, in a variant called *referential game* or *discrimination game*, a listener is defined as $L : M \times \mathcal{P}(I) \to I$, where $\mathcal{P}(I)$ is the power set of I. The goal is to distinguish i from other distractors: $i = L(S(i), C)$ for candidates $C \in \mathcal{P}(I)$ s.t. $i \in C$. An input space is typically a set of image data (Havrylov and Titov, 2017; Lazaridou et al., 2018; Bouchacourt and Baroni, 2018), sequential data (Li et al., 2020; Słowik et al., 2021), or attribute-value objects (Li and Bowling, 2019; Chaabouni et al., 2020; Ren et al., 2020). Besides, a message space is a set of discrete sequences in most studies.

[1] For example, if agents describe image data (e.g., Lazaridou et al., 2018), the meaning representations are unclear.

Agent Architecture Each agent is typically represented as a neural network, in particular, an encoder-decoder model. The speaker decoder and listener encoder are often recurrent neural networks. The speaker encoder can be a convolutional neural network, recurrent neural network, or perceptron, according to I. The listener decoder can be either the same as the speaker encoder or a classifier, depending on the goal.

Optimization Methods The speaker-listener pair is trained in an End-to-End manner, regarded as a single neural network. Previous work uses REINFORCE (Williams, 1992) and/or Gumbel-Softmax trick (Jang et al., 2017; Maddison et al., 2017), since the standard backpropagation is not applicable to discrete messages.

2.1 Existing Compositionality Measures

Compositionality is popular among those who are interested in the structural similarity between emergent and human languages. In the experiments, we compare our measures with TopSim (Brighton and Kirby, 2006) and TRE (Andreas, 2019). Let (I, M, S, L) be a signaling game defined above.

TopSim Let d_I, d_M be distance functions in I and M. TopSim is defined as Spearman correlation between $d_I(x, y)$ and $d_M(S(x), S(y))$ for all $(x, y) \in I \times I$. This score requires only d_I, d_M as structural information for I, M.

TRE The intuition of TRE is that if an emergent language is compositional, it should be approximated by another explicitly compositional function $f : I \to M$. Note that each $i \in I$ has to be *a binary tree* t in which a node n is binary node denoted as $n = (n', n'')$, unary node denoted as $n = (n')$, or a leaf node denoted as $n = l$. Besides, each $m \in M$ has to be a sequence of a fixed length k over a finite alphabet A. The calculation of TRE involves a distance δ and a composition κ_η with a trainable parameter $\eta = (V, W, \{E_l\}_l)$. δ is defined as L1 distance between $k \times |A|$ matrices. κ_η is defined as a mapping from a binary tree $t \in I$ to a $k \times |A|$ matrix:

$$\begin{aligned}\kappa_\eta(l) &= E_l \\ \kappa_\eta((n)) &= \kappa_\eta(n) \\ \kappa_\eta((n, n')) &= V\kappa_\eta(n) + W\kappa_\eta(n')\end{aligned}$$

where E_l is a $k \times |A|$ matrix for each leaf node l, and V, W are $k \times k$ matrices. Define $\texttt{one\_hot}(m)$ as a $k \times |A|$ matrix, the r-th row of which is the one-hot vector of the r-th symbol in $m \in M$. Then, TRE is computed with stochastic gradient descent as follows:

$$\mathrm{TRE} = \min_\eta \frac{1}{|I|} \sum_{i \in I} \delta(\kappa_\eta(i), \texttt{one\_hot}(S(i)))$$

Note that the lower TRE is, the higher compositionality is judged. TRE is similar to ours in the sense that inputs are assumed to be tree-structured.

3 Categorial Grammar Induction

In this section, we introduce categorial grammar (CG) and review its induction (CGI) for natural languages. CGI is also eligible for the analysis of emergent languages in signaling games, as it derives a lexicon and a parser from message-meaning pairs. Although previous work is on combinatory categorial grammar (CCG), we restrict it to CG[2].

3.1 Categorial Grammar

The formalism for our semantic parsing is *categorial grammar* (CG, Steedman, 1996, 2000). Context-free grammars are described largely with rules, whereas CGs are described largely with *lexical entries* and their rules are simple. A lexical entry $w \vdash X{:}\psi$ is a triple of a word w, a category X (defined below), and a logical form ψ. Consider the following example pair of a message and its logical form:

$$\text{“look left 1”}$$
$$\texttt{iter}(\texttt{and}(\texttt{lturn}, \texttt{look}), \texttt{1})$$

Their lexical entries can be described as follows:

$$\begin{aligned}\text{look} &\vdash \texttt{V} : \texttt{look} \\ \text{left} &\vdash \texttt{S}\backslash\texttt{V} : \lambda x.\texttt{and}(\texttt{lturn}, x) \\ \text{1} &\vdash \texttt{S}\backslash\texttt{S} : \lambda x.\texttt{iter}(x, \texttt{1})\end{aligned}$$

Symbols like $\texttt{V}$, $\texttt{S}\backslash\texttt{V}$, and $\texttt{S}\backslash\texttt{S}$ represent syntactic types or *categories*. A category is either an atomic category of the form $\texttt{N}$, $\texttt{V}$, or $\texttt{S}$, or a complex category of the form X/Y or $X\backslash Y$ where X, Y are categories. The atomic categories $\texttt{N}$, $\texttt{V}$, and $\texttt{S}$ stand for the linguistic notions of noun, intransitive verb, and sentence respectively[3].

