The Grammar of Emergent Languages

Oskar van der Wal
University of Amsterdam
oskar.vanderwal@gmail.com

Silvan de Boer
University of Amsterdam
silvandeboer@gmail.com

Elia Bruni*
Institute of Cognitive Science
University of Osnabruek
elia.bruni@gmail.com

Dieuwke Hupkes*
Institute for Logic, Language and Computation
University of Amsterdam
d.hupkes@uva.nl

Abstract

In this paper, we consider the syntactic properties of languages emerged in referential games, using unsupervised grammar induction (UGI) techniques originally designed to analyse natural language. We show that the considered UGI techniques are appropriate to analyse emergent languages and we then study if the languages that emerge in a typical referential game setup exhibit syntactic structure, and to what extent this depends on the maximum message length and number of symbols that the agents are allowed to use. Our experiments demonstrate that a certain message length and vocabulary size are required for structure to emerge, but they also illustrate that more sophisticated game scenarios are required to obtain syntactic properties more akin to those observed in human language. We argue that UGI techniques should be part of the standard toolkit for analysing emergent languages and release a comprehensive library to facilitate such analysis for future researchers.

1 Introduction

Artificial agents parameterised by deep neural networks can learn to communicate using discrete symbols to solve collaborative tasks (Foerster et al., 2016; Lazaridou et al., 2017; Havrylov and Titov, 2017). A prime reason to conduct such studies, which constitute a new generation of experiments with referential games, is that they may provide insight in the factors that shaped the evolution of human languages (Kirby, 2002).

However, the *emergent languages* developed by neural agents are not human-interpretable, and little is known about their semantic and syntactic nature. More specifically, we do not know to what extent the structure of emergent languages resembles the structure of human languages, what the languages encode, and how these two things depend on choices that need to be made by the modeller.

A substantial obstacle to better understanding emergent languages is the lack of tools to analyse their properties. Previous work has concentrated primarily on understanding languages through their *semantics*, by studying the alignment of messages and symbolic representations of the meaning space (e.g. Lazaridou et al., 2018). A substantial downside of such approaches is that they are restricted to scenarios for which a symbolic representation of the meaning space is available. Furthermore, they ignore a second important aspect of language: syntax, which is relevant not just for syntactically-oriented researchers, but also for those that are interested in semantics from a compositional perspective. In this work, we aim to address this gap in the literature by presenting an analysis of the syntax of emergent languages.

We take inspiration from unsupervised grammar induction (UGI) techniques originally proposed for natural language. In particular, we use them to investigate if the languages that emerge in the typical setup of referential games exhibit interesting syntactic structure, and to what extent this depends on the maximum message length and number of symbols that the agents are allowed to use.

We first establish that UGI techniques are suitable also for our artificial scenario, by testing them on several artificial structured languages that are distributionally similar to our emergent languages. We then use them to analyse a variety of languages emerging from a typical referential game, with various message lengths and vocabulary sizes. We

*Shared senior authorship

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3339–3359,
November 16–20, 2020.

show that short messages of up to length five do not give rise to any interesting structure, while longer messages are significantly more structured than *random* languages, but yet far away from the type of syntactic structure observed in even simple human language sentences.

As such, our results thus suggest that more interesting games scenarios may be required to trigger properties more similar to human syntax and – importantly – confirm that UGI techniques are a useful tool to analyse such more complex scenarios. Their results are informative not only for those interested in the evolution of *structure* of human languages, but can also fuel further semantic analysis of emergent languages.

2 Related work

Previous work that focused on the analysis of emergent languages has primarily concentrated on semantics-based analysis. In particular, they considered whether agents transmit information about categories or objects, or instead communicate using low-level feature information (Steels, 2010; Lazaridou et al., 2017; Bouchacourt and Baroni, 2018; Lazaridou et al., 2018; Mihai and Hare, 2019, i.a.).

2.1 Qualitative inspection

Many previous studies have relied on qualitative, manual inspection. For instance, Lazaridou et al. (2018) and Havrylov and Titov (2017) showed that emergent languages can encode category-specific information through prefixing as well as word-order and hierarchical coding, respectively. Others instead have used qualitative inspection to support the claim that messages focus on pixel information instead of concepts (Bouchacourt and Baroni, 2018), that agents consistently use certain words for specific situations (Mul et al., 2019) or re-use the same words for different property values (Lu et al., 2020), or that languages represent distinct properties of the objects (e.g. colour and shape) under specific circumstances (Kottur et al., 2017; Choi et al., 2018; Słowik et al., 2020).

2.2 RSA

Another popular approach to analyse the semantics of emergent languages relies on *representational similarity analysis* (RSA, Kriegeskorte et al., 2008). RSA is used to analyse the similarity between the language space and the meaning space, in which case it is also called *topographic similarity* (Brighton et al., 2005; Brighton and Kirby, 2006; Lazaridou et al., 2018; Andreas, 2019; Li and Bowling, 2019; Keresztury and Bruni, 2020; Słowik et al., 2020; Ren et al., 2020), It has also been used to directly compare the continuous hidden representations of a neural agent with the input space (Bouchacourt and Baroni, 2018).

2.3 Diagnostic Classification

A last technique used to analyse emergent languages is *diagnostic classification* (Hupkes et al., 2018), which is used to examine which concepts are captured by the visual representations of the playing agents (Lazaridou et al., 2018), whether the agents communicate their hidden states (Cao et al., 2018), which input properties are best retained by the agent's messages (Luna et al., 2020) and whether the agents communicate about their own objects and possibly ask questions (Bouchacourt and Baroni, 2019).

3 Method

We analyse the syntactic structure of languages emerging in referential games with UGI techniques. In this section, we describe the game setup that we consider (§3.1), the resulting languages that are the subject of our analysis (§3.2) and the UGI techniques that we use (§3.3). Lastly, we discuss our main methods of evaluating our UGI setups and the resulting grammars (§3.4) as well as several baselines that we use for comparison (§3.5).

3.1 Game

We consider a game setup similar to the one presented by Havrylov and Titov (2017), in which we vary the message length and vocabulary size. In this game, two agents develop a language in which they speak about 30×30 pixel images that represent objects of different shapes, colours and sizes ($3 \times 3 \times 2$), placed in different locations. In the first step of the game, the *sender* agent observes an image and produces a discrete message to describe it. The *receiver* agent then uses this message to select an image from a set containing the correct image and three distractor images. Following Luna et al. (2020), we generate the target and distractor images from a symbolic description with a degree of non-determinism, resulting in $75k$, $8k$, and $40k$ samples for the train, validation, and test set.

Both the sender and receiver agent are modelled by an LSTM and CNN as language and visual units,

respectively. We pretrain the visual unit of the agents by playing the game once, after which it is kept fixed throughout all experiment. All trained agents thus have the same visual unit, during training only the LSTM's parameters are updated. We use Gumbel-Softmax with a temperature of 1.2 for optimising the agents' parameters, with batch size 128 and initial learning rate 0.0001 for the Adam optimiser (Kingma and Ba, 2015). In addition to that, we use early stopping with a patience of 30 to avoid overfitting. We refer to Appendix A for more details about the architectures and a mathematical definition of the game that we used.

3.2 Languages

From the described game, we obtain several different languages by varying the maximum message length L and vocabulary size V throughout experiments. For each combination of $L \in \{3, 5, 10\}$ and $V \in \{6, 13, 27\}$, we train the agents three times. In all these runs, the agents develop successful communication protocols, as indicated by their high test accuracies (between 0.95 and 1.0). Furthermore, all agents can generalise to unseen scenarios.

For our analysis, we then extract the sender messages for all $40K$ images from the game's test set. From this set of messages, we construct a disjoint induction set (90%) and validation set (10%). Because the sender may use the same messages for several different input images, messages can occur multiple times. In our experiments, we consider only the set of unique messages, which us this smaller than the total number of images. Table 1 provides an overview of the number of messages in the induction and evaluation set for each language with maximum message length L and vocabulary size V.

In the rest of this paper we refer to the three sets by denoting the message length and vocabulary size of the game they come from. For instance, $V6L10$ refers to the set of languages trained with a vocabulary size of 6 and a maximum message length of 10. Note that while the sender agent of the game may choose to use shorter messages and fewer symbols than these limits, they typically do not.

3.3 Grammar induction

For natural language, there are several approaches to unsupervised parsing and grammar induction. Some of these approaches induce the syntactic structure (in the form of a bracketing) and the constituent labels simultaneously, but most do only one of those. We follow this common practice and use a two-stage induction process (see Figure 1), in which we first infer unlabelled constituency structures and then label them. From these labelled structures, we then read out a probabilistic context free grammar (PCFG).

		seed 0		seed 1		seed 2	
L	V	induct.	eval.	induct.	eval.	induct.	eval.
3	6	162	19	141	16	147	17
	13	440	49	390	44	358	40
	27	596	67	554	62	512	57
5	6	913	102	795	89	781	87
	13	1819	203	1337	149	1614	180
	27	2062	230	1962	219	1429	159
10	6	4526	503	4785	532	4266	475
	13	8248	917	9089	1010	7546	839
	27	9538	1060	8308	924	9112	1013

Table 1: The number of messages per language for the induction and evaluation set, for all three seeds for playing the referential game.

