Which Word Orders Facilitate Length Generalization in LMs? An Investigation with GCG-Based Artificial Languages

Nadine El-Naggar[*] Tatsuki Kuribayashi[*] Ted Briscoe
Mohamed bin Zayed University of Artificial Intelligence
{nadine.naggar, tatsuki.kuribayashi, ted.briscoe}@mbzuai.ac.ae

Abstract

Whether language models (LMs) have inductive biases that favor typologically frequent grammatical properties over rare, implausible ones has been investigated, typically using artificial languages (ALs) (White and Cotterell, 2021; Kuribayashi et al., 2024). In this paper, we extend these works from two perspectives. First, we extend their context-free AL formalization by adopting Generalized Categorial Grammar (GCG) (Wood, 2014), which allows ALs to cover attested but previously overlooked constructions, such as unbounded dependency and mildly context-sensitive structures. Second, our evaluation focuses more on the generalization ability of LMs to process unseen longer test sentences. Thus, our ALs better capture features of natural languages and our experimental paradigm leads to clearer conclusions — typologically plausible word orders tend to be easier for LMs to productively generalize.

1 Introduction

Attested natural languages (NLs) possess different grammatical properties, such as different word orders. This naturally raises a question about what kind of language is easier for language models (LMs) to learn (Cotterell et al., 2018; Mielke et al., 2019; White and Cotterell, 2021; Borenstein et al., 2024; Arnett and Bergen, 2025). This question has even been extended to counterfactual, impossible languages (Mitchell and Bowers, 2020; Kallini et al., 2024; Kuribayashi et al., 2024). Two related additional questions are why are some combinations of features typologically common and others rare (Dryer and Haspelmath, 2013), and what role if any can LMs play in exploring such questions (Chomsky et al., 2023).

To answer these questions, we need to understand how we can adequately measure the inductive bias of LMs over specific grammatical properties? There are at least two challenges from both data and evaluation metric perspectives. On the data side, NLs differ across a variety of dimensions, and thus isolating a specific grammatical factor for evaluation is challenging with NL data (Mielke et al., 2019). The use of artificial languages (ALs), instead, is a promising direction to enable more controlled experimental setups (White and Cotterell, 2021), but ALs are often highly simplified and lack critical properties underlying NLs, such as context-free Dyck languages. On the evaluation metric side, LM performance is often measured with perplexity (PPL) on the held-out dataset sampled from the same distribution (domain) as the training data.[1] An additional important aspect to be evaluated in language learning is, however, the ability to productively generalize to longer sentences from shorter stimuli, generally motivated by the argument of "infinite use of finite means".

In this paper, we advance this line of research on both data and evaluation sides. For the data, we introduce an extensible approach to defining ALs, based on Generalized Categorial Grammars (GCGs) (Wood, 2014). Our framework can support the inclusion of mildly context-sensitive (indexed language) constructions, such as cross-serial dependencies, and a general approach to unbounded filler-gap dependencies, while maintaining diverse naturalistic constructions. We exemplify this by extending the set of ALs in White and Cotterell (2021) to include object relative clauses as one exemplar of unbounded dependencies.

For the evaluation metric, we target the generalization of LMs from shorter exposures during training to a longer test set. That is, we train LMs on a set of shorter AL sentences and then evaluate their performance on the unseen, longer AL sentences. We further introduce several evaluation perspec-

[*] Equal contribution.

[1] We use the terms "in-domain" and "out-of-domain" just based on the length of the dataset in this study, while these are often relevant to more semantic differences of the data.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 35599–35613
November 4-9, 2025 ©2025 Association for Computational Linguistics

tives on the generalization test set, including PPLs on specific challenging constructions that require proper syntactic generalization, such as unbounded dependencies, as well as grammatical judgment accuracy rather than holistic PPL scores.

In our experiments, following White and Cotterell (2021) and Kuribayashi et al. (2024), we repeatedly evaluate LM's generalization ability, using different ALs with different word order configurations. Here, we try to answer the question of which word order configurations facilitate LMs to better perform generalization to longer sentences and accurately make grammaticality judgments. In particular, we are interested in whether typologically plausible word orders make it easier for LMs to perform productive linguistic generalization.