In addition, CGs have *application rules* to describe the way to combine adjacent categories.

Application rules (with semantics):

$$\begin{array}{llcll} X/Y : f & Y : a & \Rightarrow & X : f(a) & (>) \\ Y : a & X\backslash Y : f & \Rightarrow & X : f(a) & (<) \end{array}$$

where X, Y are categories. The first rule named "$>$" is called the *forward application rule*, while the second rule named "$<$" is called the *backward application rule*. Rule $>$ (resp. $<$) means that a predicate f of category X/Y (resp. $X\backslash Y$) can take an argument a of category Y to yield $f(a)$ of category X.

With the lexical entries and the application rules, we can construct a derivation tree of "look left 1" as follows:

$$\dfrac{\dfrac{\dfrac{\text{look}}{\texttt{V} : \texttt{look}} \quad \dfrac{\text{left}}{\texttt{S}\backslash\texttt{V} : \lambda x.\texttt{and}(\texttt{lturn}, x)}}{\texttt{S} : \texttt{and}(\texttt{lturn}, \texttt{look})}< \quad \dfrac{\text{1}}{\texttt{S}\backslash\texttt{S} : \lambda x.\texttt{iter}(x, \texttt{1})}}{\texttt{S} : \texttt{iter}(\texttt{and}(\texttt{lturn}, \texttt{look}), \texttt{1})}<$$

3.2 Log-linear Probabilistic CGs

Given a lexicon Λ, a set of lexical entries, there might be multiple derivations for each message. Following previous work on CG induction (e.g., Zettlemoyer and Collins, 2005), we choose the most likely derivation by using a log-linear model, which contains a feature vector ϕ and a parameter vector θ. Given a message m, the joint probability of a logical form ψ and a derivation τ is defined as:

$$P(\tau, \psi \mid m; \theta, \Lambda) = \frac{e^{\theta \cdot \phi(m, \tau, \psi)}}{\sum_{(\tau', \psi')} e^{\theta \cdot \phi(m, \tau', \psi')}}.$$

[2] One might wonder why we do not use CCG. This is because the input spaces for our signaling games are described by context-free grammars, whose expressive power is known to be equal to that of CG. Nevertheless, it is interesting to speculate whether emergent languages can have complex rules like composition or type-raising. It is left for future work.

[3] The category of intransitive verbs is usually $\texttt{S}/\texttt{N}$ ($\texttt{S}/\texttt{NP}$) or $\texttt{S}\backslash\texttt{N}$ ($\texttt{S}\backslash\texttt{NP}$), but we regard $\texttt{V}$ as a atomic category. This is because the languages and logical forms we define in Section 5.1 take an imperative form without any subject.

Then, the parsing problem is to find the most likely logical form $\hat{\psi}$ given m:

$$\begin{aligned}\hat{\psi} &= \arg\max_{\psi} p(\psi \mid m; \theta, \Lambda) \\ &= \arg\max_{\psi} \sum_{\tau} P(\tau, \psi \mid m; \theta, \Lambda).\end{aligned}$$

3.3 CG Induction Algorithm

Algorithm 1 Common Structure of CG Induction

Input: A dataset $\mathcal{E} = \{(m^j, \psi^j)\}_{j=1}^{N}$ of message-meaning pairs, a seed lexicon Λ_{seed}, the number of iterations T, and a learning rate γ.
Output: Lexicon Λ and parameter vector θ

$\Lambda_0 \leftarrow \textsc{InitLex}(\mathcal{E}, \Lambda_{\text{seed}})$
$\theta_0 \leftarrow \textsc{InitParam}(\mathcal{E}, \Lambda_{\text{seed}})$
▷ Step 0: Initialize lexicon and parameter
for $t \in \{1, \ldots, T\}$ **do**
  $\Lambda_t^+ \leftarrow \textsc{UpdateLex}(\mathcal{E}, \theta_{t-1}, \Lambda_{t-1}, \Lambda_0)$
  ▷ Step 1: Update Lexicon
  $\theta_t \leftarrow \textsc{UpdateParam}(\mathcal{E}, \theta_{t-1}, \Lambda_t^+, \gamma)$
  ▷ Step 2: Update Parameter
  $\Lambda_t \leftarrow \textsc{PruneLex}(\mathcal{E}, \theta_{t-1}, \Lambda_t^+)$
  ▷ Step 3: Prune Lexicon (optional)
end for
return Λ_T and θ_T

Several CG induction (CGI) algorithms have been proposed. Algorithm 1 shows their common structure as a pseudo code. Generally, the inputs to CGI are a training data $\mathcal{E} = \{(m^j, \psi^j)\}_{j=1}^{N}$ of message-meaning pairs, a seed lexicon Λ_{seed}, the number of iterations T, and a learning rate γ, while the outputs are a lexicon Λ and a parameter θ. CGI involves four procedures: (1) lexicon and parameter initialization (InitLex, InitParam) that helps learning in early iterations, (2) lexicon update (UpdateLex) that introduces a new potential lexicon, (3) parameter update (UpdateParam) with gradient descent, and optionally (4) lexicon pruning (PruneLex) that discards a lexicon no longer in use. ZC05 (Zettlemoyer and Collins, 2005) is the first paper formalizing CGI. ZC07 (Zettlemoyer and Collins, 2007) is its improved version. In ZC05/07, InitLex is simply $\Lambda_0 = \Lambda_{\text{seed}}$ and UpdateLex relies on hand-crafted templates to add a new lexicon. KZGS10/11 (Kwiatkowski et al., 2010, 2011) modified UpdateLex so that it can create a new lexicon by automatically merging and splitting the existing entries in use. In KZGS10/11, InitLex returns $\mathcal{E}$ themselves with category $\mathtt{S}$ in addition to Λ_{seed}:

$$\Lambda_0 \leftarrow \Lambda_{\text{seed}} \cup \{m^j \vdash \mathtt{S} : \psi^j \mid j = 1, \ldots, N\}$$