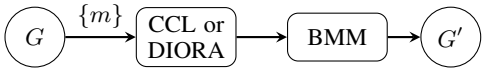


Figure 1: Our two-stage grammar induction setup. We try to reconstruct the grammar G that is hypothesised to have generated our set of messages M, using first CCL and DIORA to infer unlabeled constituency trees for all $m \in M$ and then BMM to label these trees.

3.3.1 Constituency structure induction

To induce constituency structures, we compare two different techniques: the pre-neural statistical *common cover link* parser (CCL, Seginer, 2007) and the neural parser *Deep Inside-Outside Recursive Auto-encoder* (DIORA, Drozdov et al., 2019).[1]

CCL While proposed in 2007, CCL[2] is still considered a state-of-the-art unsupervised parser. Contrary to other popular parsers from the 2000s (e.g. Klein and Manning, 2004, 2005; Ponvert et al., 2011; Reichart and Rappoport, 2010), it does not require POS-annotation of the words in the corpus, making it appropriate for our setup.

CCL is an incremental and greedy parser, that aims to incrementally add *cover links* to all words

[1] Another recent and state-of-the-art unsupervised neural parser is the *Unsupervised Recurrent Neural Network Grammar* (URNNG Kim et al., 2019). For our languages, URNNG generated exclusively right-branching trees, which is why we disregarded it in an initial stage of our experiments.

[2] http://www.seggu.net/ccl/

in a sentence. From these sets of cover links, constituency trees can be constructed. To limit the search space, CCL incorporates a few assumptions based on knowledge about natural language, such as the fact that constituency trees are generally skewed and the word distribution zipfian. In our experiments, we use the default settings for CCL.

DIORA In addition to CCL, we also experiment with the more recent *neural* unsupervised parser DIORA[3]. As the name suggests, DIORA is built on the application of recursive auto-encoders.

In our experiments with DIORA, we use a *tree-LSTM* with a hidden dimension of 50, and train for a maximum of 5 epochs with a batch size of 128. We use the GloVe framework[4] (Pennington et al., 2014) to pretrain word-embeddings for our corpus; using an embedding size of 16.

3.3.2 Constituency labelling

To label the constituency structures returned by CCL and DIORA, we use *Bayesian Model Merging* (BMM, Stolcke and Omohundro, 1994). BMM was originally approached to induce *grammars* for natural language corpora, but proved to be infeasible for that purpose. However, BMM has been successfully used to infer labels for unlabelled constituency trees (Borensztajn and Zuidema, 2007). It can therefore complement techniques such as CCL and DIORA.

The BMM algorithm starts from a set of constituency trees in which each constituent is given its own unique label. It defines an iterative search procedure that merges labels to reduce the joint description length of the data (DDL) and the grammar that can be inferred from the labelling (GDL). To find the next best merge step, the algorithm computes the effect of merging two labels on the sum of the GDL and DDL after doing the merge, where the GDL is defined as the number of bits to encode the grammar that can be inferred from the current labelled treebank with relative frequency estimation, and the DDL as the negative log-likelihood of the corpus given this grammar. To facilitate the search and avoid local minima, several heuristics and a look-ahead procedure are used to improve the performance of the algorithm. We use the BMM implementation provided by Borensztajn and Zuidema (2007)[5].

[3] https://github.com/iesl/diora

[4] https://nlp.stanford.edu/software/GloVe-1.2.zip

[5] https://github.com/pld/BMM_labels/

We refer to our complete setups with the names CCL-BMM and DIORA-BMM, respectively, depending on which constituency inducer was used in the first step.

3.4 Evaluation

As we do not know the true structure of the emergent languages, we have to resort to different measures than the traditional *precision*, *recall* and *F1 scores* that are typically used to evaluate parses and grammars. We consider three different aspects, which we explain below.

3.4.1 Grammar aptitude

To quantitatively measure how well the grammar describes the data, we compute its *coverage* on a disjoint evaluation set. Coverage is defined as the ratio of messages that the grammar can parse and thus indicates how well a grammar generalises to unseen messages of the same language. We also provide an estimate of how many messages *outside* of the language the grammar can parse – i.e. to what extent the grammar *overgenerates* – by computing its coverage on a subset of 500 randomly sampled messages.

3.4.2 Language compressibility

To evaluate the extent to which the grammar can compress a language, we consider the grammar and data description lengths (*GDL* and *DDL*), as defined by Borensztajn and Zuidema (2007). To allow comparison between languages that have a different number of messages, we consider the average message DDL.

3.4.3 Grammar nature

Lastly, to get a more qualitative perspective in the nature of the induced grammar, we consider a few statistics expressing the number of *non-terminals* and *pre-terminals* in the grammar, as well as the number of *recursive production rules*, defined as a production rule where the symbol from the left-hand side also appears on the right-hand side. Additionally, we consider the distribution of *depths* of the most probable parses of all messages in the evaluation sets.

3.5 Baselines

To ground our interpretation, we compare our induced grammars with three different language baselines that express different levels of structure. We provide a basic description here, more details can be found in Appendix D.1.

3.5.1 Random baseline

We compare all induced grammars with a grammar induced on a random language that has the same vocabulary and length distribution as the original language, but whose messages are sampled completely randomly from the vocabulary.

3.5.2 Shuffled baseline

We also compare the induced grammars with a grammar induced on languages that are constructed by *shuffling* the symbols of the emergent languages. The symbol distribution in these languages are thus identical to the symbol distribution in the languages they are created from, but the symbol *order* is entirely random.

3.5.3 Structured baseline

Aside from (semi)random baselines, we also consider a *structured baseline*, consisting of a grammar induced on languages that are similar in length and vocabulary size, but that are generated from a context-free grammar defining a basic hierarchy and terminal-class structure.[6] These structured baseline grammars indicate what we should expect if a relatively simple but yet hierarchical grammar would explain the emergent languages.

4 Suitability of induction techniques

As the grammar induction techniques we apply are defined for natural language, they are not trivially also suitable for emergent languages. In our first series of experiments, we therefore assess the suitability of the grammar induction techniques for our artificial scenario, evaluate to what extent the techniques are dependent on the exact sample taken from the training set, and we determine what is a suitable data set size for the induction techniques. The findings of these experiments inform and validate the setup for analysing the emergent languages in §5.

4.1 Grammars for structured baselines

We first qualitatively assess the extent to which CCL-BMM and DIORA-BMM are able to infer the correct grammars for the structured baseline languages described in the previous section. In particular, we consider if the induced grammars reflect the correct *word classes* defined by the preterminals, and if they capture the simple hierarchy defined on top of these word-classes.

[6] A full description, including some example grammars, can be found in Appendix B.

Results We conclude that CCL-BMM is able to correctly identify all the unique word classes for the examined languages, as well as the simple hierarchy (for some examples of induced grammars, we refer to Appendix B). DIORA-BMM performs well for the smallest languages, but for the most complex grammar ($V = 27$, $L = 10$) it is only able to find half of the word classes and some of the word class combinations. We also observe that DIORA-BMM appears to have a bias for binary trees, which results in larger and less interpretable grammars for the longer fully structured languages. Overall, we conclude that both CCL-BMM and DIORA-BMM should be able to infer interesting grammars for our artificial setup; CCL-BMM appears to be slightly more adequate.

4.2 Grammar consistency and data size

As a next step, we study the impact of the induction set sample on the resulting grammars. We do so by measuring the *consistency* of grammars induced on different sections of the training data as well as grammars induced on differently-sized sections of the training data. We consider incrementally larger message pools of size $N = \{500, 1000, 2000, 4000, 8000\}$ by sampling from the $V27L10$ language with replacement according to the original message frequencies. From each pool we take the unique messages to induce the grammar. More details on this procedure and the resulting data sets can be found in Appendix C.

We express the consistency between two grammars as the F1-score between their parses on the same test data. We furthermore consider the GDL of the induced grammars, which we compare with a baseline grammar that contains exactly one prediction rule for each message. If the GDL of the induced grammar is not smaller than the GDL of this baseline grammar, then the grammar was not more efficient than simply enumerating all messages.

The experiments described above provide information about the sensitivity of the grammar induction techniques on the exact section of the training data as well as the size of the training data that is required to obtain a consistent result. We use the results to find a suitable data set size for the rest of our experiments.

Results Overall, the experiments show that CCL-BMM has higher consistency scores than DIORA-BMM, but also more variation between different

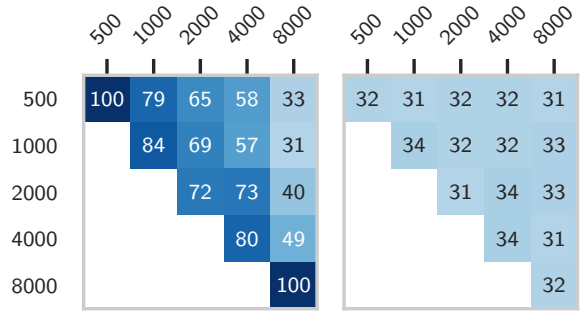


Figure 2: The consistencies for CCL-BMM (left) and DIORA-BMM (right) for language set $V27L10$. The axes show the message pool sizes (N) for inducing the compared grammars.

induction set sizes (see Figure 2). From the changing consistencies of CCL-BMM with increasing the number of messages, we conclude that differences in data-set size influence its grammar induction considerably. We believe that the low consistency scores of DIORA-BMM are due to the strongly stochastic nature of the neural parser.

