# CATEGORIAL GRAMMAR INDUCTION AS A COMPOSITIONALITY MEASURE FOR EMERGENT LANGUAGES IN SIGNALING GAMES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

This paper proposes a method to analyze the compositional structure of emergent languages using Categorial Grammar Induction (CGI). Emergent languages are communication protocols arising among agents in environments such as signaling games. Previous work has studied how similar or dissimilar emergent languages are to natural languages in compositionality. However, most of them focused on trivial compositionality, assuming flat structures in languages. We further focus on non-trivial compositionality, i.e., the relationship between hierarchical syntax and semantics. To this end, we apply CGI to emergent languages, inspired by previous NLP work. Given sentence-meaning pairs of a language, CGI induces 1) a categorial grammar that describes the syntax of the language and 2) a semantic parser that compositionally maps sentences to meanings. We also propose compositionality measures based on the grammar size and semantic parser performance. CGI and the proposed measures enable deeper insights into the non-trivial compositionality of emergent languages, while correlating well with existing measures like TopSim.

## 1 INTRODUCTION

Communication that emerges among artificial agents is called *emergent communication*, and its protocols are *emergent languages* (Lazaridou & Baroni, 2020). *Compositionality* has been an important concept in this literature since it has been pointed out that emergent languages are not similar to natural languages in that respect (Kottur et al., 2017). In addition to the research direction of how to improve compositionality (Li & Bowling, 2019; Ren et al., 2020), it has been equally important to measure the *degree* of compositionality of emergent languages (Brighton & Kirby, 2006; Andreas, 2019; Chaabouni et al., 2020). However, most of the previous work focused on trivial compositionality and assumed flat structures in languages. Little has been studied on how to measure *non-trivial compositionality* (Steinert-Threlkeld, 2019), i.e., hierarchical syntactic structure and the systematic relationship between syntax and semantics.

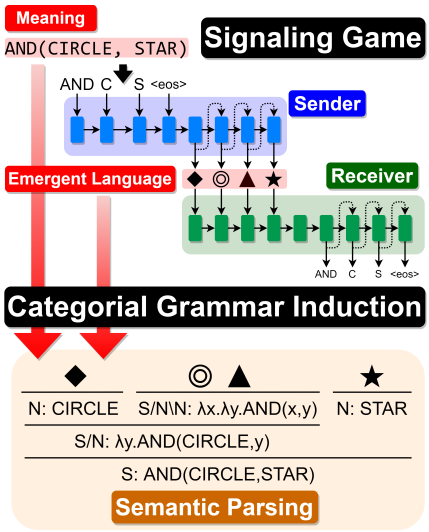



Figure 1: Illustration of a signaling game and CGI.

Inspired by previous work in the NLP literature, we propose to apply *Categorial Grammar Induction* (CGI, e.g., Zettlemoyer & Collins, 2005) in order to analyze the syntax and non-trivial compositionality of emergent languages. We also propose two CGI-based compositionality measures. Given sentence-meaning pairs of a language, a CGI algorithm induces 1) a *categorial grammar* that describes the syntax of the language and 2) a *semantic parser* that compositionally maps sentences to meanings. CGI involves useful properties for compositionality measures. For example, the performance of the induced semantic parser may indicate the systematicity and productivity (Hupkes et al., 2020) of the language, and the grammar size may indicate the compactness of the language. Another important

feature of CGI is that it is appropriate for analyzing a *signaling game*. Agents in the game are defined as either a mapping from a meaning space to a message space or its inverse. The emergent language is then represented as a set of *message-meaning pairs*, corresponding to *sentence-meaning pairs* as an input to the CGI algorithm. Figure 1 illustrates the relationship between CGI and the signaling game. In this paper, we define the meaning space as tree-structured compositional semantics, similarly to Andreas (2019).

Several compositionality measures for emergent languages have been proposed, such as TopSim (Brighton & Kirby, 2006; Lazaridou et al., 2018), TRE (Andreas, 2019), and PosDis/BosDis (Chaabouni et al., 2020). However, they assume the *trivial-compositionality*, i.e., that emergent languages have *flat* structures where each symbol is a *content* word denoting some attributes (e.g., $a, b, c, \ldots$), and the sentences are constructed with simple conjunctions of the content words (e.g., $a \wedge b \wedge c$). In contrast, the *non-trivial compositionality* assumes a hierarchical structure rather than a flat one. It involves *function* words, where some symbols have functional semantics (e.g., $\lambda x.f(x)$ in the lambda notation), and the sentences are constructed with function applications (e.g., $(\lambda x.f(x))(a) \Rightarrow f(a)$) as well as conjunctions. In natural language, for instance, verbs and prepositions are functional. Transitive verbs (e.g., "likes") can be seen as functions that take an object (e.g., "Mary") and a subject (e.g., "John"), and return a sentence ("John likes Mary"). Prepositions can be seen as functions that take a noun phrase and return a prepositional phrase. TRE is partially a non-trivial compositionality measure in the sense that it assumes hierarchical semantics, but not fully because it does not reveal whether each symbol is content or function.

Previous work (van der Wal et al., 2020) applied *Unsupervised Grammar Induction* (UGI) to analyze the syntax of emergent languages.[1] The UGI algorithm induces a grammar, given a dataset of plain sentences. UGI focuses solely on syntax because it does not involve semantic composition. UGI does not clarify which part of the syntactic composition corresponds to which part of the semantic composition. In contrast, CGI is more compositionality-oriented, as it takes both sentences and meanings as datasets and induces compositional semantics parsers. CGI can represent the semantic composition that proceeds parallel to the syntactic composition.

Our contributions are 1) we propose to apply Categorial Grammar Induction (CGI) to emergent languages for analyzing their compositional structure and 2) propose two CGI-based compositionality measures that are more aware of non-trivial compositionality. We show they can indeed measure compositionality. With the explicit assumption of hierarchical structures, our measures provide deeper insight into the non-trivial compositionality of emergent languages, while correlating with existing trivial compositionality measures like TopSim.

## 2 Background: Signaling Game and Compositionality

### 2.1 Definition of Signaling Game

Most studies on emergent communication employ *Lewis's signaling game* (Lewis, 1969) or its variant, as an environment for agents to communicate. A signaling game is defined as a quadruple $(\mathcal{I}, \mathcal{M}, S, R)$, where $\mathcal{I}$ is an *input space*, $\mathcal{M}$ is a *message space*, $S : \mathcal{I} \rightarrow \mathcal{M}$ is a *sender*, and $R : \mathcal{M} \rightarrow \mathcal{I}$ is a *receiver*. The goal is the successful comunication from $S$ to $R$, i.e., reconstruction $i = R(S(i))$ for a sampled input $i \in \mathcal{I}$. The input space $\mathcal{I}$ can vary between a set of image data (Havrylov & Titov, 2017; Lazaridou et al., 2018; Bouchacourt & Baroni, 2018), sequential data (Lu et al., 2020; Słowik et al., 2021), and attribute-value objects (Li & Bowling, 2019; Chaabouni et al., 2020; Ren et al., 2020). The message space $\mathcal{M}$ is a set of discrete sequences in most studies. The agents $S, R$ are often represented as neural networks like RNN.