Our experimental results offer several novel findings. First, out-of-domain evaluation with longer sentences makes LMs' inductive bias clearer, compared to preference across different word orders on in-domain (same length as training data) evaluation. Second, stronger correlations between LM performance and typological distributions emerge once the scope is extended from in-domain to generalization based evaluation. That is, typologically plausible word order tends to be easier for LMs to perform *generalization* to longer sentences, rather than just fitting to in-domain data. Third, RNN's performance is overall better aligned with typological plausibility throughout our three experiments, than other architectures, such as Transformers, which supports that working memory constraints shape typologically frequent word orders in natural language (Hawkins, 1994; Futrell et al., 2020; Hahn et al., 2021).

2 Background

2.1 Artificial Language Learning

ALs are often used in targeted evaluation of LMs. One line of research uses ALs to assess whether LMs can learn patterns corresponding to different levels of the Chomsky Hierarchy. Someya et al. (2024), for instance, test whether LMs can learn regular, context-free, and context-sensitive languages, specifically those involving nested, long-distance, and cross-serial dependencies. Additional studies use context-free and mildly context-sensitive languages, like Dyck languages and $a^n b^n c^n$, to test how well LMs generalize to longer sequences (Suzgun et al., 2019; Weiss et al., 2018; El-Naggar et al., 2022), and explore how different LM architectures correspond to various levels of the Chomsky Hierarchy (Delétang et al., 2022). However, a key limitation is that many of these ALs are far removed from natural language, involving highly simplified vocabularies, unrealistic degrees of (self-)embedding and limited constructional variety.

Another area of research builds on the claim by Chomsky et al. (2023) that neural LMs can learn both possible and impossible human languages, making them unable to distinguish between the two. Kallini et al. (2024) constructed typologically impossible ALs by systematically permuting and modifying an English dataset, following Ravfogel et al. (2019). Their experiments show that GPT-2 models struggle to learn these impossible ALs, which is inconsistent with the claims by Chomsky et al. (2023). Still, it is difficult to pinpoint exactly which linguistic features make language learning more challenging, due to the complex, multidimensional nature of the modified input.

Inspired by Ravfogel et al. (2018), White and Cotterell (2021) use ALs generated by a probabilistic context-free grammar (PCFG) to study the inductive biases of LMs towards particular word orders. By defining six structural parameters that reverse the order of constituents in different syntactic rules, they generate a range of word order configurations and evaluate LSTM and Transformer performance across them. Kuribayashi et al. (2024) extend this work by evaluating cognitively inspired LMs on the same ALs. However, due to the constraints of the PCFGs used, their ALs do not include several attested grammars or constructions, such as Verb-Subject-Object (VSO) word order, and mildly context-sensitive constructions. They also do not test LMs' generalization to longer sentences.

Concurrent works have also explored other formulations of ALs; for example, Xu et al. (2025) explored dependency-based corpus modification, Hunter (2025) proposed ALs with constituency-based non-adjacency, and Yang et al. (2025) used multiple languages as a seed NLs to develop ALs. Our previous work introduced GCG-based ALs, but did not test length generalization and focused on the replication and extension of existing PPL-based studies (El-Naggar et al., 2025).

2.2 Generalization to Longer Sentences

Human language possesses the property of productive and systematic compositionality, where new sentences are formed from known basic components (Montague, 1970; Chomsky, 1957). Humans are able to produce and comprehend an open-ended number of sentences from early limited exposure to short ones. This indicates that humans are able to generalize during learning from short (in-domain) sentences to longer (out-of-domain) sentences. There is a long-standing debate on whether neural network (NN) models can generalize productively and systematically (Fodor and Pylyshyn, 1988; Baroni, 2020). Notably, the generalization of LMs to complex (potentially longer) sequences has been evaluated in a wide range of tasks, e.g., deductive reasoning (Clark et al., 2021; Saparov et al., 2023), arithmetic reasoning (Kudo et al., 2023), or programming (Dziri et al., 2023).

ALs are often used to evaluate LM's fundamental linguistic competence and their generalization ability to longer sentences, typically with, e.g., Dyck languages and $a^n b^n (c^n)$ (Weiss et al., 2018; Suzgun et al., 2019; El-Naggar et al., 2022, 2023). Weiss et al. (2018), for example, empirically test LSTM, GRU (Cho et al., 2014), and Elman RNN (Elman, 1990) LMs, and LSTMs learn $a^n b^n$ most effectively, but they eventually fail on longer sequences. Similarly, Suzgun et al. (2019) empirically assess the ability of their LSTM models to learn Dyck languages effectively and generalize to longer sequences. However, they do not address whether this behavior is precise enough to generalize to sequences that are significantly longer. El-Naggar et al. (2022) use Dyck languages to evaluate long-term generalization of counting on LSTM, ReLU and GRU models. They use training and test sets of the same size and sequence length as Suzgun et al. (2019), but additionally test their models on significantly longer sequences, and find that their models do not generalize effectively to these very long sequences.