For both CCL-BMM and DIORA-BMM, the evaluation set coverage increases with the induction set-size, although CCL-BMM reaches a near perfect coverage much faster than DIORA-BMM. Furthermore, the GDL implies a lower bound for the required induction set size, since the GDL is only smaller than its baseline for $N > 2000$ with CCL-BMM, while the crossover point is even larger for DIORA-BMM. More details on the progressions of the coverage and GDL can be found in the appendix in Figures C.1 and C.2 respectively.

To conclude, while a small induction set would suffice for CCL, we decide to use all messages of the induction set, because DIORA requires more data for good results, and we see no evidence that this impairs the performance of CCL-BMM.

5 Analysing emergent languages

Having verified the applicability of both CCL-BMM and DIORA-BMM, we use them to induce grammars for all languages described in §3.2. We analyse the induced grammars and parses, comparing with the structured, shuffled, and random baselines introduced in §3.5.

5.1 Grammar aptitude and compressibility

We first quantitatively evaluate the grammars, considering the description lengths and their evaluation and overgeneration coverage, as described in §3.4.

As a general observation, we note that the GDL increases with the vocabulary size. This is not surprising, as larger vocabularies require a larger number of lexical rules and allow for more combinations of symbols, but indicates that comparisons across different types of languages should be taken with care.

5.1.1 L3 and L5

As a first finding, we see that little to no structure appears to be present in the shorter languages with messages of length 3 and 5: there are no significant differences between the emergent languages and the random and shuffled baseline (full plots can be found in the appendix, Figures D.1 and D.2). Some of the grammars for the emergent $L3$ languages and random baselines, however, have a surprisingly low GDL. Visual inspection of the trees suggests that this is due to the fact that the grammars approach a trivial form, in which there is only one pre-terminal X that expands to every lexical item in the corpus, and one production rule $S \rightarrow XXX$.[7] This result is further confirmed by the *coverages* presented in Table 2, which illustrates that the grammars for the $L3$ and $L5$ languages can parse not only all sentences in these languages, but also all other possible messages with the same length and vocabulary.

Interestingly, for DIORA-BMM, there are also no significant differences for the *structured* baselines. We hypothesise that this may stem from DIORA's inductive bias and conclude that for the analysis of shorter languages, CCL-BMM might be more suitable.

5.1.2 L10

In the $L10$ languages, we find more indication of structure. As can be seen in Figure 3, the emergent grammars differ all significantly from all baselines grammars ($p < .05$) and most strongly from the *random baseline* ($p < .001$). The GDL of the shuffled baseline grammar is in-between the language and random baseline grammar, suggesting that some regularity may be encoded simply in the frequency distribution of the symbols.

The average DDL of the $L10$ languages, however, also differs considerably from the baselines, but in the other direction: both the structured and the completely random baseline are much smaller than the emergent language DDL. An explanation for this discrepancy is suggested when looking at

[7] In the case of DIORA-BMM, it is a trivial *binary* tree $S \rightarrow AX$ and $A \rightarrow XX$.

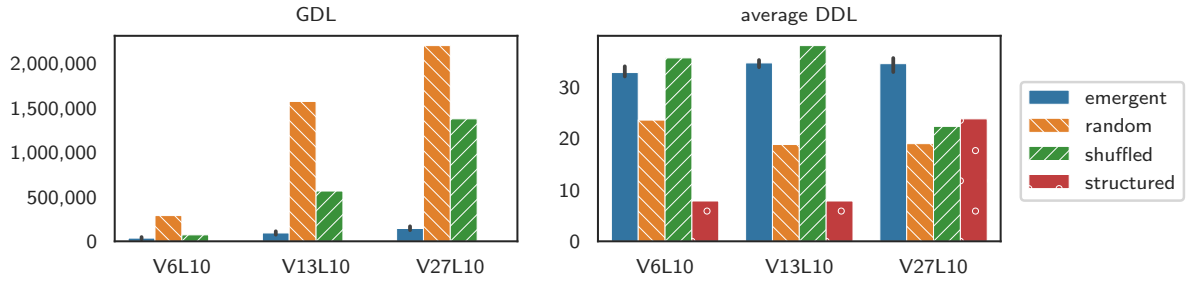


Figure 3: The grammar description lengths (GDL) and average data description lengths (DDL) for the CCL-BMM induced grammars with $L = 10$. The languages with $L = \{3, 5\}$ and the DIORA-BMM induced grammars are left out and can be found in Figures D.1 and D.2. The GDL of the *structured baseline* is too small to be seen.

their *coverages*. A good grammar has a high coverage on an independent evaluation set with messages from the same language, but a low coverage on a random sample of messages outside of the language (which we measure with *overgeneration coverage*, see §3.4). A perfect example of such a grammar is the CCL-BMM grammar inferred for the structured baseline, which has a coverage of 100% for the evaluation set but approximately 0% outside of it (see Table 2). For the $V13L10$ and $V27L10$ languages, we observe a similar pattern.

Coming back to the random languages, we can see that their grammars do not generalise to *any* message outside of their induction set. This result suggests that for these languages, the induction method resulted in a large grammar that keeps the DDL low at the expense of a larger GDL, by simply overfitting to exactly the induction set.

Concerning the coverage, another interesting finding is that the shuffled baseline often has a higher coverage than the random baseline. Combined with the generally higher average DDL, this suggests that the induction methods are less inclined to overfit the shuffled baselines. This might be explained by the regularities present in the shuffled messages through the frequencies of the symbols, as well as their co-occurrences within messages.

5.2 Nature of syntactic structure

The description lengths and coverage give an indication of whether there is *any* structure present in the languages, we finish with an explorative analysis of the *nature* of this structure. We focus our analysis on the $V13L10$ and $V27L10$ languages, which we previously found most likely to contain interesting structure.

		evaluation (%)		overgeneration (%)			
L	V	emerg.	struct.	emerg.	rand.	shuf.	struct.
3	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
5	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
10	6	100	100	78±2	0	94	0
	13	**98±1**	100	**3±1**	0	13	0
	27	**96±1**	100	**1±1**	0	0	0

Table 2: The average evaluation and overgeneration coverage for the CCL-BMM induced grammars. In bold we emphasise where we recognise a pattern of high evaluation coverage, but low overgeneration coverage. Standard deviations of < 0.5 for the emergent languages are left out.

5.2.1 Word class structure

We first examine if there is any structure at the lexical level, in the form of *word classes*. We consider the number of terminals per pre-terminal and vice versa. We will discuss the most important results here, the complete results can be found in the appendix, in Figure D.3.

A first observation is that in all grammars each symbol is unambiguously associated with only one pre-terminal symbol, indicating that there is no ambiguity with respect to the word class it belongs to. The number of terminals per pre-terminal suggests that our grammar induction algorithms also do not find many word classes: with some notable exceptions, every pre-terminal symbols expand only to a single terminal symbol. Interestingly, some of these exceptions overlap between CCL-BMM and DIORA-BMM (see Table 3), suggesting that they in fact are indicative of some form of lexical structure.

seed	CCL-BMM	DIORA-BMM
0	$\{14, \mathbf{16}, \mathbf{24}\}$	$\{\mathbf{16}, 19, \mathbf{24}\}$
1	$\{0, \mathbf{10}\}$	$\{\mathbf{10}, 22\}$
2	none	$\{0, 18\}$

Table 3: An overview of the captured word classes found in language $V27L10$ by CCL-BMM and DIORA-BMM. The overlap between the word-classes found by both setups is indicated in bold.

L	V	emergent	random	shuffled	structured
10	6	34.7 ±0.9	36.0	36.0	2.0*
	13	78.7 ±6.0	169*	137*	2.0*
	27	192 ±65	441*	262	2.0

Table 4: The number of unique *pre-terminal groups* in the CCL-BMM induced grammars for $L = 10$. A pre-terminal group constitutes the right-hand side of a production rule leading only to pre-terminals or symbols. An asterisk (*) indicates a significant difference with the baseline value ($p < .05$).

5.2.2 Higher level structure

We next check if the trees contain structure one level above the pre-terminals, by computing if pre-terminals can be grouped based on the non-terminal that generates them (e.g. if there is a rule $K \rightarrow A\ B$ we say that K generates the group $A\ B$). Specifically, we count the unique number of *pre-terminal groups*, defined by each right-hand side consisting solely of pre-terminals and symbols. If there is an underlying linguistic structure that prescribes which pre-terminals belong together (and in which order), it is expected that fewer groups are required to explain the messages than if no such hierarchy were present. Indeed, the number of *pre-terminal groups* (see Table 4) shows this pattern, as we discover a significantly smaller number of groups than the random baseline. These results thus further confirm the presence of structure in the $V13L10$ and $V27L10$ languages.

As a tentative explanation, we would like to suggest that perhaps the symbols in the emergent languages are more akin to characters than to words. In that case, the pre-terminal groups would represent the words, and the generating non-terminals the word-classes. For both CCL-BMM and DIORA-BMM, the average number of pre-terminal groups generated by these non-terminals is $2.4\pm < 0.01$ for the emergent languages, while it is 1.0 for the shuffled and random baselines. This suggests that the pre-terminal groups share in syntactic function. Such observations could form a fruitful basis for further semantic analysis of the languages.

5.2.3 Recursion

Lastly, we would like to note the lack of recursive production rules in nearly all induced grammars. While this is not surprising given both the previous results as well as the simplicity of the meaning space, it does suggest that perhaps more interesting input scenarios are required for referential games.