### 2.2 Compositionality of Emergent Language

*Compositionality* has been crucial in the emergent communication literature. Kottur et al. (2017) noted that emergent languages are not necessarily similar to natural languages. The compositionality, in this context, is how separately the symbols of a message $m \in \mathcal{M}$ denote the pieces of meanings $i \in \mathcal{I}$. The input $i \in \mathcal{I}$ is often defined as an *attribute-value object* so that the compositionality can easily be measured. For example, Kottur et al. (2017) assumed an environment

[1] Specifically, they adopted CCL (Seginer, 2007) and DIORA (Drozdov et al., 2019).

in which there are two attributes: *color* and *shape*, each of which can take four values (e.g., *blue*, *green*, *red*, and *purple* for attribute color). If an emergent language is compositional in this environment, each message should be a combination of symbols separately denoting the color and shape attributes. Several compositionality measures have been proposed including Topographic Similarity (TopSim, Brighton & Kirby, 2006; Lazaridou et al., 2018) and Tree Reconstrctuon Error (TRE, Andreas, 2019). The former is a de facto measure in the emergent communication literature, while the latter is similar to ours in the sense that inputs $i \in \mathcal{I}$ are assumed to be tree-structured.

**Definition of TopSim** Recall that $G = (\mathcal{I}, \mathcal{M}, S, R)$ is a signaling game. Let $d_{\mathcal{I}}, d_{\mathcal{M}}$ be distance functions in $\mathcal{I}, \mathcal{M}$ respectively. TopSim is defined as the Spearman correlation between $d_{\mathcal{I}}(x, y)$ and $d_{\mathcal{M}}(S(x), S(y))$ for all combinations $x, y \in \mathcal{I}$, without repetition. In this paper, both $d_{\mathcal{I}}$ and $d_{\mathcal{M}}$ are defined as the edit distance.

**Definition of TRE** The intuition behind TRE is that if a language is compositional, the sender $S : \mathcal{I} \to \mathcal{M}$ should be approximated well by another explicitly compositional function $C : \mathcal{I} \to \mathcal{M}$. Each $i \in \mathcal{I}$ is a *binary tree* and each $m \in \mathcal{M}$ is a sequence of a fixed length $L$ over a finite alphabet $\mathcal{A}$. The computation of TRE involves a composition $C_\eta : \mathcal{I} \to \mathbb{R}^{L \times |\mathcal{A}|}$ with a trainable parameter $\eta = \langle W_1 \in \mathbb{R}^{L \times L}, W_2 \in \mathbb{R}^{L \times L}, \{E_o \in \mathbb{R}^{L \times |\mathcal{A}|}\}_o \rangle$. $C_\eta$ is defined as:

$$C_\eta(i) := \begin{cases} E_o & (i \text{ is a leaf node } o) \\ W_1 C_\eta(i_1) + W_2 C_\eta(i_2) & (i \text{ is a binary node } \langle i_1, i_2 \rangle) \end{cases} \tag{1}$$

The $r$-th row of $C_\eta(i)$ represents the logits of a categorical distribution over the $r$-th symbol of a message. Here, TRE is computed by approximating $S(i)$ with $C_\eta(i)$ via stochastic gradient descent:

$$\text{TRE} := \min_\eta \frac{1}{L \cdot |\mathcal{I}|} \sum_{i \in \mathcal{I}} \delta(C_\eta(i), S(i)), \tag{2}$$

where $\delta$ is the cross entropy loss with the Softmax layer.[2] Note that the lower TRE is, the higher compositionality is judged.

## 3 Background: Categorial Grammar Induction

This section introduces Categorial Grammar (CG) and reviews its induction (CGI) for natural languages. CGI is also feasible for analyzing emergent languages in signaling games, as it derives a lexicon and a parser from message-meaning pairs.[3]

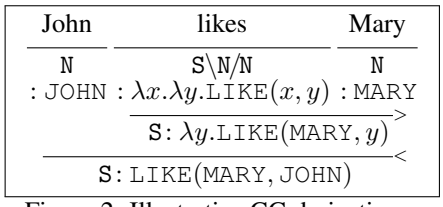


Figure 2: Illustrative CG derivation.

### 3.1 Categorial Grammar

The formalism for our semantic parsing is *Categorial Grammar* (CG, Steedman, 1996; 2000). CG consists of lexical entries and forward/backward application rules. A *lexical entry* $w \vdash X : \psi$ is a triple of a word $w$, a category $X$ (defined below), and a logical form $\psi$. Consider the following example pair of a message and its logical form:

"John likes Mary" and $\texttt{LIKE}(\texttt{MARY}, \texttt{JOHN})$

Their lexical entries can be described as follows:

$$\text{John} \vdash \texttt{N} : \texttt{JOHN}, \quad \text{likes} \vdash \texttt{S\N/N} : \lambda x.\lambda y.\texttt{LIKE}(x, y), \quad \text{Mary} \vdash \texttt{N} : \texttt{MARY}$$

Symbols like N and S\N/N represent syntactic types or *categories*. A category is either an atomic category of the form N, S or a complex category of the form $X/Y, X\backslash Y$ where $X, Y$ are categories. Nouns and sentences have atomic categories N and S respectively, while functional words such as

[2] Andreas (2019) defines $\delta$ as the L1 distance. In our preliminary experiments, however, it turned out to be too weak to obtain reasonable scores in our setting. Also, we add the normalizer $1/L$ since we vary $L$.

[3] Previous work often uses Combinatory Categorial Grammar (CCG), but we restrict it to CG. The extensive application of CCG induction is left for future work.

transitive verbs have complex categories. In addition, CG has *application rules* to describe the way to combine adjacent categories:

$$X/Y : f \quad Y : a \quad \Rightarrow \quad X : f(a) \tag{>}$$
$$Y : a \quad X\backslash Y : f \quad \Rightarrow \quad X : f(a) \tag{<}$$

where $X, Y$ are categories. The first rule ">" is the *forward application rule*, while the second one "<" is the *backward application rule*. The forward (resp. backward) application rule 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 forward/backward application rules, we can construct a derivation tree of "John likes Mary" as shown in Figure 2.

### 3.2 Log-linear Probabilistic Categorial Grammars

Given a set of lexical entries $\Lambda$ (henceforth *lexicon*), there can be multiple derivations for each message. Following Zettlemoyer & Collins (2005), we choose the most likely derivation by using a log-linear model that contains a feature vector function $\boldsymbol{\phi}$ and a parameter vector $\boldsymbol{\theta}$. Given a message $m$, the joint probability of a logical form $\psi$ and a derivation $\tau$ is defined as:

$$P(\tau, \psi \mid m; \boldsymbol{\theta}, \Lambda) = \frac{e^{\boldsymbol{\theta}^\top \boldsymbol{\phi}(m,\tau,\psi)}}{\sum_{(\tau',\psi')} e^{\boldsymbol{\theta}^\top \boldsymbol{\phi}(m,\tau',\psi')}}. \tag{3}$$

The semantic parsing problem is to find the most likely logical form $\hat{\psi}$ given a message $m$:

$$\hat{\psi} = \arg\max_{\psi} P(\psi \mid m; \boldsymbol{\theta}, \Lambda) = \arg\max_{\psi} \sum_{\tau} P(\tau, \psi \mid m; \boldsymbol{\theta}, \Lambda). \tag{4}$$

### 3.3 Categorial Grammar Induction Algorithm

Algorithm 1 is a pseudo-code for previous CG induction (CGI) algorithms. In general, the inputs are a training data $\mathcal{E} = \{(m^j, \psi^j)\}_{j=1}^N$ of message-meaning pairs, a seed lexicon $\Lambda_{\text{seed}}$, the number of iterations $T$, and a learning rate $\gamma$, while the outputs are a lexicon $\Lambda$ and a parameter vector $\boldsymbol{\theta}$. 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) via gradient descent, and optionally (4) lexicon pruning (PruneLex) that discards a lexicon no longer in use.