Then, the lexical entries are split or merged during the iteration, seeking an appropriate segmentation. A problem in KZGS10/11 is that the lexicon size increases monotonically over iterations. ADP14 (Artzi et al., 2014) addressed this issue by adding a lexicon pruning process (PruneLex), which discards the lexical entries no longer in use[4].

4 CGI as a Compositionality Measure

We propose two compositionality measures CGF and CGL, which are based on an induced categorial grammar. Let $\mathcal{E}_{\text{train}}, \mathcal{E}_{\text{test}}$ be a training and test data for CGI. We train a log-linear model with $\mathcal{E}_{\text{train}}$ to derive a lexicon Λ and a parameter θ and test it with $\mathcal{E}_{\text{test}}$ to calculate the F1-score for semantic parsing:

$$\begin{aligned}\text{F1-score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ \text{precision} &= \frac{\#\,\text{correctly parsed}}{\#\,\text{parsed}} \\ \text{recall} &= \frac{\#\,\text{correctly parsed}}{|\mathcal{E}_{\text{test}}|}\end{aligned}$$

following previous work (Zettlemoyer and Collins, 2005) [5]. Then, CGF and CGL are defined as:

$$\text{CGF} = \text{F1-score}, \quad \text{CGL} = |\Lambda|$$

Note that the higher CGF (resp. lower CGL) is, the more compositional a language is judged, since a compositional language should be generalized and described by a minimal lexicon.

4.1 Difference from Existing Measures

Although existing compositionality measures such as TopSim and TRE are also mappings from message-meaning pairs to a real number, neither they clarify the structure of a message space M nor they derive any compositional function from M to an input space I.

Remember that TopSim only involves distance functions d_I, d_M, the choice of which is left to humans, and it does not clarify the structure of M. On the other hand, our approach can derive the structure of M by deriving a lexicon. TRE induces a composition $\kappa_\eta : I \to M$, but not the inverse. As Andreas (2019) is aware, it causes a language with identical messages for all meanings to be judged

[4] ADP14 also has improvements in UpdateLex, but we do not go into them in this paper.

[5] If Λ does not have sufficient lexical entries, the model fails to parse messages regardless of correctness.

compositional, contrary to our intuition. Again, ours would not regard it as compositional since a CGI parser is a function $M \to I$.

Therefore, what differentiates us from the existing measures is that our approach can derive an explicit lexicon and a semantic parser, whereas the existing measures cannot [6].

5 Experimental Setup

This section introduces a signaling game, optimization method, CGI algorithm, and evaluation metrics specific to our experiments. The overall experimental procedure is as follows:

1. Split an input space I in half: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}$.
2. Train a speaker S and a listener L with $\mathcal{D}_{\text{train}}$. Validate and test them with $\mathcal{D}_{\text{test}}$.
3. Given trained S, make datasets for CGI by pairing each message with its logical form:
$$\mathcal{E}_x = \{(S(i), \langle i \rangle) \mid i \in \mathcal{D}_x\}$$
where $x \in \{\text{train}, \text{test}\}$ and $\langle i \rangle$ is the logical form of i, to which CGI is applicable.
4. Train a CG parser with $\mathcal{E}_{\text{train}}$, test it with $\mathcal{E}_{\text{test}}$, and calculate CGF with $\mathcal{E}_{\text{test}}$ and CGF with a derived lexicon Λ.
5. Calculate TopSim and TRE with $\mathcal{D}_{\text{train}}$.

5.1 Input Space for Signaling Game

We define two input spaces for our signaling game: *Lang-attval* and *Lang-conj* [7]. Lang-attval is the same as attribute-value inputs in previous work (e.g., Kottur et al., 2017), while Lang-conj is more complex. Moreover, we define logical forms for each input.

Lang-attval *Lang-attval* is defined as the set of sequences derived from the following context-free grammar with a start symbol S:

$$\begin{aligned}
\mathrm{S} &\to \mathrm{V}'\ \mathrm{R} \\
\mathrm{V}' &\to \mathrm{V}\ \ \mathrm{D} \\
\mathrm{V} &\to \text{look} \mid \text{jump} \mid \text{walk} \mid \text{run} \\
\mathrm{D} &\to \text{left} \mid \text{right} \mid \text{up} \mid \text{down} \\
\mathrm{R} &\to 1 \mid 2 \mid 3 \mid 4
\end{aligned}$$

Lang-attval is regarded as attribute-value objects (Kottur et al., 2017; Andreas, 2019; Li and Bowling, 2019; Ren et al., 2020). In our case, attributes are *verb*, *direction*, and *repetition*, each of which has 4 values (e.g., *look*, *jump*, *walk*, and *run* for *verb*).