5.3 CCL vs DIORA

We ran all our experiments with both CCL-BMM and DIORA-BMM. There were similarities, but also some notable differences. Based on the GDL, CCL-BMM seems more suitable to analyse shorter languages, but earlier tests with reconstructing the structured baseline grammars (see §4.1) suggest that DIORA-BMM also performs worse on languages with a *larger* message length and vocabulary size; leading us to believe that CCL-BMM is more appropriate for our setup.

Another difference concerns the distribution of the tree depths, which reflects mostly skewed and binary trees for CCL-BMM for $L = 10$, but more evenly distributed depths for DIORA-BMM (for a plot of the depth distributions, we refer to D.4). An example of this difference is shown in Figure 4. A possible explanation is that CCL-BMM is more biased towards fully right-branching syntax trees, since these are a good baseline for natural language. Alternatively, these trees might actually reflect the emergent languages best, perhaps because of the left-to-right nature of the agents' LSTMs. Additional work is required to establish which type of trees better reflect the true structure of the emergent languages.

6 Conclusion

While studying language and communication through *referential games* with *artificial agents* has recently regained popularity, there is still a very limited amount of tools available to facilitate the analysis of the resulting emergent languages. As a consequence, we still have very little understanding of what kind of information these languages encode. In this paper, for the first time, we focus on *syntactic* analysis of emergent languages.

We test two different unsupervised grammar induction (UGI) algorithms that have been successful for natural language: a pre-neural statistical one, CCL, and a neural one, DIORA. We use them to

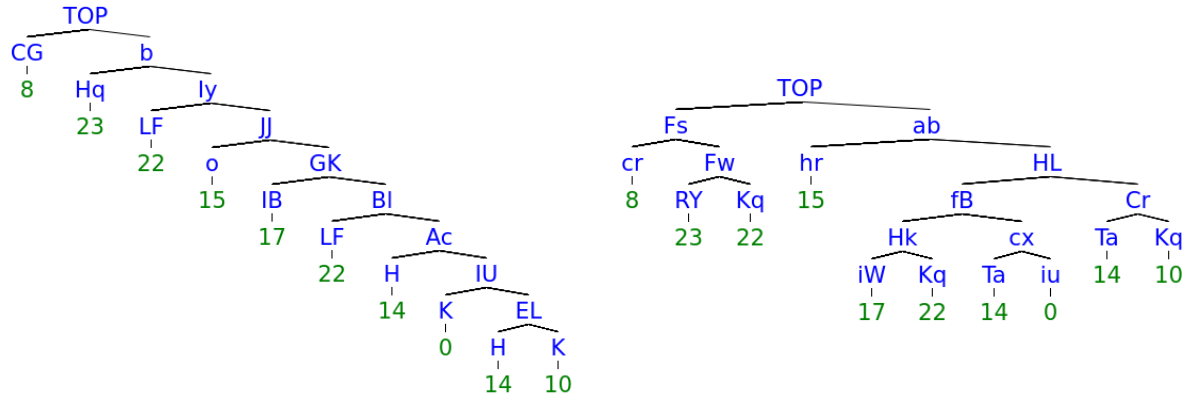


Figure 4: Example parse trees from the same $V27L10$ evaluation set by a CCL-BMM (left) and DIORA-BMM (right) induced grammar. It should be noted that this is one of the few exceptions for $L = 10$ where some symbols share a pre-terminal.

infer grammars for a variety of languages emerging from a simple referential game and then label those trees with BMM, considering in particular the effect of the message length and vocabulary size on the extent to which structure emerges.

We first confirm that the techniques are capable of inferring interesting grammars for our artificial setup and demonstrate that CCL appears to be a more suitable constituency parser than DIORA. We then find that the shorter languages, with messages up to 5 symbols, do not contain any interesting structure, while languages with longer messages appear to be substantially more structured than the two random baselines we compare them with. Interestingly, our analysis shows that even these languages do not appear to have a notion of *word classes*, suggesting that their symbols may in fact be more akin to *letters* than to words. In light of these results, it would be interesting to explore the use of unsupervised tokenisers that work well for languages without spaces (e.g. SentencePiece Kudo and Richardson, 2018) prior to our approach and to try other word embedding models for DIORA, such as the *character*-based ELMo embeddings[8] (Peters et al., 2018) or the more recent BERT (Devlin et al., 2019).

Our results also suggest that more sophisticated game scenarios may be required to obtain more interesting structure. UGI could provide an integral part in analysing the languages emerging in such games, especially since it – contrary to most techniques previously used for the analysis of emergent languages – does not require a description of the hypothesised semantic content of the messages. Examples of more sophisticated game scenarios are bidirectional conversations where multi-symbol messages are challenging to analyse (Kottur et al., 2017; Bouchacourt and Baroni, 2019) or games with image sequences as input (Santamaría-Pang et al., 2019).

We argue that while the extent to which syntax develops in different types of referential games is an interesting question in its own right, a better understanding of the syntactic structure of emergent languages could also provide pivotal in better understanding their *semantics*, especially if this is considered from a compositional point of view. To facilitate such analysis, we bundled our tests in a comprehensive and easily usable evaluation framework.[9] We hope to have inspired other researchers to apply syntactic analysis techniques and encourage them to use our code to evaluate new emergent languages trained in other scenarios.

[8] DIORA already supports ELMo vectors besides GloVe.

[9] https://github.com/i-machine-think/emergent_grammar_induction

Acknowledgments

DH is funded by the Netherlands Organization for Scientific Research (NWO), through a Gravitation Grant 024.001.006 to the Language in Interaction Consortium. EB the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 790369 (MAGIC).

References

J. Andreas. 2019. Measuring compositionality in representation learning. In *Proceedings of the 7th Inter-*

national Conference on Learning Representations (ICLR).

G. Borensztajn and W. Zuidema. 2007. Bayesian model merging for unsupervised constituent labeling and grammar induction. Technical report, ILLC.

D. Bouchacourt and M. 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 (EMNLP)*, pages 981–985.

D. Bouchacourt and M. Baroni. 2019. Miss tools and mr fruit: Emergent communication in agents learning about object affordances. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 3909–3918.

H. Brighton and S. Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242.

H. Brighton, K. Smith, and S. Kirby. 2005. Language as an evolutionary system. *Physics of Life Reviews*, 2(3):177–226.

K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, and S. Clark. 2018. Emergent communication through negotiation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

E. Choi, A. Lazaridou, and N. de Freitas. 2018. Compositional obverter communication learning from raw visual input. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 4171–4186.

A. Drozdov, P. Verga, M. Yadav, M. Iyyer, and A. McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 1129–1141.

J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2137–2145.

S. Havrylov and I. Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2149–2159.

D. Hupkes, S. Veldhoen, and W. Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

B. Keresztury and E. Bruni. 2020. Compositional properties of emergent languages in deep learning. *CoRR*, abs/2001.08618.

Y. Kim, A. M. Rush, L. Yu, A. Kuncoro, C. Dyer, and G. Melis. 2019. Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 1105–1117.

D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

S. Kirby. 2002. Natural language from artificial life. *Artificial life*, 8(2):185–215.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, page 478.

D. Klein and C. D. Manning. 2005. Natural language grammar induction with a generative constituent-context model. *Pattern Recognition*, 38(9):1407–1419.

S. Kottur, J. M. F. Moura, S. Lee, and D. 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)*, pages 2962–2967.

N. Kriegeskorte, M. Mur, and P. A. Bandettini. 2008. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

T. Kudo and J. Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), System Demonstrations*, pages 66–71.

A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

A. Lazaridou, A. Peysakhovich, and M. Baroni. 2017. Multi-Agent Cooperation and the Emergence of (Natural) Language. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

F. Li and M. Bowling. 2019. Ease-of-teaching and language structure from emergent communication. In *Advances in Neural Information Processing Systems (NIPS)*, pages 15825–15835.

Y. Lu, S. Singhal, F. Strub, O. Pietquin, and A. C. Courville. 2020. Countering language drift with seeded iterated learning. *CoRR*, abs/2003.12694.

D. R. Luna, E. M. Ponti, D. Hupkes, and E. Bruni. 2020. Internal and external pressures on language emergence: Least effort, object constancy and frequency. In *EMNLP-findings 2020*.

D. Mihai and J. Hare. 2019. Avoiding hashing and encouraging visual semantics in referential emergent language games. *CoRR*, abs/1911.05546.

M. Mul, D. Bouchacourt, and E. Bruni. 2019. Mastering emergent language: learning to guide in simulated navigation. *CoRR*, abs/1908.05135.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 2227–2237.

E. Ponvert, J. Baldridge, and K. Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1077–1086.

R. Reichart and A. Rappoport. 2010. Improved fully unsupervised parsing with zoomed learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–693.

Y. Ren, S. Guo, M. Labeau, S. B. Cohen, and S. Kirby. 2020. Compositional languages emerge in a neural iterated learning model. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

A. Santamaría-Pang, J. R. Kubricht, C. Devaraj, A. Chowdhury, and P. H. Tu. 2019. Towards semantic action analysis via emergent language. In *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 224–228.

Y. Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 384–391.

A. Słowik, A. Gupta, W. L. Hamilton, M. Jamnik, S. B. Holden, and C. Pal. 2020. Exploring structural inductive biases in emergent communication. *CoRR*, abs/2002.01335.

L. Steels. 2010. Modeling the formation of language in embodied agents: Methods and open challenges. In *Evolution of Communication and Language in Embodied Agents*, pages 223–233. Springer.