**Algorithm 1** Common Structure of CG Induction

**Input:** A dataset $\mathcal{E} = \{(m^j, \psi^j)\}_{j=1}^N$, a seed lexicon $\Lambda_{\text{seed}}$, the number of iterations $T$, and a learning rate $\gamma$.
**Output:** Lexicon $\Lambda$ and parameter vector $\boldsymbol{\theta}$

$\Lambda_0 \leftarrow \textsc{InitLex}(\mathcal{E}, \Lambda_{\text{seed}})$
$\boldsymbol{\theta}_0 \leftarrow \textsc{InitParam}(\mathcal{E}, \Lambda_{\text{seed}})$
**for** $t \in \{1, \ldots, T\}$ **do**
  $\Lambda_t^+ \leftarrow \textsc{UpdateLex}(\mathcal{E}, \theta_{t-1}, \Lambda_{t-1}, \Lambda_0)$
  $\theta_t \leftarrow \textsc{UpdateParam}(\mathcal{E}, \theta_{t-1}, \Lambda_t^+, \gamma)$
  $\Lambda_t \leftarrow \textsc{PruneLex}(\mathcal{E}, \theta_{t-1}, \Lambda_t^+)$
**end for**
**return** $\Lambda_T$ and $\theta_T$

ZC05 (Zettlemoyer & Collins, 2005) first formalized CGI. ZC07 (Zettlemoyer & Collins, 2007) is its improved version. In ZC05/07, InitLex is simply defined as $\Lambda_0 \leftarrow \Lambda_{\text{seed}}$ and UpdateLex relies on hand-crafted templates. KZGS10/11 (Kwiatkowski et al., 2010; 2011) modified UpdateLex so that it can create a new lexicon by automatically merging and splitting the existing lexical entries. In KZGS10/11, InitLex returns $\mathcal{E}$ itself with category S, in addition to $\Lambda_{\text{seed}}$:

$$\Lambda_0 \leftarrow \textsc{InitLex}(\mathcal{E}, \Lambda_{\text{seed}}) := \Lambda_{\text{seed}} \cup \{m^j \vdash \mathtt{S} : \psi^j \mid j = 1, \ldots, N\}. \tag{5}$$

The lexical entries are then split and merged during the iteration, seeking an appropriate segmentation (see the illustrative example in Figure 3). A problem in KZGS10/11 is that the lexicon size $|\Lambda|$ 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.

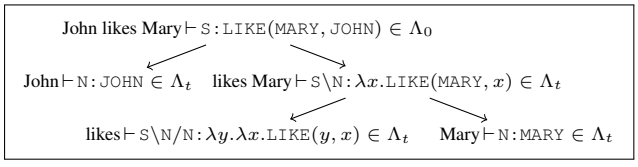



Figure 3: Illustrative splitting procedure.

# 4 PROPOSAL: CGI AS A COMPOSITIONALITY MEASURE

This section proposes two compositionality measures we call CGF and CGL. CGI has some beneficial properties from which we can define compositionality measures. We focus on the following two observations that motivate CGF and CGL, respectively. First, we can access the *F1-score* of CG-based semantic parsing. It may indicate the *systematicity* and *productivity* (Hupkes et al., 2020) of a language, i.e., the extent to which the language can combine existing lexical entries and produce novel representations. Second, we can yield a *lexicon* that implies how compactly a language can be described by CG.

## 4.1 DEFINITION OF CGF AND CGL

Let $\mathcal{E}_{\rm train}/\mathcal{E}_{\rm dev}/\mathcal{E}_{\rm test}$ be a train/validation/test dataset for the CGI algorithm. We train/validate a parser with $\mathcal{E}_{\rm train}/\mathcal{E}_{\rm dev}$ to derive a lexicon $\Lambda$ and a parameter vector $\boldsymbol{\theta}$ and test with $\mathcal{E}_{\rm test}$ to compute the F1-score.[4] Here, CGF and CGL are defined as:

$$\text{CGF} := \text{F1-score}, \quad \text{CGL} := \frac{|\Lambda|}{|\mathcal{E}_{\rm train}|}. \tag{6}$$

CGF is the F1-score of CG-based semantic parsing. By its definition, CGF $\in [0, 1]$. CGL is the lexicon size $|\Lambda|$ normalized by the train dataset size $|\mathcal{E}_{\rm train}|$. The normalization makes CGL $= 1$ at the beginning of the CGI algorithm, because we initialize the lexicon by Eq. 5. Note that the higher CGF (resp. lower CGL) is, the more compositional a language is expected to be.

## 4.2 CGI ALGORITHM SPECIFIC TO EMERGENT LANGUAGES

We have to consider the following requirements for the CGI algorithm, in accordance with Occam's razor, since there is no prior structural knowledge on emergent languages:

(1) *The feature vector function $\boldsymbol{\phi}(\cdot)$ should be as simple as possible*.

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

(3) *The lexicon size $|\Lambda|$ should be minimal*. Otherwise, it is hard to interpret CGL.

We combine the following three existing methods because no previous study satisfies them simultaneously. We follow ZC05 for (1): each feature is the count of each lexical entry used in a derivation. However, ZC05 violates (2) as it relies on manual templates. Instead, we follow KZGS10 to create a new lexicon by merging and splitting the entries already in use. KZGS10 then violates (3) since the lexicon size increases monotonically during iterations. We follow ADP14 to discard the entries no longer in use. Other modifications are detailed in Appendix A.

## 4.3 DIFFERENCE FROM EXISTING COMPOSITIONALITY MEASURES

What differentiates CGI from the existing measures like TopSim and TRE is that it can derive an explicit lexicon and a semantic parser. Although the existing measures are also mappings from message-meaning pairs $\mathcal{E}$ to a real number, they neither clarify the structure of a message space $\mathcal{M}$ nor derive any compositional function $\mathcal{M} \to \mathcal{I}$. TopSim only involves distance functions in the spaces, the choice of which is left to humans, and it does not reveal the structure of $\mathcal{M}$. In contrast, CGI can derive the structure of $\mathcal{M}$ by deriving a lexicon. TRE induces a composition $C_\eta : \mathcal{I} \to \mathcal{M}$, but not the inverse. It causes a *degenerate* language (e.g., identical messages for all meanings) to be judged compositional, contrary to our intuition (Andreas, 2019). CGI would not regard such a degenerate language as compositional since a CGI parser is a function $\mathcal{M} \to \mathcal{I}$.

# 5 EXPERIMENTAL SETUP

This section introduces a signaling game, optimization method, CGI algorithm, and evaluation metrics *specific to our experiments*.

[4]The precision and recall are defined as (#correctly parsed)/(#parsed) and (#correctly parsed)/$|\mathcal{E}_{\rm test}|$ respectively (Zettlemoyer & Collins, 2005).

