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 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 signaling game is represented as pairs of a message and 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 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

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*.

To analyze the syntax of emergent languages, we focus on categorial grammar induction (CGI, e.g., Zettlemoyer & Collins, 2005) and propose to apply it to emergent languages. Since CGI derives a 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 & 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 to compare with ours, since it is the most popular in this area (e.g., Lazaridou et al., 2018).

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. 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 CATEGORIAL GRAMMAR INDUCTION

In this section, we introduce categorial grammar (CG), CG-based semantic parsing, and its induction (CGI) for natural languages[1]. CGI is also eligible for emergent languages in signaling games, as it derives a lexicon and a parser from message-meaning pairs. Note that semantic parsing means a conversion of messages to the corresponding meanings.

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

Figure 1: Example derivation tree of "look left 1" by categorial grammar.

2.1 CATEGORIAL GRAMMAR

The formalism for our semantic parsing is *categorial grammar* (CG, Steedman, 1996; 2000). 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: $\langle$"look left 1", $\texttt{iter}(\texttt{and}(\texttt{lturn},\texttt{look}),\texttt{1})\rangle$. Their lexical entries can be described as follows:

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

Symbols like V and S\V represent *categories*. A category is either an atomic category of the form N, V, or S, or a complex category of the form X/Y or $X\backslash Y$ where X, Y are categories. The atomic categories N, V, and S stand for the linguistic notions of noun, intransitive verb, and sentence respectively[2]. In addition, CGs have *application rules* to describe the way to combine adjacent categories.

$$X/Y : f \quad Y : a \quad \Rightarrow \quad X : f(a) \qquad (>)$$
$$Y : a \quad X\backslash Y : f \quad \Rightarrow \quad X : f(a) \qquad (<)$$

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 shown in Figure 1.

2.2 LOG-LINEAR PROBABILISTIC CGS AND CG INDUCTION

Given a set of lexical entries Λ, there might be multiple derivations for each message. Following previous work (e.g., Zettlemoyer & Collins, 2005), we choose the most likely derivation by using a *log-linear model* that contains a feature vector function 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:

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

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

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

Thus far, several studies have proposed methods for *CG induction*, the task of which is to find a suitable Λ and θ from a given set of message-meaning pairs $\{(m, \psi)\}$ (e.g., Zettlemoyer & Collins, 2005; Kwiatkowski et al., 2010; Artzi et al., 2014). The induction algorithm updates Λ and θ so that $\sum_{(m,\psi)} \log P(\psi \mid m; \theta, \Lambda)$ is maximized.

3 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

[1] Although previous work is on combinatory categorial grammar (CCG), we restrict it to CG.

[2] The category of intransitive verbs is usually S/N (S/NP) or S\N (S\NP), but we regard V as an atomic category. It is because the languages we define in Section 4.1 take an imperative form without any subject.

$\mathcal{E}_{\text{train}}$ 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. Here, precision is defined as #correctly parsed/#parsed, while recall is defined as #correctly parsed/$|\mathcal{E}_{\text{test}}|$ (Zettlemoyer & Collins, 2005). Then, CGF and CGL are defined as:

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

The higher CGF (resp. lower CGL) is, the more compositional a language is judged, since a compositional language is expected to be generalized for the communication of unseen data and described by a minimal lexicon.

4 Experimental Setup

4.1 Signaling Game

Input Space We define two input spaces for our signaling game: *Lang-attval* and *Lang-conj* [3]. Lang-attval is the same as attribute-value inputs (e.g., Kottur et al., 2017), while Lang-conj is more complex. First, *Lang-attval* is defined as the set of sequences derived from the following context-free grammar with a start symbol S:

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

Next, 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:

$$\mathrm{S}'' \rightarrow \mathrm{S} \mid \mathrm{S} \, \mathrm{S}' \qquad \mathrm{S}' \rightarrow \text{and} \, \mathrm{S}$$

Game Procedure In our signaling game, the input space I is either Lang-attval or Lang-conj except that each element of I is attached with `eos` marker. The message space M is a set of discrete sequences of fixed length k over a finite alphabet A: $M \equiv \{a_1 \cdots a_k \mid a_i \in A\}$. The goal of the game is to minimize Hamming distance between an input and an output.