A. Stolcke and S. Omohundro. 1994. Inducing probabilistic grammars by bayesian model merging. In *International Colloquium on Grammatical Inference*, pages 106–118. Springer.

A Definition of the referential game

The languages emerge from two agents playing a referential game with a setup similar to Havrylov and Titov (2017). In each round of the game, the sender samples a message m describing the target image t to the receiver. m consists of up to L symbols sampled from a vocabulary with size V.[10] The receiver has to identify the described image from a set with t and three other distracting images in random order. The images are created by generating a shape with a certain colour and size, on a logical grid. In the game, two images are the same if they have the same colour, shape, and size, even when differently positioned. Table A.1 provides an overview of the agents' architectures used in this game.

LSTM	Embedding size	256
	Hidden layer size	512
CNN	# of convolutional layers	5
	# of filters	20
	Kernel size	3
	Stride	2
	No padding	
	Activation function	ReLU

Table A.1: Parameters for the sender and receiver architecture. The convolutional layers are followed by batch normalisation.

B Fully structured languages

For all the configurations of L and V of our emergent languages (see §3.2), we create a simple grammar containing word classes, each with a disjoint set of symbols. Furthermore, two pre-terminals form a group that can be placed either at the beginning or the end of the message or both, while the other pre-terminals occupy the remaining spots in fixed order. The smaller grammars repeat word classes to ensure enough messages for the induction and evaluation.

All the possible messages are randomly divided over a induction and evaluation set (80% and 20% respectively). Table B.1 provides more details on the data sets used for each language configuration.

[10] Technically, the vocabulary also contains a *stop character* and the sender is allowed to generate messages shorter than L. However, typically the messages have a length of L. For the analyses in this paper we have removed all stop characters in a pre-processing step and we do not count it as part of L and V.

L	V	total	induction	evaluation
3	6	16	12	4
	13	160	128	32
	27	1458	1166	292
5	6	24	19	5
	13	378	302	76
	27	15480	2000	500
10	6	24	19	5
	13	32	25	7
	27	52488	2000	500

Table B.1: An overview of the total number of possible messages that can be generated for each L and V configuration, as well as the sizes of the induction and evaluation sets. The size of the induction set is capped at 2000 to keep the grammar induction computationally feasible. When evaluating the grammars a maximum number of 500 messages of either set is used.

B.1 Example grammars

In the following examples, `TOP` denotes the start symbol, `NP` the pre-terminal group, and the numbers the terminals that represent the symbols in the generated messages.

The structured baseline grammar for $V = 13$ and $L = 5$ is represented as:

```
TOP -> NP AP
TOP -> AP NP
TOP -> NP VP NP
NP  -> A B
AP  -> E C D
VP  -> E
A   -> 0 | 1 | 2
B   -> 3 | 4 | 5
C   -> 6 | 7 | 8
D   -> 9 | 10
E   -> 11 | 12
```

The resulting CCL-BMM induced grammar is:

```
TOP -> H E A
TOP -> A G
G   -> H A | H E
E   -> B F
A   -> C D
C   -> 0 | 1 | 2
D   -> 3 | 4 | 5
B   -> 6 | 7 | 8
F   -> 9 | 10
H   -> 11 | 12
```

and DIORA-BMM finds:

```
TOP -> K A
TOP -> L D
TOP -> B J
TOP -> N H
TOP -> J B
E -> C K
A -> O D | F H | D J
G -> B C | K F
O -> F K | D E
F -> D C
H -> M I
L -> J K | B E | G K | K O
B -> K D
J -> E D | C B | C H
```

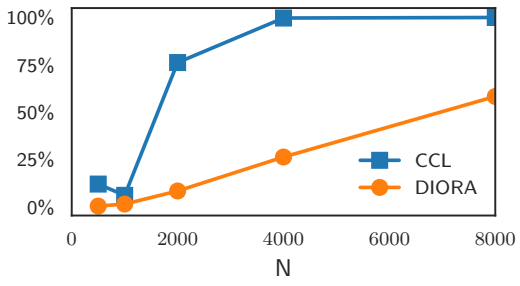


Figure C.1: Average evaluation coverage of the CCL-BMM and DIORA-BMM induced grammars ($V27L10$) against the induction pool size N.

```
N -> K F
K -> 0 | 1 | 2
D -> 3 | 4 | 5
M -> 6 | 7 | 8
I -> 9 | 10
C -> 11 | 12
```

C Consistency and suitable data set size

The number of messages in the induction set might influence the properties of the grammars induced from it. To investigate these effects, we perform induction experiments on different sub-samples of the language $V27L10$. We compare the induced grammars on their consistency and study the progression of the evaluation coverage and GDL.

The consistency of a setup is computed on different samples of a data set to study the effect of the data set size as well as to show how dependent the algorithm is on the exact selection of induction messages. We create incrementally larger pools by sampling a fixed number of randomly selected messages from the data-set, resulting in pool sizes $N = \{500, 1000, 2000, 4000, 8000\}$. The messages are sampled with replacement according to the frequency in the original language. From these pools we then only consider the unique messages. The procedure is repeated three times for each N to obtain an average consistency.

Subsequently, we study the average evaluation coverage and GDL for these grammars. The resulting progression of the evaluation coverage is shown in Figure C.1. The coverage is evaluated with respect to the disjoint set consisting of 10% of the language's messages. We study the GDL against the number of messages compared to the baseline grammar of one production rule for each message in the induction set in Figure C.2.

L	V	shuffled	random
3	6	147	150
	13	358	396
	27	512	554
5	6	913	829
	13	1819	1590
	27	1962	1817
10	6	4266	4525
	13	8248	8294
	27	9112	8986

Table D.1: Number of messages per language for the shuffled and random baseline.

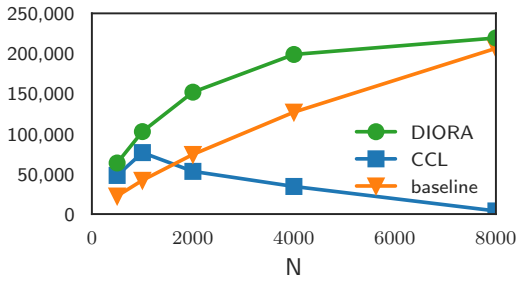


Figure C.2: Progression of the average GDL of the induced grammars ($V27L10$) compared to the baseline grammar of one production rule for each message.

D Analysing emergent languages

Here we present a complete overview of the results from analysing the languages in §5. To aid in interpreting the different metrics, we compare these with several baselines. To test for significance, we report the p-values from a *one-sample t-test*, where the baseline value is assumed to be the population mean.

D.1 Baselines

The *shuffled baselines* are constructed by randomly shuffling the messages of the *induction set* for a randomly selected seed, such that they are unique in the shuffled set. We create the *random baselines* by randomly sampling the same number of unique messages as the *induction set*, also for one seed. See Table D.1 for the number of messages used for each baseline per language.

D.2 Description lengths

Tables D.2, D.3, and D.4 give an overview of the description lengths for the induction sets, the evaluation sets, and their ratios, respectively. The description lengths are also visualised in Figures D.1 and D.2.

D.3 Coverage

We show the evaluation and overgeneration coverage in Table D.5.

D.4 Nature of syntactic structure

Table D.6 gives an overview of the total number of unique pre-terminals and terminals in the induced grammars. We show the average number of pre-terminals per terminal in Table D.8 and Figure D.3. The average number of pre-terminals per terminal is one for every language and baseline, and is therefore omitted. The number of pre-terminal groups and the number of non-terminals generating these groups are presented in Table D.7.

D.5 Parse tree distributions

In Figure D.4 we show the parse tree distributions. For the CCL induced $L3$ grammars, we see all depths are 1, while for DIORA all depths are 2. A parse depth of 1 indicates a flat grammar, without hierarchical structure. The depth of 2 reflects the bias of DIORA towards binary trees.

The $L5$, and especially $L10$, grammars show deeper trees, often the maximum tree depth, which would mean binary skewed trees. DIORA shows more variation in the tree depth distributions.