### 5.1 SIGNALING GAME SETUP

**Input Space for Signaling Game** We define an input space $\mathcal{I}$ as a synthetic compositional semantics $\mathcal{D}_k$. First, let $\mathcal{D}$ be the set defined inductively as:

1. $\mathcal{D}$ contains 4 atomic objects: $\{\texttt{CIRCLE}, \texttt{TRIANGLE}, \texttt{SQUARE}, \texttt{STAR}\} \subseteq \mathcal{D}$.
2. If $x, y \in \mathcal{D}$, then $\texttt{AND}(x, y) \in \mathcal{D}$.

Then let $\mathcal{I} := \mathcal{D}_k := \{x \in \mathcal{D} \mid \texttt{AND} \text{ occurs } k \text{ times in } x\}$. For instance, $\texttt{AND}(\texttt{CIRCLE}, \texttt{STAR}) \in \mathcal{D}_1$ and $\texttt{AND}(\texttt{AND}(\texttt{CIRCLE}, \texttt{STAR}), \texttt{TRIANGLE}) \in \mathcal{D}_2$.[5] When an input $i \in \mathcal{I}$ is fed into or output from neural agents, it is flattened to the *Polish Notation* similarly to the seq2seq semantic parsing literature (Dong & Lapata, 2016).

**Message Space for Signaling Game** The message space $\mathcal{M}$ is defined as a set of discrete sequences of fixed length $L$ over a finite alphabet $\mathcal{A}$: $\mathcal{M} := \{a_1 \cdots a_L \mid a_i \in \mathcal{A}\}$.

**Architecture and Optimization** Sender and receiver agents are represented respectively as a seq2seq model based on single-layer GRUs (Cho et al., 2014) with standard attention mechanisms (Bahdanau et al., 2015). The game's goal is redefined as the minimization of Hamming distance between $i$ and $R(S(i))$, as each input $i$ is represented sequentially. As it is indifferentiable, we use REINFORCE (Williams, 1992) for optimization.

For more detailed information (e.g., hyper-parameters), see Appendix B.

### 5.2 EXPERIMENTAL PROCEDURE

The overall experimental procedure is as follows:

- For each $(\mathcal{I}, L, |\mathcal{A}|) \in \{\mathcal{D}_2, \mathcal{D}_3\} \times \{4, 8\} \times \{8, 16, 32\}$ and random_seed $\in \{1, \ldots, 8\}$:
  - Define a signaling game $G = (\mathcal{I}, \mathcal{M}, S, R)$, where $\mathcal{M} = \mathcal{A}^L$.
  - Split $\mathcal{I}$ randomly 9:1 into $\mathcal{I}_{\text{train}}$ and $\mathcal{I}_{\text{test}}$.
  - Train and validate $S, R$ with $\mathcal{I}_{\text{train}}$.
  - Split $\mathcal{I}_{\text{train}}$ randomly 8:1 into $\mathcal{I}'_{\text{train}}$ and $\mathcal{I}''_{\text{train}}$.
  - Make datasets for CGI by pairing each $i$ with the corresponding $S(i)$:

$$\mathcal{E}_{\text{train}} := \{(i, S(i))\}_{i \in \mathcal{I}'_{\text{train}}} \quad \mathcal{E}_{\text{dev}} := \{(i, S(i))\}_{i \in \mathcal{I}''_{\text{train}}} \quad \mathcal{E}_{\text{test}} := \{(i, S(i))\}_{i \in \mathcal{I}_{\text{test}}}$$

  - Train/Validate/Test a CG parser with $\mathcal{E}_{\text{train}}/\mathcal{E}_{\text{dev}}/\mathcal{E}_{\text{test}}$.

### 5.3 HOW TO EVALUATE THE EFFECTIVENESS OF OUR MEASURES

To evaluate the effectiveness of our measures, we focus on three perspectives: 1) relationship to generalization ability, 2) correlation with existing compositionality measures, and 3) score comparison between different languages.[6]

1. **Relationship to Generalization Ability** The Relationship between compositionality and generalization ability is often discussed in the emergent communication literature. We measure the generalization ability of agents with the test loss:

$$\mathcal{L}_{\text{test}} := \frac{1}{|\mathcal{I}_{\text{test}}|} \sum_{i \in \mathcal{I}_{\text{test}}} \text{Hamming}(i, R(S(i))). \tag{7}$$

   Chaabouni et al. (2020) pointed out that high compositionality implies good generalization, whereas the inverse does not. We investigate if our measures show similar tendencies.

[5] This is similar to Andreas (2019) except that our $\mathcal{D}_k$ has variable tree structures. In Andreas (2019), every input has the fixed branching structure $\texttt{AND}(\texttt{AND}(o_1, o_2), \texttt{AND}(o_3, o_4))$ where $o_1, \ldots, o_4$ are atomic objects.

[6] In Appendix C, we tried an additional perspective: 4) whether the existing method proposed to improve TopSim is also effective for our measures. We picked up the *ease-of-teaching paradigm* (Li & Bowling, 2019), and it turned out that sometimes it is effective, sometimes not.

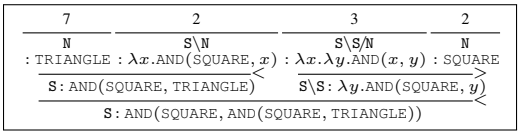
2. **Correlation with Existing Measures** As mentioned earlier, we choose TopSim and TRE for comparison. A significant correlation between the existing measures and our own would be supporting evidence for the effectiveness of our measures.
3. **Score comparison between different languages** Another possible paradigm is to compare the measured scores among several (possibly synthetic) languages (van der Wal et al., 2020). If obviously compositional (non-compositional) languages are judged compositional (non-compositional) by any measure, it is expected to be effective. To this end, we use a) the *Polish-Notated* input space (**Input**) as a fully compositional language, b) emergent language (**Emergent**) as a partially compositional language, c) *shuffled* emergent language (**Shuffled**) as a less compositional language, and d) *random* language (**Random**) as a totally non-compositional language. Here, **Input**, **Shuffled**, and **Random** are defined as:

$$\textbf{Input} := \{\text{Polish Notation of } i \mid i \in \mathcal{I}\}, \tag{8}$$

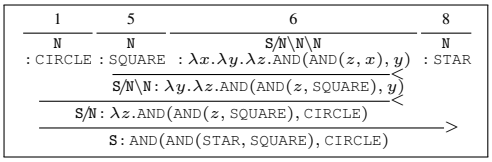
$$\textbf{Shuffled} := \{\text{Random permutation of } m \mid m = S(i), i \in \mathcal{I}\}, \tag{9}$$

$$\textbf{Random} := \{x_1^i \cdots x_L^i \mid x_j^i \sim \text{Uniform}(\mathcal{A}), i \in \mathcal{I}\}. \tag{10}$$

## 6 Experimental Results




(a) CG derivation of the message "7 2 3 2."




(b) CG derivation of the message "1 5 6 8."

Figure 4: Two derivation trees for one emergent language.