Lang-conj Let S'' be a start symbol. Then, *Lang-conj* is the set of sequences derived from the above context-free grammar in addition to the following rules:

$$\begin{aligned}
\mathrm{S}'' &\to \mathrm{S} \mid \mathrm{S}\ \mathrm{S}' \\
\mathrm{S}' &\to \text{and}\ \mathrm{S}
\end{aligned}$$

Each element in Lang-conj is either an element in Lang-attval or a conjunction of two elements in Lang-attval.

Logical Form We define the logical form of each element in Lang-attval/conj, to which CGI is simply applicable. We temporarily denote elements *parenthetically* to clarify their derivation trees (e.g., "$\mathrm{S}(\mathrm{V}'(\mathrm{V}(\text{jump}), \mathrm{D}(\text{left})), \mathrm{R}(2))$" for "jump left 2"). Then, the logical form $\langle i \rangle$ of a derivation i is defined inductively as follows:

$$\begin{aligned}
\langle \mathrm{S}''(\mathrm{S}(x)) \rangle &= \langle \mathrm{S}(x) \rangle \\
\langle \mathrm{S}''(\mathrm{S}(x), \mathrm{S}'(y)) \rangle &= \texttt{and}(\langle \mathrm{S}(x) \rangle, \langle \mathrm{S}'(y) \rangle) \\
\langle \mathrm{S}'(\text{and}, \mathrm{S}(x)) \rangle &= \langle \mathrm{S}(x) \rangle \\
\langle \mathrm{S}(\mathrm{V}'(x), \mathrm{R}(y)) \rangle &= \texttt{iter}(\langle \mathrm{V}'(x) \rangle, \langle \mathrm{R}(y) \rangle) \\
\langle \mathrm{V}'(\mathrm{V}(x), \mathrm{D}(y)) \rangle &= \texttt{and}(\langle \mathrm{D}(y) \rangle, \langle \mathrm{V}(x) \rangle) \\
\langle X(x) \rangle &= \langle x \rangle \ \ (X \in \{\mathrm{V}, \mathrm{D}, \mathrm{R}\}),
\end{aligned}$$

and for terminal symbols, $\langle \text{look} \rangle = \texttt{look}$, $\langle \text{left} \rangle = \texttt{lturn}$, $\langle 1 \rangle = \texttt{1}$, and so forth.

Examples Here are some examples:

$$\begin{aligned}
i &= \text{"jump left 2"} \in \text{Lang-attval} \cap \text{Lang-conj} \\
\langle i \rangle &= \texttt{iter}(\texttt{and}(\texttt{lturn}, \texttt{jump}), \texttt{2}).
\end{aligned}$$

Also,

$$\begin{aligned}
i' &= \text{"jump left 2 and walk up 3"} \in \text{Lang-conj} \\
\langle i' \rangle &= \texttt{and}(\texttt{iter}(\texttt{and}(\texttt{lturn}, \texttt{jump}), \texttt{2}), \\
&\qquad \texttt{iter}(\texttt{and}(\texttt{uturn}, \texttt{walk}), \texttt{3})).
\end{aligned}$$

5.2 Signaling Game for Sequential Data

Agent architectures and game procedure have to be adapted to the sequential inputs defined above. Hence, our signaling game takes a sequence-to-sequence-to-sequence procedure.

[6] TopSim and TRE are still reasonable if our purpose is to distinguish partially (but insufficiently) compositional languages from the ones not compositional at all.

[7] They are inspired by the commands of Chaabouni et al. (2019b) or SCAN (Lake and Baroni, 2018).

Architecture Speaker and listener agents are represented as a seq2seq model based on single-layer LSTMs (Hochreiter and Schmidhuber, 1997) with standard attention mechanisms (Dong and Lapata, 2016), similarly to Chaabouni et al. (2019b).

Game Procedure A *sequential signaling game* consists of a tuple $(I, A, k, \texttt{eos}, S, L)$, where I is an input space, A is a finite alphabet s.t. $\texttt{eos} \notin A$, k is a message length, and eos is a special symbol for end-of-sentence. A message space M is defined as the set of sequences of length k over A, $S : I \to M$ is a speaker, and $L : M \to I$ is a listener. Note that $x + \texttt{eos}$ denotes *a sequence x attached with* eos. The goal of the game is to minimize

$$\Delta(i + \texttt{eos}, L(S(i + \texttt{eos}) + \texttt{eos}))$$

for a uniformly sampled $i \in I$, where Δ is the humming distance.

5.3 Optimization for Agents

As Δ is indifferentiable, we use REINFORCE (Williams, 1992), which gives the following differentiable loss:

$$\begin{aligned}
&\mathbb{E}[\{\Delta(i + \texttt{eos}, o) - b\} \log P_S(m|i + \texttt{eos})] \\
+&\mathbb{E}[\{\Delta(i + \texttt{eos}, o) - b\} \log P_L(o|m + \texttt{eos})] \\
+&\mathbb{E}[\lambda_S \mathcal{H}(P_S) + \lambda_L \mathcal{H}(P_L)]
\end{aligned}$$

where P_S (resp. P_L) is the output distribution of speaker (resp. listener) over a message m (resp. output o) given an input i (resp. message m), b is a mean baseline, $\mathcal{H}$ denotes entropy, and λ_S, λ_L are nonnegative hyper-parameters. The last term is an entropy regularizer (Williams and Peng, 1991).

5.4 CGI for Emergent Languages

We apply CGI to emergent languages. As there is no prior knowledge on them, CGI should avoid ad hoc methods, considering the following:

(1) *Features in a log-linear model have to be as simple as possible.*

(2) *Lexical entries have to be generated automatically without any manual templates.*

(3) *Lexicon size has to be minimal*; otherwise it is hard to interpret results, e.g., to measure compositionality with CGL.