L	V	GDL emergent	random	shuffled	structured	average DDL emergent	random	shuffled	structured
3	6	28±0.0	28	28	52*	11.2±0.0	11.2	11.2	7.8*
	13	74±10	67	67	1.0E02*	15.7±0.1	16.0*	15.7	10.5*
	27	1.6E02±12	1.3E02	1.5E02	2.1E02*	18.9±0.4	19.2	18.2	15.2*
5	6	2.0E02±18	62*	1.8E02	97*	19.6±0.0	18.6*	19.7	6.8*
	13	1.9E03±1.1E03	1.7E02	6.1E02	1.6E02	27.1±0.9	26.4	26.6	12.3*
	27	1.0E03±8.3E02	1.1E02	4.3E02	4.0E02	33.3±2.0	31.2	34.0	20.3*
10	6	3.6E04±9.4E03	2.9E05*	7.2E04*	1.3E02*	32.7±0.9	23.5*	35.6*	7.8*
	13	9.3E04±1.6E04	1.6E06*	5.6E05*	2.9E02*	34.6±0.6	18.8*	38.0*	7.8*
	27	1.4E05±1.8E04	2.2E06*	1.4E06*	9.8E02*	34.5±1.2	18.9*	22.3*	23.8*

(a) CCL-BMM

L	V	GDL emergent	random	shuffled	structured	average DDL emergent	random	shuffled	structured
3	6	61±18	62	42	62	12.3±0.8	12.5	11.9	7.4*
	13	1.9E02±38	1.3E02	2.0E02	1.5E02	17.4±0.3	17.3	16.6	11.6*
	27	2.5E02±85	1.8E02	1.9E02	3.0E02	20.1±0.2	19.3*	20.0	16.5*
5	6	2.9E02±1.8E02	1.9E02	4.8E02	1.2E02	29.5±0.7	20.1	20.9	7.4*
	13	1.2E03±1.7E02	7.7E02	1.4E03	4.0E02*	28.5±0.7	27.1	29.2	13.8*
	27	2.3E03±6.4E02	1.4E03	3.6E03	5.0E02	30.8±1.0	35.0*	32.0	20.4*
10	6	2.9E04±3.1E03	2.9E05*	9.1E04*	1.3E02*	35.0±0.7	23.5*	36.4	14.5*
	13	2.6E05±3.3E04	1.6E06*	7.2E05*	3.9E02*	33.6±1.3	18.8*	29.3*	7.8*
	27	2.9E05±4.3E04	1.6E06*	1.3E06*	2.9E03*	33.5±0.8	18.9*	20.2*	23.0*

(b) DIORA-BMM

Table D.2: Description Lengths (GDL and average DDL) for the induced grammars and their baselines. We indicate significant differences with the baseline value at $p < .05$ with an asterisk (*).

L	V	emergent	structured
3	6	11.2±0.0	8.2*
	13	15.9±0.0	10.8*
	27	19.1±0.5	15.2*
5	6	19.6±0.1	7.7*
	13	27.0±0.9	12.4*
	27	33.6±1.8	20.3*
10	6	32.9±0.9	8.0*
	13	35.3±0.5	8.4*
	27	35.6±1.3	23.8*

(a) CCL-BMM

L	V	emergent	structured
3	6	12.4±0.8	7.6*
	13	17.6±0.3	11.9*
	27	20.3±0.2	16.5*
5	6	19.5±0.7	7.7*
	13	28.4±0.6	13.7*
	27	30.8±1.2	20.5*
10	6	35.0±0.6	13.7*
	13	35.7±1.4	8.5*
	27	35.9±1.4	23.1*

(b) DIORA-BMM

Table D.3: Average data description lengths on the evaluation set (average evaluation DDL) for the grammars induced on the languages and their structured baselines. We indicate significant differences with the baseline value at $p < .05$ with an asterisk (*).

L	V	DDL:GDL emergent	random	shuffled	structured	evaluation DDL:GDL emergent	structured
3	6	60.87	60.99	59.66	1.83	7.07	0.64
	13	85.67	94.42	83.70	12.99	9.69	3.32
	27	65.60	84.57	64.42	81.56	7.41	20.51
5	6	83.09	249.42	98.04	1.33	9.29	0.40
	13	44.72	249.45	79.70	23.92	4.98	6.05
	27	115.86	494.86	156.57	102.85	12.99	25.71
10	6	4.39	0.37	2.10	1.12	0.49	0.30
	13	3.15	0.10	0.56	0.68	0.36	0.20
	27	2.18	0.08	0.15	48.43	0.25	12.12

(a) CCL-BMM

L	V	DDL:GDL emergent	random	shuffled	structured	evaluation DDL:GDL emergent	structured
3	6	33.50	30.41	41.29	1.43	3.91	0.49
	13	38.57	51.90	29.18	9.94	4.37	2.55
	27	49.52	60.86	52.72	64.78	5.57	16.29
5	6	127.10	86.48	39.66	1.21	14.22	0.33
	13	39.15	56.20	38.28	10.52	4.35	2.63
	27	25.38	43.59	17.63	81.71	2.83	20.49
10	6	5.54	0.37	1.70	2.09	0.62	0.52
	13	1.10	0.10	0.34	0.50	0.13	0.15
	27	1.06	0.11	0.15	15.79	0.13	3.96

(b) DIORA-BMM

Table D.4: An overview of the ratios of DDL:GDL and *evaluation DDL*:GDL for all the languages and their baselines.

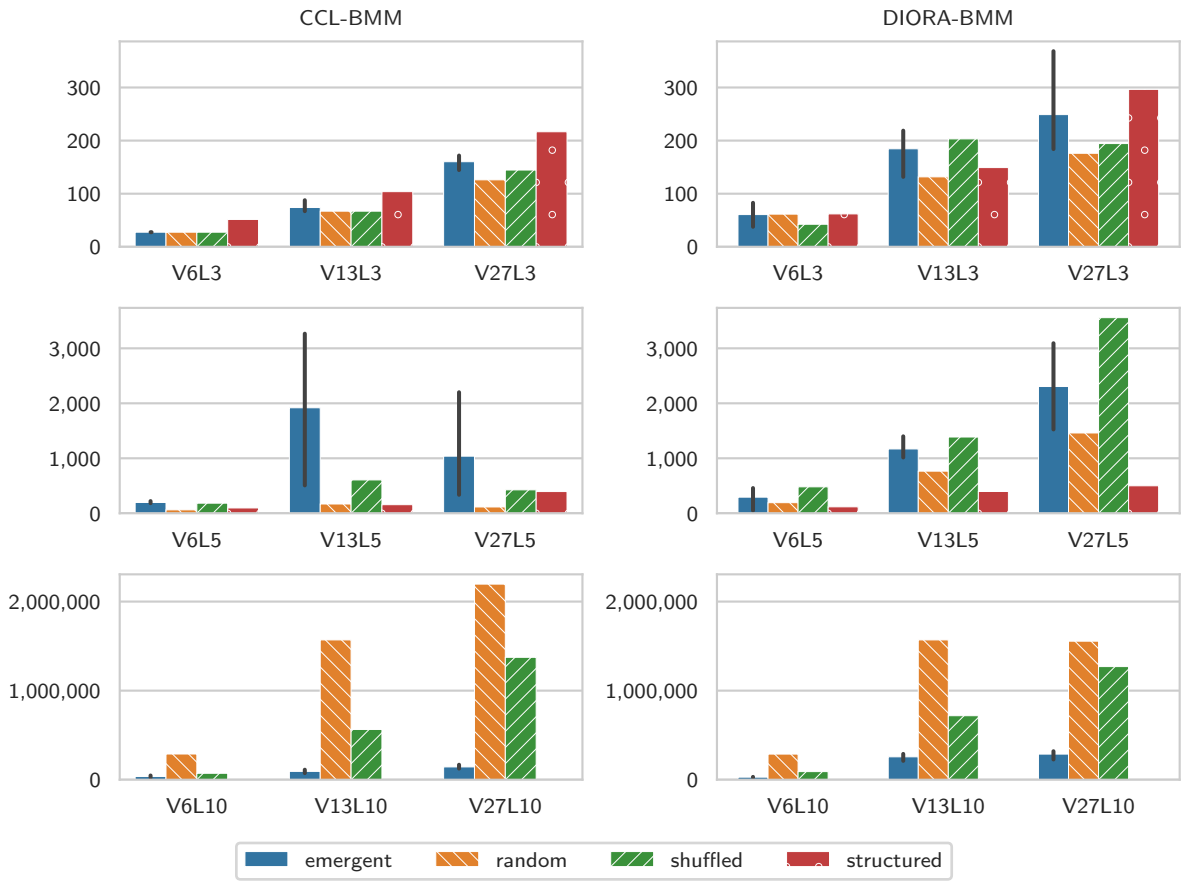


Figure D.1: Overview of the grammar description lengths (GDL) for the induced grammars. Note that for $L = 10$ the structured baseline GDL is too small to be visible in the chart.