This section shows the experimental results based on the evaluation criteria in the previous section. However, first, let us show several concrete results of the CGI algorithm. Figure 4 shows two CG derivations obtained in the test split of an emergent language with $(\mathcal{I}, L, |\mathcal{A}|) = (\mathcal{D}_2, 4, 8)$. The message "7 2 3 2" is converted into the logical form $\texttt{AND}(\texttt{SQUARE}, \texttt{AND}(\texttt{SQUARE}, \texttt{TRIANGLE}))$ in Figure 4a, while "1 5 6 8" into $\texttt{AND}(\texttt{AND}(\texttt{STAR}, \texttt{SQUARE}), \texttt{CIRCLE})$ in Figure 4b. Interestingly, the CGI algorithm induces content-word-like lexical items (e.g., $7 \vdash \texttt{N} : \texttt{TRIANGLE}$), function-word-like lexical items (e.g., $6 \vdash \texttt{S}/\texttt{N}\backslash\texttt{N}\backslash\texttt{N} : \lambda x.\lambda y.\lambda z.\texttt{AND}(\texttt{AND}(x, z), y)$), and their intermediate (e.g., $2 \vdash \texttt{S}\backslash\texttt{N} : \lambda x.\texttt{AND}(\texttt{SQUARE}, x)$).[7]

### 6.1 Relationship to Generalization Ability

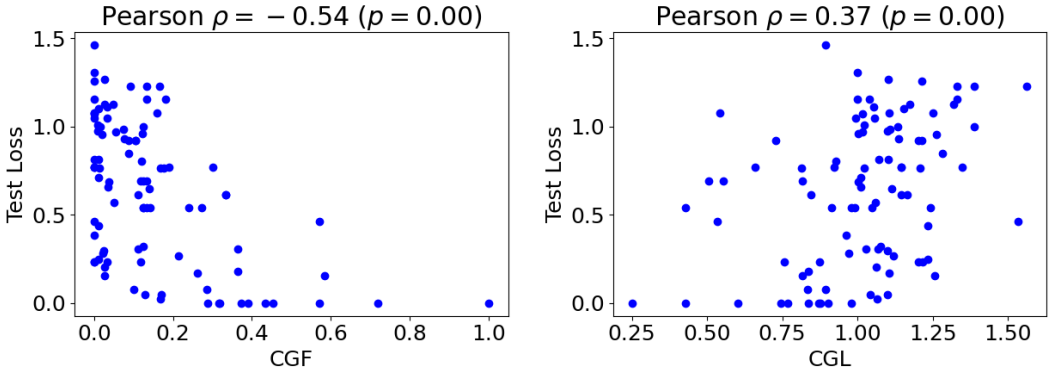



Figure 5: Relationship between the test loss $\mathcal{L}_{\text{test}}$ and our compositionality measures: CGF and CGL. The x-axis represents either CGF or CGL and the y-axis represents $\mathcal{L}_{\text{test}}$.

Figure 5 shows the relationships between the test loss $\mathcal{L}_{\text{test}}$ (Eq. 7) and our compositionality measures: CGF and CGL. CGF shows a similar tendency as Chaabouni et al. (2020) reported for other compositionality measures. That is, high CGF implies a good generalization ability, whereas the inverse implication does not hold. CGL shows similar tendencies, although several data points indicate a good CGL yet bad generalization ability.

[7]The complete list of the induced lexical items of this emergent language is found at Appendix D.

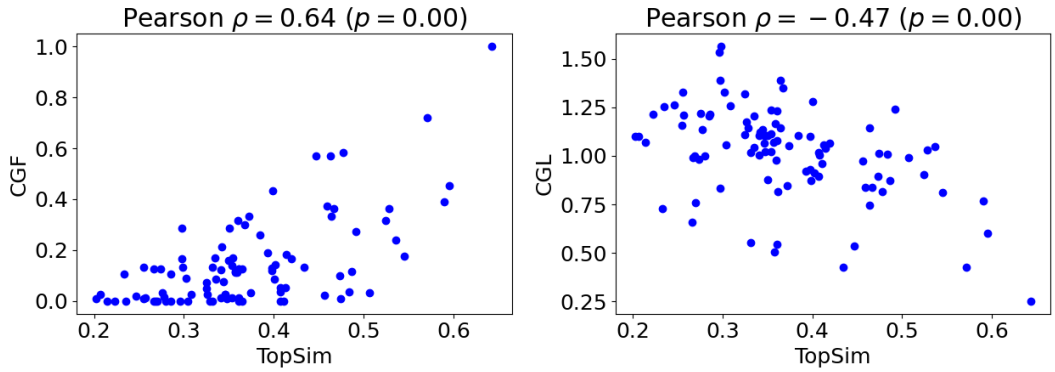



Figure 6: Correlation between TopSim and our measures: CGF and CGL. The x-axis represents the TopSim score, and the y-axis represents either CGF or CGL.

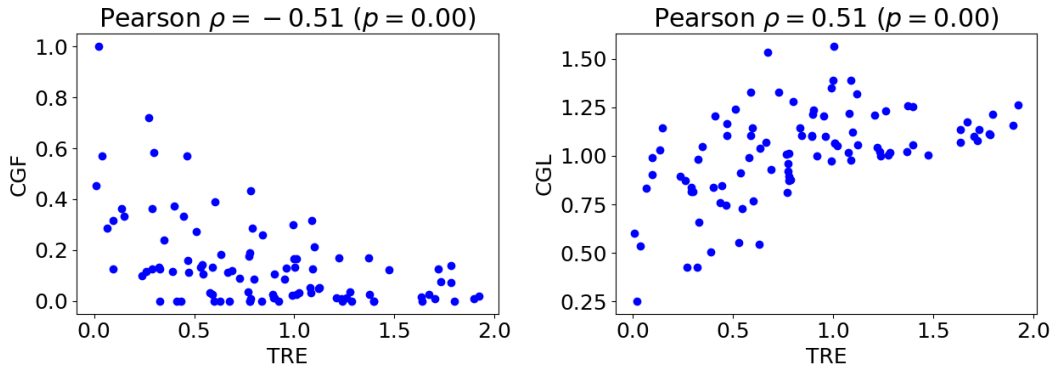



Figure 7: Correlation between TRE and our measures: CGF and CGL. The x-axis represents the TRE score, and the y-axis represents either CGF or CGL.

### 6.2 Correlation with Existing Measures

Figure 6 shows the correlations between TopSim and CGF/CGL. Our measures are significantly correlated with TopSim according to the Pearson correlation and its p-value. The Pearson $\rho = 0.64$ ($p = 0.00$) for the TopSim-CGF scores, while $\rho = -0.47$ ($p = 0.00$) for the TopSim-CGL scores. However, CGF and CGL seem to be *stricter* than TopSim. For instance, the high CGF score implies the high TopSim score, but the inverse implication does not hold. We speculate that the CGI algorithm is more sensitive to word segmentation and word orders than the edit distances $d_{\mathcal{I}}, d_{\mathcal{M}}$ used to compute the TopSim score. The edit distances may work even for languages with no clear word segmentation and orders. Figure 7 shows the correlations between TRE and CGF/CGL. Figure 7 indicates a similar tendency to Figure 6. CGF and CGL are significantly correlated with TRE, while at the same time, they are stricter measures. The Pearson $\rho = -0.51$ ($p = 0.00$) for the TRE-CGF scores, while $\rho = 0.51$ ($p = 0.00$) for the TRE-CGL scores.