There is no existing method satisfying all of them simultaneously. We combine three methods. For (1), we follow ZC05 (Zettlemoyer and Collins, 2005): each feature is the count of times that each lexical entry is used in a derivation. However, ZC05 generates lexical entries with manual templates, contrary to (2). Instead, we follow KZGS10 (Kwiatkowski et al., 2010) that creates a new lexicon by merging and splitting the existing entries in use. The problem in KZGS10 is that the lexicon size increases monotonically during iterations, which is against (3). Thus, we follow ADP14 (Artzi et al., 2014) to discard the entries no longer in use. Other modifications are detailed in Appendix A.

5.5 Other Languages for Comparison

To evaluate the effectiveness of our measures, we need more and less compositional languages as well as emergent languages to apply CGI. To this end, we use Lang-attval/conj and *AdjSwap-x* ($x \in \{1, 2\}$). AdjSwap-x is made by applying x-times random adjacent swaps to each message in emergent languages. As Lang-attval and Lang-conj are fully compositional by definition, they should be judged more compositional than emergent languages. On the other hand, AdjSwap-x should be judged less compositional. van der Wal et al. (2020) adopted three languages for the same purpose: fully-structured, random, and shuffled emergent languages. The fully-structured corresponds to Lang-attval/conj in our case. We use AdjSwap-x as instances of less-compositional languages rather than random and shuffled emergent languages. This is because preliminary experiments revealed that CGI totally fails for these languages (see Appendix C). While this is an expected behavior, we additionally employ AdjSwap-x as a language supposed to be more compositional than random and shuffled emergent languages, for obtaining more insights.

5.6 Evaluation Metrics for Compositionality

We use CGF/L as well as TopSim and TRE. When clarifying the target language, we write the metrics as (*measure*)-(*language*), e.g., TopSim-Emergent, CGF-AdjSwap-1, and CGL-Lang-attval.

6 Experiments

We show the experimental results in this section. Let $(I, A, k, \texttt{eos}, S, L)$ be a sequential signaling game as defined in Section 5.2.

For (hyper-)parameter settings, see Appendix B.

$$\dfrac{\dfrac{1,1,1}{\mathtt{S/S} : \lambda x.\mathtt{and}(x, \mathtt{iter}(\mathtt{and}(\mathtt{rturn},\mathtt{walk}),2))} \quad \dfrac{\dfrac{16,13}{\mathtt{V}:\mathtt{run}} \quad \dfrac{25,1,1}{\mathtt{S\backslash V}: \lambda x.\mathtt{iter}(\mathtt{and}(\mathtt{rturn},x),3)}}{\mathtt{S}:\mathtt{iter}(\mathtt{and}(\mathtt{rturn},\mathtt{run}),3)}<}{\mathtt{S}: \mathtt{and}(\mathtt{iter}(\mathtt{and}(\mathtt{rturn},\mathtt{run}),3), \mathtt{iter}(\mathtt{and}(\mathtt{rturn},\mathtt{walk}),2))}>$$

Figure 2: Example correct derivation tree of a message $1, 1, 1, 16, 13, 25, 1, 1$ when $(I, k, |A|) = (\text{Lang-conj}, 8, 31)$.

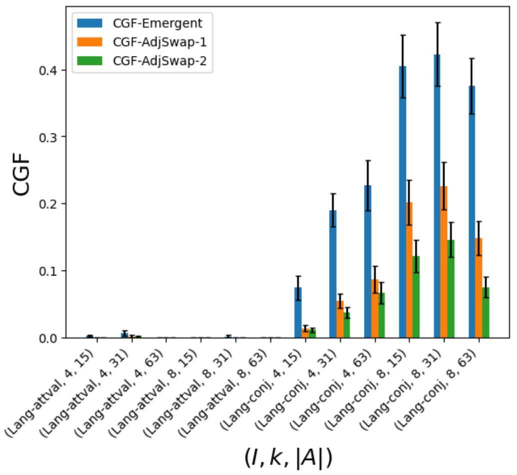

Figure 3: CGF plotted under various $(I, k, |A|)$. The error bars represent one standard error of mean.

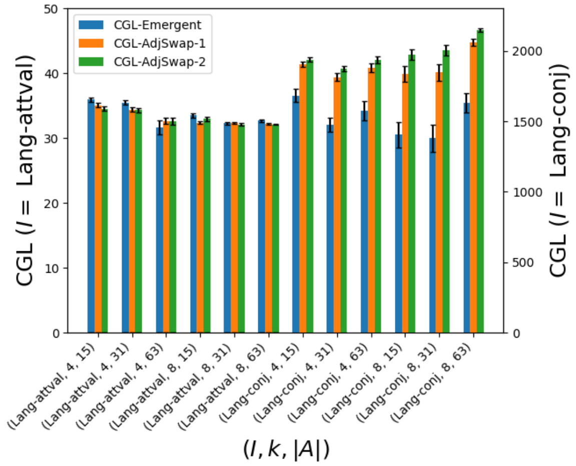

Figure 4: CGL plotted under various $(I, k, |A|)$. The error bars represent one standard error of mean.