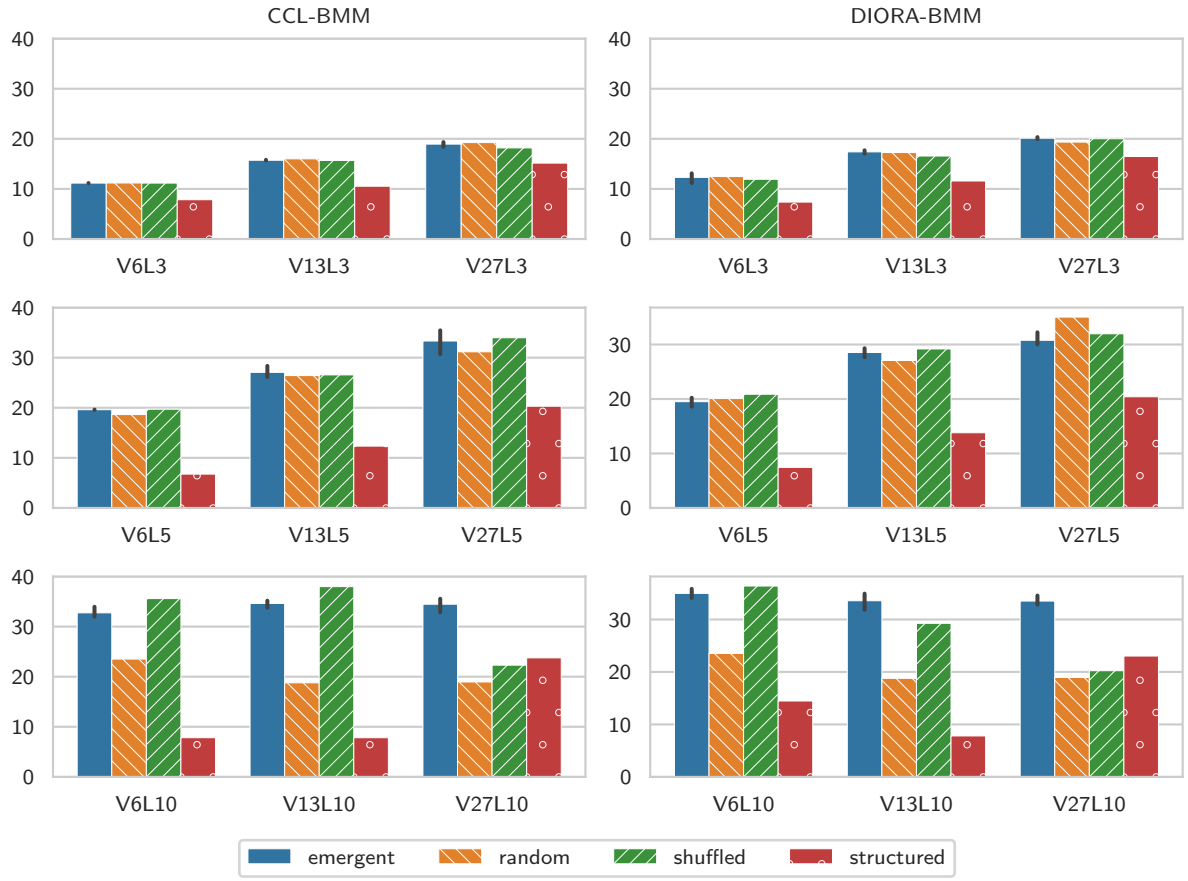


Figure D.2: Overview of the data description lengths (DDL) for the induced grammars.

L	V	evaluation (%) emergent	evaluation (%) structured	overgeneration (%) emergent	overgeneration (%) random	overgeneration (%) shuffled	overgeneration (%) structured
3	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
5	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
10	6	100	100	78±2	0	94	0
	13	98±1	100	3±1	0	13	0
	27	96±1	100	1±1	0	0	0

(a) CCL-BMM

L	V	evaluation (%) emergent	evaluation (%) structured	overgeneration (%) emergent	overgeneration (%) random	overgeneration (%) shuffled	overgeneration (%) structured
3	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
5	6	100	100	100	100	100	0
	13	100	100	100	100	100	0
	27	100	100	100	100	100	0
10	6	100	100	98±2	0	100	0
	13	96±3	100	12±10	0	2	0
	27	92±3	100	0	0	0	0

(b) DIORA-BMM

Table D.5: Average evaluation and overgeneration coverage for the induced grammars. Standard deviations of < 0.5 for the emergent languages are left out.

L	V	number of preterminals emergent	random	shuffled	structured	number of terminals emergent	random	shuffled	structured
3	6	1.0±0.0	1.0	1.0	3.0	6.0±0.0	6.0	6.0	6.0
	13	1.0±0.0	1.0	1.0	3.0	13.0±0.0	13.0	13.0	13.0
	27	1.0±0.0	1.0	1.0	3.0	22.7±1.2	22.0	21.0	27.0
5	6	2.0±0.0	1.0	2.0	4.0	6.0±0.0	6.0	6.0	6.0
	13	3.3±1.7	1.0	2.0	5.0	12.3±0.5	12.0	13.0	13.0
	27	2.0±1.4	1.0	1.0	6.0	20.0±2.2	20.0	21.0	27.0
10	6	6.0±0.0	6.0	6.0	4.0	6.0±0.0	6.0	6.0	6.0
	13	12.7±0.5	13.0	12.0	10.0	13.0±0.0	13.0	13.0	13.0
	27	19.7±2.1	21.0	18.0	12.0	21.0±2.2	21.0	18.0	27.0

(a) CCL-BMM

L	V	number of preterminals emergent	random	shuffled	structured	number of terminals emergent	random	shuffled	structured
3	6	1.0±0.0	1.0	1.0	3.0	6.0±0.0	6.0	6.0	6.0
	13	1.7±0.5	1.0	2.0	3.0	13.0±0.0	13.0	13.0	13.0
	27	1.3±0.5	1.0	1.0	3.0	22.7±1.2	22.0	21.0	27.0
5	6	2.3±0.9	2.0	3.0	4.0	6.0±0.0	6.0	6.0	6.0
	13	2.7±0.5	2.0	3.0	5.0	12.3±0.5	12.0	13.0	13.0
	27	4.0±0.8	1.0	4.0	6.0	20.0±2.2	20.0	21.0	27.0
10	6	6.0±0.0	6.0	6.0	4.0	6.0±0.0	6.0	6.0	6.0
	13	13.0±0.0	13.0	13.0	10.0	13.0±0.0	13.0	13.0	13.0
	27	20.0±1.6	21.0	18.0	17.0	21.0±2.2	21.0	18.0	27.0

(b) DIORA-BMM

Table D.6: Average number of pre-terminals and terminals per grammar.

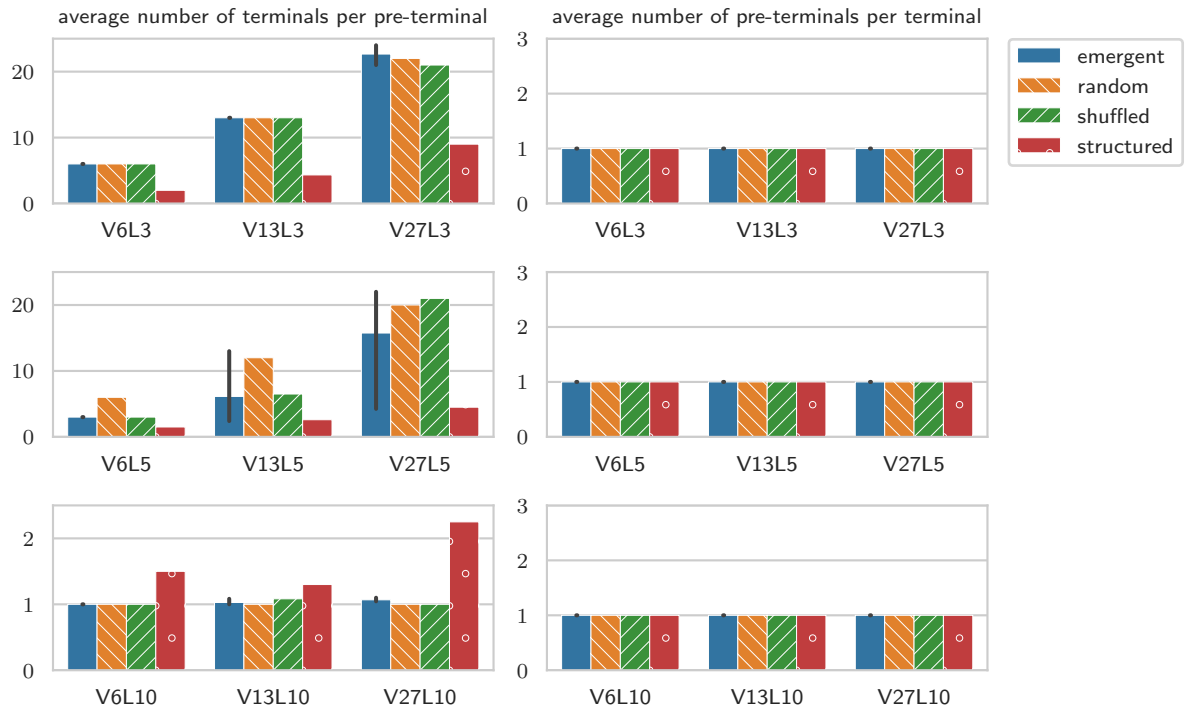
L	V	number of pre-terminal group-generating non-terminals emergent	random	shuffled	structured	number of pre-terminal groups emergent	random	shuffled	structured
3	6	1.0±0.0	1.0	1.0	2.0*	1.0±0.0	1.0	1.0	3.0*
	13	1.3±0.5	1.0	1.0	1.0	1.3±0.5	1.0	1.0	1.0
	27	2.0±0.0	1.0*	2.0	1.0*	2.0±0.0	1.0*	2.0	1.0*
5	6	1.0±0.0	1.0	1.0	1.0	4.0±0.0	1.0*	4.0	2.0*
	13	7.3±1.2	4.0	8.0	2.0*	24.3±16.3	4.0	10.0	2.0
	27	7.3±2.1	1.0*	6.0	3.0	14.7±15.1	1.0	4.0	3.0
10	6	14.0±6.4	36.0*	1.0	1.0	34.7±0.9	36.0	36.0	2.0*
	13	46.3±9.7	169.0*	16.0*	2.0*	78.7±6.0	169.0*	137.0*	2.0*
	27	64.0±15.0	441.0*	233.0*	2.0*	192.3±64.8	441.0*	262.0	2.0