### 6.3 Score Comparison between Different Languages

Figure 8 shows the CGF and CGL scores for languages **Input**, **Emergent**, **Shuffled**, and **Random**. The following clear tendency is observed in CGF: "the CGF score for **Input**" $>$ "the one for **Emergent**" $>$ "the one for **Shuffled**" $\geq$ "the one for **Random**" in each $(\mathcal{I}, L, |\mathcal{A}|)$ configuration. The tendency implies that CGF is effective as a compositionality measure. The CGF scores for **Shuffled** and **Random** are almost zero, though **Shuffled** is expected to be more compositional than **Random**. We speculate that it is due to the order sensitivity of the CGI algorithm. As we hypothesized, the CGL scores for **Input** are significantly lower than those for other languages, though the CGL score cannot clearly distinguish **Emergent**, **Shuffled**, and **Random**.

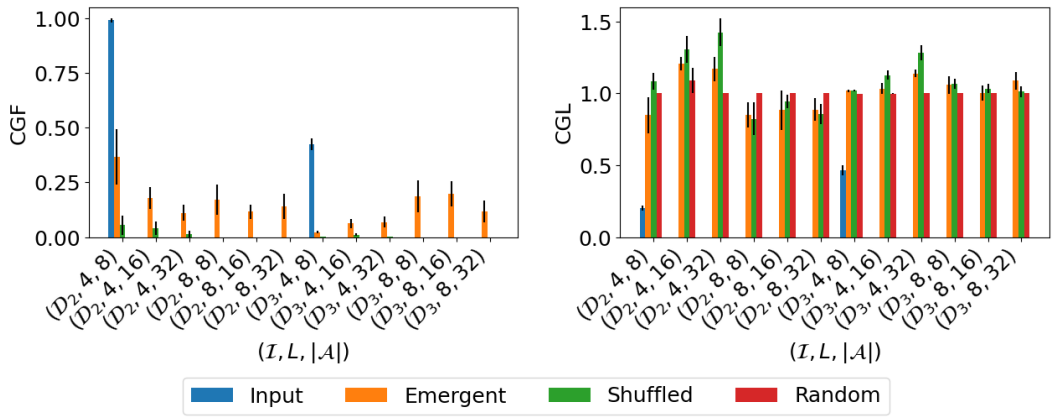



Figure 8: CGF and CGL for languages **Input**, **Emergent**, **Shuffled**, and **Random**. The x-axis represents the $(\mathcal{I}, L, |\mathcal{A}|)$ configuration and the y-axis represents each score. The CGF scores for **Random** are too small to display. The score for **Input** is displayed only twice in each figure because it only depends on the input space $\mathcal{I}$. The error bars represent the standard error.

## 7 Discussion and Limitations

Overall, the experiments showed that CGF and CGL indeed measure the compositionality of emergent languages. We evaluated their effectiveness from three perspectives: 1) relationship to generalization ability, 2) correlation with existing measures, and 3) comparison with other languages. CGF showed the expected behavior as compositionality measures. CGL might be a less effective measure than CGF. CGL showed a significant correlation with the existing compositionality measures, but it could not clearly distinguish **Emergent** from **Shuffled/Random**. Another finding is that our measures are stricter than TopSim and TRE. Whenever TopSim and TRE indicate that a language is compositional, CGF and CGL also indicate that it is compositional. In contrast, the inverse implication does not hold. It is arguably because the CGI algorithm is more sensitive to word segmentation and word order of a given language, with its strong inductive bias towards inducing CGs. It is an important feature of our CGI-based measures to analyze the syntax and its systematic relationship to semantics. Of course, the CGI algorithm may fail to find out the syntax that actually exists behind the language, since it is an approximate inference algorithm with a finite dataset. It seems safe to use our CGI-based measures in conjunction with the existing measures like TopSim. Another possible way to mitigate the issue is to run the CGI algorithm several times with different random seeds and take the best semantic parser, though we did not for the simple judgment of our measures.

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 (Figure 4). We hope that it brings deeper insights into the syntax, semantics, and non-trivial compositionality of emergent languages. A crucial limitation of our method is that it requires explicit meaning representations as an input space $\mathcal{I}$. For instance, the meaning representations are unclear when we use image data as $\mathcal{I}$. However, we speculate that *situated CGI* (Artzi & Zettlemoyer, 2013) is applicable in this case, which induces a (combinatory) categorial grammar, taking an external world into consideration. In other words, CGI may be applicable to visual referential games and 2D-grid world communication in the future. Another limitation is that the grammar formalism is restricted to CG in this paper. Some complex linguistic phenomena can be analyzed with CCG but not with CG (Steedman, 1996; 2000). The extension from CGI to *CCG Induction* will be desired in the future, especially when emergent languages are expected to be more complex than those in the standard signaling games.

## 8 Conclusion

This paper utilizes *Categorial Grammar Induction* (CGI) as a compositionality measure for emergent languages. We proposed two CGI-based compositionality measures CGF and CGL. The experiments revealed that they can measure compositionality as we hypothesized. As non-trivial compositionality measures, they also turned out to be stricter than existing measures, probably due to their sensitivity to word order and segmentation, while correlating well with the existing measures.

## REPRODUCIBILITY STATEMENT

This paper ensures the reproducibility of our experiments and results in the following way:

- The (hyper-)parameters of our experiments are specified in Section 5.1 and Appendix B.
- The overall experimental procedure is stated in Section 5.2.
- Our experimental code will be released upon acceptance.

## REFERENCES

Jacob Andreas. Measuring compositionality in representation learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJz05o0qK7.

Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Trans. Assoc. Comput. Linguistics*, 1:49–62, 2013. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/27.

Yoav Artzi, Dipanjan Das, and Slav Petrov. Learning compact lexicons for CCG semantic parsing. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *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*, pp. 1273–1283. ACL, 2014. doi: 10.3115/v1/d14-1134. URL https://doi.org/10.3115/v1/d14-1134.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 981–985. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1119. URL https://doi.org/10.18653/v1/d18-1119.

Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artif. Life*, 12(2):229–242, 2006. doi: 10.1162/artl.2006.12.2.229. URL https://doi.org/10.1162/artl.2006.12.2.229.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguistics*, 19(2):263–311, 1993.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 4427–4442. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.407. URL https://doi.org/10.18653/v1/2020.acl-main.407.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *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*, pp. 1724–1734. ACL, 2014. URL https://doi.org/10.3115/v1/d14-1179.

Li Dong and Mirella Lapata. 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, 2016. doi: 10.18653/v1/p16-1004. URL https://doi.org/10.18653/v1/p16-1004.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 1129–1141. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1116. URL https://doi.org/10.18653/v1/n19-1116.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 2149–2159, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/70222949cc0db89ab32c9969754d4758-Abstract.html.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *J. Artif. Intell. Res.*, 67:757–795, 2020. doi: 10.1613/jair.1.11674. URL https://doi.org/10.1613/jair.1.11674.

Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 2962–2967. Association for Computational Linguistics, 2017. doi: 10.18653/v1/d17-1321. URL https://doi.org/10.18653/v1/d17-1321.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 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*, pp. 1223–1233. ACL, 2010. URL https://aclanthology.org/D10-1119/.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 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*, pp. 1512–1523. ACL, 2011. URL https://aclanthology.org/D11-1140/.

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

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 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, 2018. URL https://openreview.net/forum?id=HJGv1Z-AW.

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

Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15825–15835, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/b0cf188d74589db9b23d5d277238a929-Abstract.html.