6.1 Compositionality of Emergent Languages

We investigate whether CGF/L works as a measure of compositionality. If CGF works, the following inequality should hold: CGF-Lang-attval/conj $>$ CGF-Emergent $>$ CGF-AdjSwap-1 $>$ CGF-AdjSwap-2. Likewise, if CGL works, CGL-Lang-attval/conj $<$ CGL-Emergent $<$ CGL-AdjSwap-1 $<$ CGL-AdjSwap-2. First, we report that CGF-Lang-attval is 0.984 (± 0.0463), CGL-Lang-attval is 12.3 (± 0.852), CGF-Lang-conj is 0.868 (± 0.1173), and CGL-Lang-conj is 23.8 (± 17.59), where ($\pm$_) denotes a standard error of mean [8]. For the rest, Figure 3 (resp. Figure 4) shows CGF (resp. CGL) under various $(I, k, |A|)$.

For I = Lang-attval, Figure 3 shows surprisingly that CGI fails: CGF-Emergent is near or equal to 0. Besides, CGL-Emergent and CGL-AdjSwap-x in Figure 4 do not show clear differences. Hence, neither CGF nor CGL does not recognize the compositionality of emergent languages. CGF is almost 0 (Figure 3) and CGL concentrates around the size of training data 32 (Figure 4), which means the models overfit the training data. There are two possible reasons for it: emergent languages are not compositional or the training data for CGI is insufficient. We suppose the former is true since CGF-Lang-attval is near perfect (0.984) and CGL-Lang-attval is almost minimal (12.3) with the same size of training data.

For I = Lang-conj, Figure 3 reveals that CGF exactly shows the order of compositionality as expected: CGF-Lang-conj $>$ CGF-Emergent $>$ CGF-AdjSwap-1 $>$ CGF-AdjSwap-2. Likewise, CGL in Figure 4 shows the expected order: CGL-Lang-conj $<$ CGL-Emergent $<$ CGL-AdjSwap-1 $<$ CGL-AdjSwap-2. Hence, CGF and CGL recognize the compositionality of emergent languages. Nevertheless, CGF-Emergent is less than half of CGF-Lang-conj and CGL-Emergent is over 50 times larger than CGL-Lang-conj. It suggests that emergent languages are not fully compositional.

6.2 Comparison with Existing Measures

Next, we check the relationships among CGF/L, TopSim, and TRE. We show the results for I = Lang-conj, where CGF/L recognizes the compositionality of emergent languages. Figure 5 shows the scatter plot of TopSim and CGF. It shows a correlation with Pearson $\rho = 0.644$ ($p = 8.77 \times 10^{-24} \ll 0.01$). We also note that TopSim and CGL show a correlation with Pearson $\rho = -0.689$

[8]We train models 32 times for Lang-attval and Lang-conj respectively.

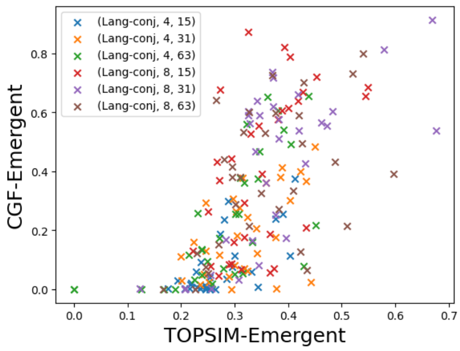

Figure 5: Scatter plot of CGF-Emergent and TopSim-Emergent, when $I=$ Lang-conj. Pearson correlation is $\rho = 0.644$ ($p = 8.77 \times 10^{-24} \ll 0.01$).

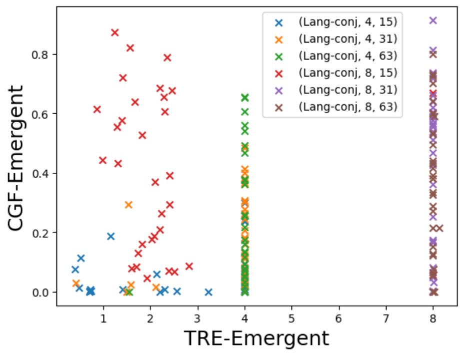

Figure 6: Scatter plot of CGF-Emergent and TRE-Emergent, when $I=$ Lang-conj. Unnatural concentration around $k \in \{4, 8\}$ is observed.

($p = 2.88 \times 10^{-28} \ll 0.01$). Although p-values are considerably small, ρs are moderate. Besides, Figure 5 shows several data points with high TopSim but low CGF. It suggests that TopSim tends to judge partially compositional languages more compositional than CGF.

Figure 6 shows the scatter plot of TRE and CGF. Astonishingly, it shows no correlation because of the unnatural concentration of TRE around $k \in \{4, 8\}$ if $|A| \in \{31, 63\}$. It means that a composition κ_η fails to learn so that its outputs are trapped between 0 and $1/|A|$. We speculate that the definition of κ_η or δ in Section 2.1 should have involved any nonlinear function. The scatter plots for CGLs are listed in Appendix D.

6.3 Example Derivation Tree of Emergent Language

Finally, Figure 2 exemplifies a derivation tree in an emergent language that CGI judges highly compositional (CGF $= 0.914$, CGL $= 423$). We can see how the message is combined to yield the meaning, which is a striking feature of CGI that the existing compositionality measures do not have. In this example, 16,13 means "run," 25,1,1 means "__ right 3," and 1,1,1 means "__ and walk right 2." Interestingly, it suggests message and meaning segmentation does not necessarily match the intuitive segmentation as shown in Section 3.1.