(a) CCL-BMM

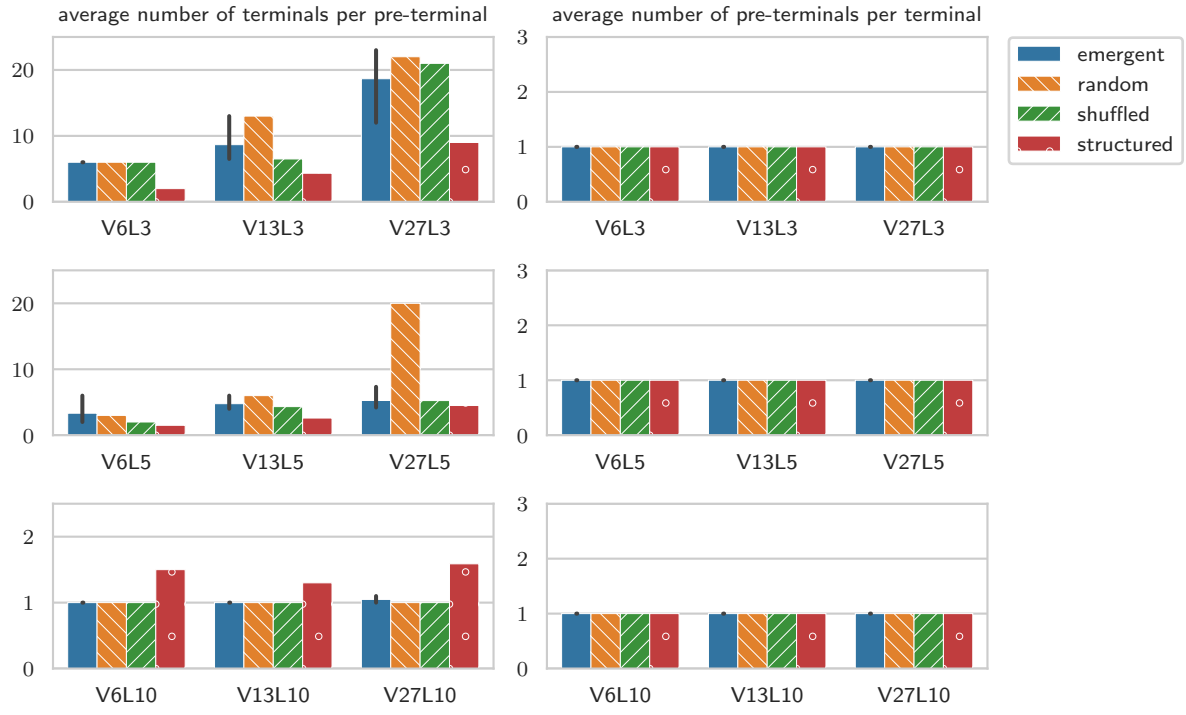
L	V	number of pre-terminal group-generating non-terminals emergent	random	shuffled	structured	number of pre-terminal groups emergent	random	shuffled	structured
3	6	2.0±0.8	2.0	1.0	3.0	1.0±0.0	1.0	1.0	3.0*
	13	3.0±0.0	3.0	3.0	3.0	3.0±1.4	1.0	4.0	3.0
	27	3.0±0.8	2.0	3.0	4.0	2.0±1.4	1.0	1.0	3.0
5	6	1.0±0.0	1.0	1.0	1.0	6.3±3.8	4.0	9.0	2.0
	13	2.3±0.5	3.0	2.0	4.0*	7.3±2.4	4.0	9.0	4.0
	27	3.7±1.2	15.0*	17.0*	4.0	16.7±6.5	1.0	16.0	6.0
10	6	11.3±6.3	36.0*	3.0	2.0	35.3±0.5	36.0	36.0	2.0*
	13	71.0±9.2	169.0*	117.0*	3.0*	130.7±9.0	169.0*	144.0	3.0*
	27	96.0±14.2	437.0*	262.0*	13.0*	225.3±21.8	437.0*	265.0	15.0*

(b) DIORA-BMM

Table D.7: Average number of pre-terminal groups and their generating non-terminals. The right-hand side of a production rule leading only to pre-terminals or symbols, constitutes a *pre-terminal group*, while the non-terminal on the left-hand side is the respective *pre-terminal group-generating non-terminals*. We indicate significant differences with the baseline value at $p < .05$ with an asterisk (*).

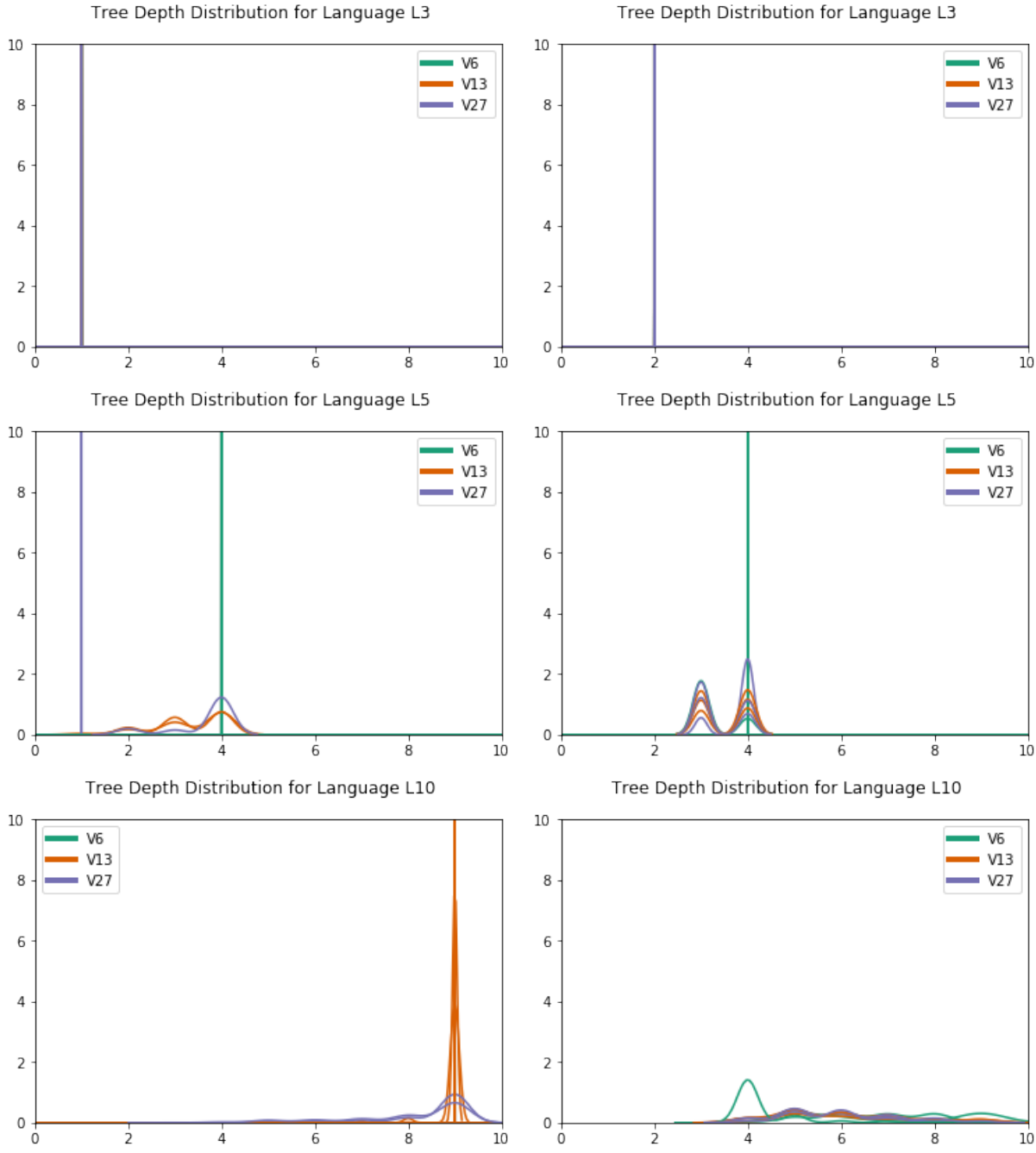


(a) CCL-BMM



(b) DIORA-BMM

Figure D.3: An overview of the average number of *terminals per pre-terminal* and the average number of *pre-terminals per terminal*.



(a) CCL-BMM

(b) DIORA-BMM

Figure D.4: Visualisations of the *parse tree depth* distributions for the most probable parses of the evaluation messages for all emergent languages.

L	V	average # terminals/pre-terminal emergent	random	shuffled	structured
3	6	6.0±0.0	6.0	6.0	2.0*
	13	13.0±0.0	13.0	13.0	4.3*
	27	22.7±1.2	22.0	21.0	9.0*
5	6	3.0±0.0	6.0*	3.0	1.5*
	13	6.1±4.9	12.0	6.5	2.6
	27	15.8±8.1	20.0	21.0	4.5
10	6	1.0±0.0	1.0	1.0	1.5*
	13	1.0±0.0	1.0	1.1	1.3*
	27	1.1±0.0	1.0	1.0	2.2*

(a) CCL-BMM

L	V	average # terminals/pre-terminal emergent	random	shuffled	structured
3	6	6.0±0.0	6.0	6.0	2.0*
	13	8.7±3.1	13.0	6.5	4.3
	27	18.7±4.8	22.0	21.0	9.0
5	6	3.3±1.9	3.0	2.0	1.5
	13	4.8±0.9	6.0	4.3	2.6
	27	5.3±1.5	20.0*	5.2	4.5
10	6	1.0±0.0	1.0	1.0	1.5*
	13	1.0±0.0	1.0	1.0	1.3*
	27	1.0±0.0	1.0	1.0	1.6*

(b) DIORA-BMM

Table D.8: Average number of terminals per pre-terminals. We indicate significant differences with the baseline value at $p < .05$ with an asterisk (*).