Yuchen Lu, Soumye Singhal, Florian Strub, Aaron C. Courville, and Olivier Pietquin. Countering language drift with seeded iterated learning. 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*, pp. 6437–6447. PMLR, 2020. URL http://proceedings.mlr.press/v119/lu20c.html.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. 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, 2020. URL https://openreview.net/forum?id=HkePNpVKPB.

Yoav Seginer. Fast unsupervised incremental parsing. In John A. Carroll, Antal van den Bosch, and Annie Zaenen (eds.), *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, 2007. URL https://aclanthology.org/P07-1049/.

Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher J. Pal. Structural inductive biases in emergent communication. In *Proceedings of The 43rd Annual Meeting of the Cognitive Science Society, CogSci 2021*, 2021. URL https://arxiv.org/abs/2002.01335.

Mark Steedman. *Surface structure and interpretation*, volume 30 of *Linguistic inquiry*. MIT Press, 1996. ISBN 978-0-262-69193-2.

Mark Steedman. *The syntactic process*. Language, speech, and communication. MIT Press, 2000. ISBN 978-0-262-69268-7.

Shane Steinert-Threlkeld. Towards the emergence of non-trivial compositionality, December 2019. URL http://philsci-archive.pitt.edu/16750/. Forthcoming in the journal _Philosophy of Science_.

Oskar van der Wal, Silvan de Boer, Elia Bruni, and Dieuwke Hupkes. The grammar of emergent languages. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 3339–3359. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.270. URL https://doi.org/10.18653/v1/2020.emnlp-main.270.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

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

Luke S. Zettlemoyer and Michael Collins. 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*, pp. 658–666. AUAI Press, 2005. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1209&proceeding_id=21.

Luke S. Zettlemoyer and Michael Collins. Online learning of relaxed CCG grammars for parsing to logical form. In Jason Eisner (ed.), *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*, pp. 678–687. ACL, 2007. URL https://aclanthology.org/D07-1071/.

# A MODIFICATIONS OF CGI

## A.1 ABOUT INITLEX

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

## A.2 ABOUT 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.

## A.3 ABOUT INITPARAM

Since the algorithm can only search limited space in practice, a reasonable parameter initialization is required. We follow KZGS10 to use a statistical translation method, namely, IBM Model 1 (Brown et al., 1993).

# B SOME MORE DETAIL ON EXPERIMENTAL SETUP

## B.1 (HYPER-)PARAMETERS

**Agents** For agent architecture, the hidden state size is $100$. For agent optimization, the number of mini-batches per epoch is 128, the size of mini-batches is 1024, and the learning rate is 0.001. Agents train either for 500 epochs or until the validation loss reaches 0. Besides, the weight of sender's (resp. receiver's) entropy regularizer $\lambda_S = 0.2$ (resp. $\lambda_R = 1$). These parameters are determined according to our preliminary experiments.

**Signaling Game** For signaling games, an input space $\mathcal{I} \in \{\mathcal{D}_2, \mathcal{D}_3\}$, the alphabet size $|\mathcal{A}|$ is in $\{8, 16, 32\}$, and a message length $L \in \{4, 8\}$.

**CGI** For CGI, the number of iterations $T = 20$, 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).

## B.2 ARCHITECTURE AND OPTIMIZATION

Sender and receiver agents are represented respectively as a seq2seq model based on single-layer GRUs (Cho et al., 2014) with standard attention mechanisms (Bahdanau et al., 2015). The goal of the game is relaxed to the minimization of Hamming distance between $i$ and $R(S(i))$ since each input $i$ is represented sequentially. As it is indifferentiable, we use REINFORCE (Williams, 1992) which gives the following estimated gradient:

$$\mathbb{E}[\{\text{Hamming}(i, o) - b\}\nabla \log P_S(m \mid i)P_R(o \mid m)] + \lambda_S \nabla H(P_S) + \lambda_R \nabla H(P_R),$$

where $P_S$ (resp. $P_R$) is the output distribution of sender (resp. receiver) over a message $m$ (resp. output $o$) given an input $i$ (resp. message $m$), $b$ is a mean baseline, $H(\cdot)$ denotes entropy, and $\lambda_S, \lambda_R$ are nonnegative hyper-parameters. The last terms are entropy regularizers (Williams & Peng, 1991).

## C EASE-OF-TEACHING PARADIGM AND COMPOSITIONALITY

Recall that, in the main content, we focused on the following three perspectives to evaluate the effectiveness of our compositionality measures: 1) relationship to generalization ability, 2) correlation with existing compositionality measures, and 3) comparison with other (synthetic) languages. In this section, we additionality tried the fourth perspective: 4) whether the existing method proposed to improve TopSim is also effective for our measures. To this end, we picked up the *ease-of-teaching (EoT) paradigm* (Li & Bowling, 2019), as it is a simple yet effective way to improve TopSim.

**Setup** What we have to do in the EoT setting is simple: *reset the receiver agent $R$ periodically during training*. In our experiment, we reset the receiver $R$ every 50 epochs.

**Result** Table 1 shows the results for TopSim and TRE. In most of the $(\mathcal{I}, L, |\mathcal{A}|)$ configurations, the EoT turned out to be effective for TopSim and TRE. Likewise, Table 2 shows the results for our compositionality measures CGF and CGL. It turned out that the EoT is sometimes effective, sometimes not for our measures. The EoT is not effective for CGF roughly when $(\mathcal{I}, L, |\mathcal{A}|) = (\mathcal{D}_3, *, *)$. On the other hand, the EoT is not effective for CGL when $(\mathcal{I}, L, |\mathcal{A}|) = (\mathcal{D}_2, 8, *)$.

**Discussion** These results seem to be related to our discussion in the main content that our measures CGF and CGL are *stricter* than existing measures TopSim and TRE. That is, factors contributing to the improvement of TopSim and TRE do not necessarily contribute to the improvement of CGF and CGL.

| $\mathcal{I}$ | $L$ | $\lvert\mathcal{A}\rvert$ | TopSim | TopSim (EoT) | Effective? | TRE | TRE (EoT) | Effective? |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_2$ | 4 | 8 | $0.44 \pm 0.04$ | $0.60 \pm 0.03$ | ✓ | $0.50 \pm 0.10$ | $0.16 \pm 0.09$ | ✓ |
| | | 16 | $0.39 \pm 0.04$ | $0.64 \pm 0.03$ | ✓ | $0.77 \pm 0.15$ | $0.06 \pm 0.04$ | ✓ |
| | | 32 | $0.33 \pm 0.02$ | $0.58 \pm 0.05$ | ✓ | $0.88 \pm 0.09$ | $0.08 \pm 0.04$ | ✓ |
| | 8 | 8 | $0.39 \pm 0.03$ | $0.46 \pm 0.03$ | ✓ | $0.41 \pm 0.09$ | $0.11 \pm 0.04$ | ✓ |
| | | 16 | $0.34 \pm 0.03$ | $0.47 \pm 0.07$ | ✓ | $0.43 \pm 0.08$ | $0.31 \pm 0.15$ | ✓ |
| | | 32 | $0.37 \pm 0.04$ | $0.45 \pm 0.07$ | ✓ | $0.27 \pm 0.06$ | $0.30 \pm 0.12$ | |
| $\mathcal{D}_3$ | 4 | 8 | $0.40 \pm 0.03$ | $0.50 \pm 0.03$ | ✓ | $1.04 \pm 0.10$ | $0.27 \pm 0.08$ | ✓ |
| | | 16 | $0.38 \pm 0.04$ | $0.48 \pm 0.04$ | ✓ | $1.32 \pm 0.14$ | $0.24 \pm 0.12$ | ✓ |
| | | 32 | $0.30 \pm 0.02$ | $0.42 \pm 0.06$ | ✓ | $1.73 \pm 0.06$ | $0.25 \pm 0.09$ | ✓ |
| | 8 | 8 | $0.37 \pm 0.02$ | $0.47 \pm 0.04$ | ✓ | $0.77 \pm 0.10$ | $0.41 \pm 0.08$ | ✓ |
| | | 16 | $0.38 \pm 0.04$ | $0.36 \pm 0.04$ | | $0.90 \pm 0.07$ | $0.39 \pm 0.11$ | ✓ |
| | | 32 | $0.34 \pm 0.02$ | $0.37 \pm 0.03$ | ✓ | $1.10 \pm 0.13$ | $0.23 \pm 0.14$ | ✓ |