Architecture and Optimization Speaker and listener agents are represented as a seq2seq model based on single-layer LSTMs (Hochreiter & Schmidhuber, 1997) with standard attention mechanisms (Dong & Lapata, 2016), similarly to Chaabouni et al. (2019). As Hamming distance is indifferentiable, we use REINFORCE (Williams, 1992) for optimization.

4.2 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*, and (3) *lexicon size has to be minimal*; otherwise it is hard to interpret results. There is no existing method satisfying all of them simultaneously. Thus, we combine the methods of Zettlemoyer & Collins (2005), Kwiatkowski et al. (2010), and Artzi et al. (2014). For more detail, see Appendix A.

4.3 Other Languages for Comparison and Compositionality Metrics

To evaluate the effectiveness of our measures, we need *less* compositional languages as well as emergent languages to apply CGI. To this end, we use *AdjSwap-x* ($x \in \{1, 2\}$). AdjSwap-x is made by applying x-times random adjacent swaps to each message in the emergent language. As they are partially destroyed, AdjSwap-x should be judged less compositional.

For compositionality metrics, we use CGF, CGL, and TopSim. When clarifying the target language, we write the metrics as (*measure*)-(*language*), e.g., TopSim-Emergent and CGF-AdjSwap-1.

5 Experiments and Results

We show the experimental results in this section. For (hyper-)parameter settings, see Appendix B. First of all, Figure 2 exemplifies a derivation tree in an emergent language that CGI judges highly

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

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

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

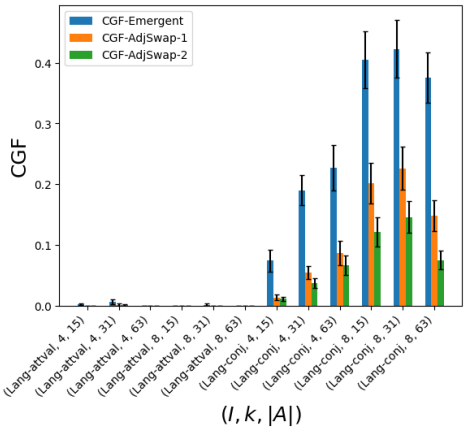

Figure 3: CGF for various $(I, k, |A|)$. The error bars represent one standard error of mean.

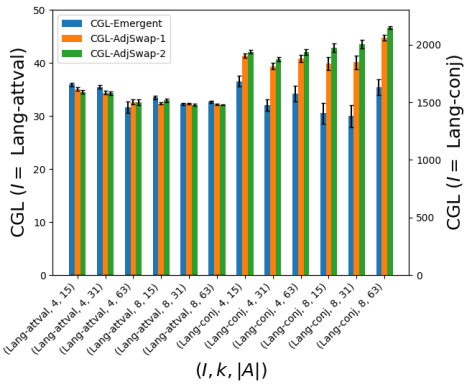

Figure 4: CGL for various $(I, k, |A|)$. The error bars represent one standard error of mean.

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.

Next, we investigate whether CGF/L works as a measure of compositionality. If CGF works, the following inequality should hold: CGF-Emergent $>$ CGF-AdjSwap-1 $>$ CGF-AdjSwap-2. Likewise, if CGL works, CGL-Emergent $<$ CGL-AdjSwap-1 $<$ CGL-AdjSwap-2. 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. For $I =$ Lang-conj, Figure 3 reveals that CGF exactly shows the order of compositionality as expected: CGF-Emergent $>$ CGF-AdjSwap-1 $>$ CGF-AdjSwap-2. Likewise, CGL in Figure 4 shows the expected order: CGL-Emergent $<$ CGL-AdjSwap-1 $<$ CGL-AdjSwap-2. Hence, CGF and CGL recognize the compositionality of emergent languages.