7 Discussion

The experimental results show that CGF and CGL work as a compositionality measure for emergent languages. Note that the observations on Lang-conj are consistent with those of van der Wal et al. (2020) in a sense that fully structured languages are judged the most syntactical, the emergent languages are judged the second, and lower structured broken languages are the least. However, neither CGF nor CGL recognizes the compositionality when an input space is a small set of attribute-value objects. It casts doubt on attribute-value settings for studying structural similarities between emergent and human languages. We found a moderate correlation between CGF/L and TopSim which suggests that CGI is not as sensitive to partial compositionality as TopSim. On the other hand, TRE does not work if the alphabet size is too large, probably due to the choice of δ or κ_η in Section 2.1. Finally, we can directly observe the systematic composition of a message to a meaning, which is a salient feature of CGI that previous work does not have. We hope that it brings deeper insights on the syntax and semantics of emergent languages.

8 Conclusion

This paper introduces categorial grammar induction (CGI) as a new compositionality measure for the structure of emergent languages. We proposed to apply CGI to emergent languages and define two compositionality measures CGF and CGL. Our experiments revealed that CGF/L can measure compositionality as we expected. Unlike existing measures, our approach meets compositionality in a traditional sense, allowing us to analyze emergent languages with a lexicon and derivation trees. For future work, it would be interesting to study the structure of the derivations of emergent languages. Besides, we speculate that *situated CCGs* (Artzi and Zettlemoyer, 2013) are applicable, which induce CGs considering an external world. Hence, CGI may be applicable to visual referential games as well as 2D-grid world communication.

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yoav Artzi, Dipanjan Das, and Slav Petrov. 2014. Learning compact lexicons for CCG semantic parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1273–1283. ACL.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Trans. Assoc. Comput. Linguistics*, 1:49–62.

Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 981–985. Association for Computational Linguistics.

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artif. Life*, 12(2):229–242.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4427–4442. Association for Computational Linguistics.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019a. Anti-efficient encoding in emergent communication. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6290–6300.

Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. 2019b. Word-order biases in deep-agent emergent communication. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5166–5175. Association for Computational Linguistics.

Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2021. Co-evolution of language and agents in referential games. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2993–3004. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1129–1141. Association for Computational Linguistics.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2137–2145.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. 2019. Emergent linguistic phenomena in multi-agent communication games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3698–3708. Association for Computational Linguistics.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2149–2159.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2020. Entropy minimization in emergent languages. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5220–5230. PMLR.

Simon Kirby. 2001. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Trans. Evol. Comput.*, 5(2):102–110.

Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2962–2967. Association for Computational Linguistics.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1223–1233. ACL.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1512–1523. ACL.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Angeliki Lazaridou and Marco Baroni. 2020. Emergent multi-agent communication in the deep learning era. *CoRR*, abs/2006.02419.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7663–7674. Association for Computational Linguistics.

David K. Lewis. 1969. *Convention: A Philosophical Study*. Wiley-Blackwell.

Fushan Li and Michael Bowling. 2019. Ease-of-teaching and language structure from emergent communication. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15825–15835.

Yaoyiran Li, Edoardo Maria Ponti, Ivan Vulic, and Anna Korhonen. 2020. Emergent communication pretraining for few-shot machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4716–4731. International Committee on Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1495–1502. AAAI Press.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. 2020. Compositional languages emerge in a neural iterated learning model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher J. Pal. 2021. Structural inductive biases in emergent communication. In *Proceedings of The 43rd Annual Meeting of the Cognitive Science Society, CogSci 2021*.

Mark Steedman. 1996. *Surface structure and interpretation*, volume 30 of *Linguistic inquiry*. MIT Press.

Mark Steedman. 2000. *The syntactic process*. Language, speech, and communication. MIT Press.

Oskar van der Wal, Silvan de Boer, Elia Bruni, and Dieuwke Hupkes. 2020. The grammar of emergent languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3339–3359. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

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

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666. AUAI Press.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 678–687. ACL.

A Modifications of CGI

INITLEX We set $\Lambda_{\text{seed}} = \emptyset$, as we do not have any prior knowledge on emergent languages.

UPDATELEX In KZGS10, UPDATELEX includes part of a potential new lexicon pruning the rest, while ours includes all of them. This is because the PRUNELEX of ADP14 would implicitly do the same thing. Moreover, the original UPDATELEX splits lexical entries as a higher-order unification problem to find f and g s.t. $h = f(g)$ or $h = f \circ g$, given a logical form h. On the other hand, ours splits the entries as a problem only to find $h = f(g)$, ensuring that $f \neq \lambda x.x.$ and g is not a function.

INITPARAM Since the algorithm can only search limited space in practice, a reasonable parameter initialization is required. KZGS10 used a statistical translation method[9], while we simply compute mean pointwise mutual information (pmi) between n-grams and logical constants. Formally, given a feature, i.e., a lexical entry $m \vdash X : \psi$, its initial parameter is defined as:

$$\frac{1}{|\mathrm{Cnst}(\psi)|} \sum_{c \in \mathrm{Cnst}(\psi)} \mathrm{pmi}(m, c)$$

if $|\mathrm{Cnst}(\psi)| > 0$ otherwise 0. $\mathrm{Cnst}(\psi)$ enumerates the logical constants (e.g. `look`, `left`, or `1`) occurring in ψ.

[9]Giza++ Model 1 (Och and Ney, 2003).