Table 1: The TopSim scores and the TRE scores in the vanilla and EoT settings. "$\pm$" represents the standard error. The EoT is effective in most cases.

| $\mathcal{I}$ | $L$ | $\lvert\mathcal{A}\rvert$ | CGF | CGF (EoT) | Effective? | CGL | CGL (EoT) | Effective? |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_2$ | 4 | 8 | $0.37 \pm 0.13$ | $0.60 \pm 0.11$ | ✓ | $0.85 \pm 0.13$ | $0.49 \pm 0.10$ | ✓ |
| $\mathcal{D}_2$ | 4 | 16 | $0.18 \pm 0.05$ | $0.70 \pm 0.07$ | ✓ | $1.21 \pm 0.05$ | $0.53 \pm 0.12$ | ✓ |
| $\mathcal{D}_2$ | 4 | 32 | $0.11 \pm 0.04$ | $0.33 \pm 0.09$ | ✓ | $1.17 \pm 0.08$ | $0.65 \pm 0.10$ | ✓ |
| $\mathcal{D}_2$ | 8 | 8 | $0.17 \pm 0.07$ | $0.24 \pm 0.08$ | ✓ | $0.85 \pm 0.09$ | $0.85 \pm 0.09$ | |
| $\mathcal{D}_2$ | 8 | 16 | $0.12 \pm 0.03$ | $0.22 \pm 0.11$ | ✓ | $0.88 \pm 0.14$ | $1.02 \pm 0.13$ | |
| $\mathcal{D}_2$ | 8 | 32 | $0.14 \pm 0.06$ | $0.17 \pm 0.09$ | ✓ | $0.89 \pm 0.08$ | $0.94 \pm 0.06$ | |
| $\mathcal{D}_3$ | 4 | 8 | $0.02 \pm 0.01$ | $0.02 \pm 0.01$ | | $1.02 \pm 0.01$ | $0.97 \pm 0.02$ | ✓ |
| $\mathcal{D}_3$ | 4 | 16 | $0.06 \pm 0.02$ | $0.01 \pm 0.00$ | | $1.03 \pm 0.04$ | $0.99 \pm 0.01$ | ✓ |
| $\mathcal{D}_3$ | 4 | 32 | $0.07 \pm 0.02$ | $0.03 \pm 0.02$ | | $1.14 \pm 0.03$ | $0.96 \pm 0.03$ | ✓ |
| $\mathcal{D}_3$ | 8 | 8 | $0.19 \pm 0.07$ | $0.24 \pm 0.07$ | ✓ | $1.06 \pm 0.06$ | $0.88 \pm 0.05$ | ✓ |
| $\mathcal{D}_3$ | 8 | 16 | $0.20 \pm 0.06$ | $0.13 \pm 0.08$ | | $1.00 \pm 0.05$ | $0.91 \pm 0.06$ | ✓ |
| $\mathcal{D}_3$ | 8 | 32 | $0.12 \pm 0.05$ | $0.00 \pm 0.00$ | | $1.09 \pm 0.06$ | $1.01 \pm 0.03$ | ✓ |

Table 2: The CGF scores and the CGL scores in the vanilla and EoT settings. "$\pm$" represents the standard error. The EoT is sometimes effective but sometimes not.

## D FULL LIST OF LEXICAL ITEMS OF ONE EMERGENT LANGUAGE

$3 \vdash \texttt{S\textbackslash N\textbackslash N}: \lambda x4.\lambda x5.\texttt{And}(\texttt{STAR}, \texttt{And}(x4, x5))$
$1\ 6 \vdash \texttt{N}: \texttt{CIRCLE}$
$6 \vdash \texttt{S/N\textbackslash N\textbackslash N}: \lambda x3.\lambda x4.\lambda x5.\texttt{And}(\texttt{And}(x5, x3), x4)$
$6\ 3 \vdash \texttt{S\textbackslash S}: \lambda x5.\texttt{And}(\texttt{STAR}, x5)$
$7\ 6 \vdash \texttt{N}: \texttt{TRIANGLE}$
$6\ 6 \vdash \texttt{N}: \texttt{TRIANGLE}$
$2 \vdash \texttt{S\textbackslash N}: \lambda x5.\texttt{And}(\texttt{SQUARE}, x5)$
$3 \vdash \texttt{S\textbackslash S/N}: \lambda x4.\lambda x5.\texttt{And}(x4, x5)$
$1 \vdash \texttt{S\textbackslash N}: \lambda x5.\texttt{And}(\texttt{CIRCLE}, x5)$
$3 \vdash \texttt{S\textbackslash N/N\textbackslash N}: \lambda x3.\lambda x4.\lambda x5.\texttt{And}(x4, \texttt{And}(x3, x5))$
$2 \vdash \texttt{S/N}: \lambda x5.\texttt{And}(x5, \texttt{SQUARE})$
$1 \vdash \texttt{N}: \texttt{CIRCLE}$
$3 \vdash \texttt{S/N}: \lambda x5.\texttt{And}(x5, \texttt{STAR})$
$7 \vdash \texttt{S\textbackslash N}: \lambda x5.\texttt{And}(\texttt{TRIANGLE}, x5)$
$6 \vdash \texttt{N}: \texttt{TRIANGLE}$
$4\ 8 \vdash \texttt{N}: \texttt{STAR}$
$8 \vdash \texttt{S\textbackslash N}: \lambda x5.\texttt{And}(\texttt{STAR}, x5)$
$8\ 6 \vdash \texttt{N}: \texttt{STAR}$
$2 \vdash \texttt{N}: \texttt{SQUARE}$
$3 \vdash \texttt{N}: \texttt{STAR}$
$7 \vdash \texttt{N}: \texttt{TRIANGLE}$
$8 \vdash \texttt{N}: \texttt{STAR}$
$5 \vdash \texttt{N}: \texttt{SQUARE}$
$4 \vdash \texttt{N}: \texttt{STAR}$
$4 \vdash \texttt{S/N\textbackslash S}: \lambda x4.\lambda x5.\texttt{And}(x5, x4)$
$6 \vdash \texttt{N}: \texttt{STAR}$

Figure 9: The complete list of induced lexical items of the same emergent language as shown in Figure 4.