Finally, we check the relationship between CGF/L and TopSim. We only consider the results for $I =$ Lang-conj, where CGF/L recognizes the compositionality of emergent languages. We report that TopSim and CGF show a correlation with Pearson $\rho = 0.644$ ($p = 8.77 \times 10^{-24} \ll 0.01$). Likewise, TopSim and CGL show a correlation with Pearson $\rho = -0.689$ ($p = 2.88 \times 10^{-28} \ll 0.01$). Although ρs are moderate, p-values are considerably small. Thus, there are significant correlations between TopSim and our measures. The scatter plot between TopSim and CGF (resp. CGL) is shown in Figure 5 (resp. Figure 6) in Appendix C.

6 Conclusion and Future Work

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 lexical entries 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 & 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. 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.

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.

Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. Word-order biases in deep-agent emergent communication. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 5166–5175. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1509. URL https://doi.org/10.18653/v1/p19-1509.

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.

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.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

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/.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer G. Dy and Andreas Krause (eds.), *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*, pp. 2879–2888. PMLR, 2018. URL http://proceedings.mlr.press/v80/lake18a.html.

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.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51, 2003. doi: 10.1162/089120103321337421. URL https://doi.org/10.1162/089120103321337421.

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/.

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.

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.

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 REVIEWS AND MODIFICATIONS OF CGI

A.1 REVIEWS ON EXISTING METHODS

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 & Collins, 2005) is the first paper formalizing CGI. ZC07 (Zettlemoyer & 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}:

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

```
Λ_0 ← InitLex(E, Λ_seed)
θ_0 ← InitParam(E, Λ_seed)
    ▷ Step 0: Initialize lexicon and parameter
for t ∈ {1, ..., T} do
    Λ_t^+ ← UpdateLex(E, θ_{t-1}, Λ_{t-1}, Λ_0)
        ▷ Step 1: Update Lexicon
    θ_t ← UpdateParam(E, θ_{t-1}, Λ_t^+, γ)
        ▷ Step 2: Update Parameter
    Λ_t ← PruneLex(E, θ_{t-1}, Λ_t^+)
        ▷ Step 3: Prune Lexicon (optional)
end for
return Λ_T and θ_T
```

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

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].

A.2 MODIFICATION OF CGI

For (1), we follow ZC05 (Zettlemoyer & 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.

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[5], while we simply compute

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

[5] Giza++ Model 1 (Och & Ney, 2003).

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.

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.

C CORRELATION BETWEEN TOPSIM AND CGF/CGL

We show the scatter plot between CGF-Emergent and TopSim in Figure 5. Likewise, we show the scatter plot between CGL-Emergent and TopSim in Figure 6.

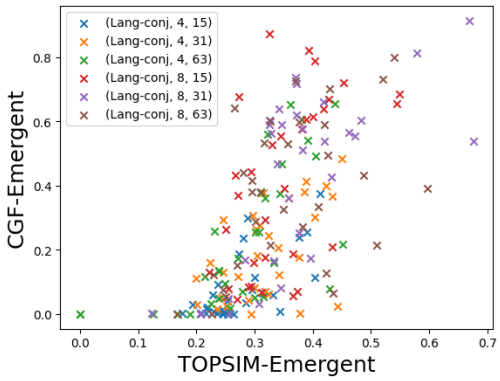

Figure 5: Scatter plot of CGF-Emergent and TopSim-Emergent.

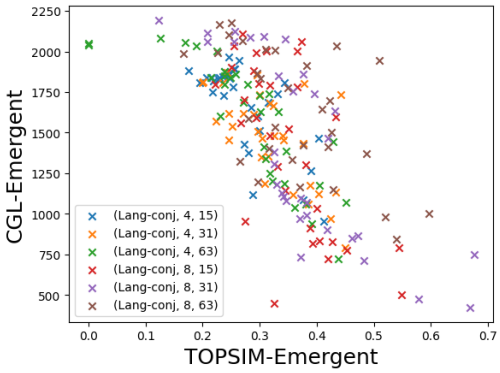

Figure 6: Scatter plot of CGL-Emergent and TopSim-Emergent.