B (Hyper-)parameters

Agents For agent architecture, the hidden state size is 100. For agent optimization, the number of mini-batches per epoch is 100, the size of mini-batches is 1000, and the learning rate is 0.001. Agents train either for 200 epochs or until loss $\mathcal{L}$ for a validation dataset reaches 0. Besides, the weight of speaker's (resp. listener's) entropy regularizer $\lambda_S = 0.1$ (resp. $\lambda_L = 1$). These parameters are determined according to our preliminary experiments.

Signaling Game For signaling games, an input space $I \in \{\text{Lang-attval}, \text{Lang-conj}\}$, the size $|A|$ of an alphabet A is in $\{15, 31, 63\}$, and a message length $k \in \{4, 8\}$.

CGI For CGI, the number of iterations $T = 10$, a learning rate $\gamma = 0.1$, and a beam size for CKY parsing is 10, referring to Artzi et al. (2014) and our preliminary experiments.

TRE For TRE, a learning rate is 0.01 and the number of steps is 1000 following the implementation of Andreas (2019).

C Shuffled Emergent Language and Random Sequence

Figure 7 and Figure 9 show the comparison among CGF/L-Emergent, CGF/L-Shuffled, CGF/L-Random.

D Other Experimental Results

Figure 8 shows the scatter plot of TopSim and CGL when $I = \text{Lang-conj}$. Figure 10 shows the scatter plot of TRE and CGL when $I = \text{Lang-conj}$.

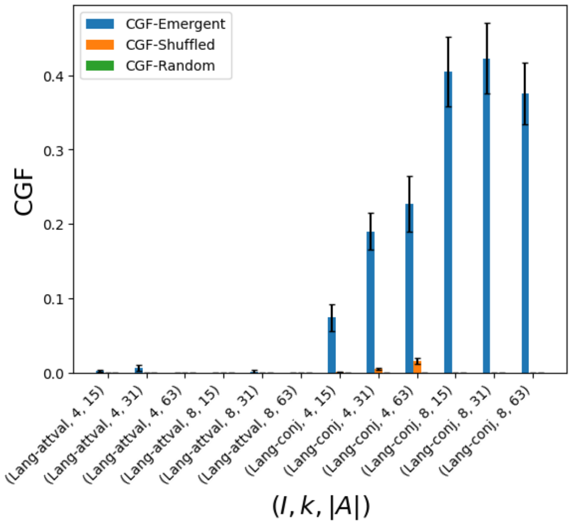

Figure 7: CGF plotted under various $(I, k, |A|)$. The error bars represent one standard error of mean.

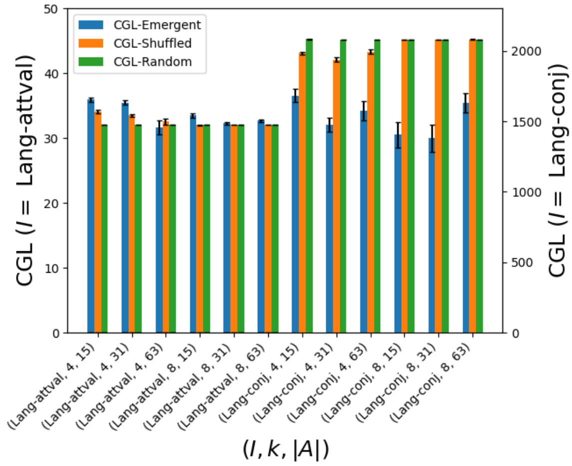

Figure 9: CGL plotted under various $(I, k, |A|)$. The error bars represent one standard error of mean.

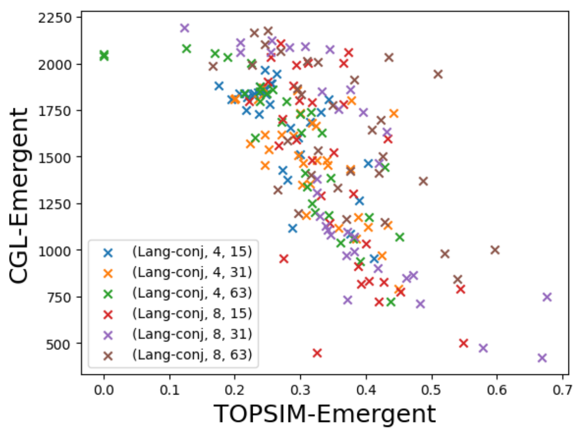

Figure 8: Scatter plot of CGL-Emergent and TopSim-Emergent.

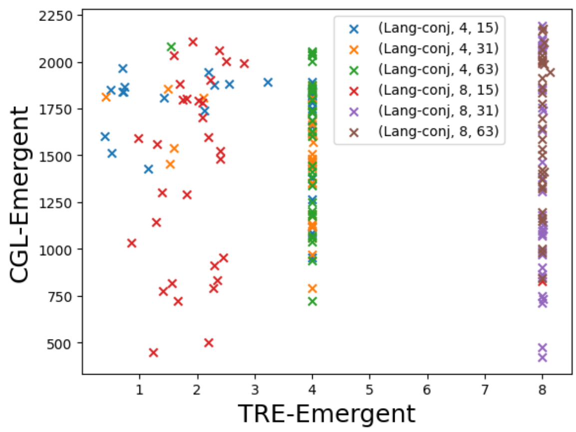

Figure 10: Scatter plot of CGL-Emergent and TRE-Emergent.