Another commonly used AL for model generalization is SCAN (Lake and Baroni, 2018). They evaluate the models' generalizability to new combinations from familiar components, e.g., from "jump" and "twice" to "jump twice." Still, the mentioned ALs for generalization tests, including $a^n b^n (c^n)$ languages, Dyck languages, and SCAN, do not reflect many of the properties of attested NLs, and may not be adequate to evaluate inductive biases in realistic language learning scenarios.

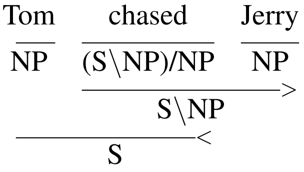
2.3 Categorial Grammar

A categorial grammar (CG) consists of a lexicon that assigns each word a basic or functor category, along with a set of rules that define how functor categories combine with basic categories in both syntax and semantics. Slash notation is used to indicate the direction of the argument relative to the resulting category: for example, α/β denotes a functor that expects a β to its right to form an expression of category α. Classical CG includes just two combinatory rules: **forward functional application** (a) and **backward functional application** (b), as shown below.

(a) $\alpha/\beta\ \beta \Rightarrow \alpha$

(b) $\beta\ \alpha\backslash\beta \Rightarrow \alpha$

Below is an example of forward and backward application using the English transitive verb "chased", which is the functor category $(S\backslash NP)/NP$.



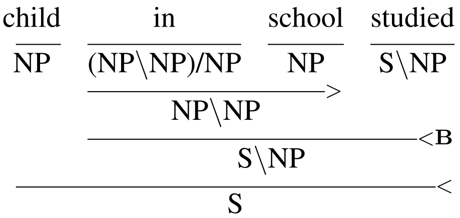
We use English examples to demonstrate rules and derivations. In CG, the majority, if not all, of the variation across languages can be attributed to differences in the lexical categories assigned to words.

CG, which has the expressive power of context-free grammar (CFG), has been extended to combinatory categorial grammar (CCG) (Steedman, 1996), and generalized categorial grammars (GCG) (Wood, 2014) by introducing additional operations to combine categories. One such operation is **composition**, which, like functional application, has forward (a) and backward (b) variants, shown below.

(a) $\alpha/\beta\ \beta/\gamma \Rightarrow \alpha/\gamma$

(b) $\beta\backslash\gamma\ \alpha\backslash\beta \Rightarrow \alpha\backslash\gamma$

Composition (**B**) is demonstrated below.



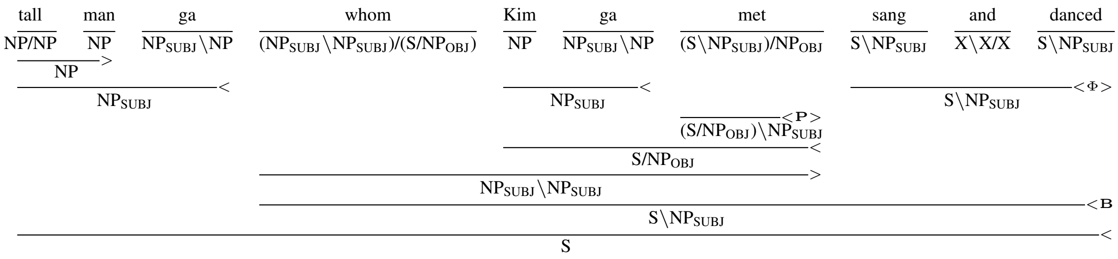
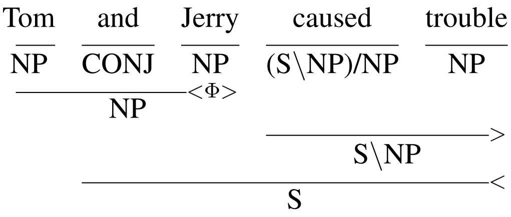


Figure 1: Example of a sentence and its derivation.

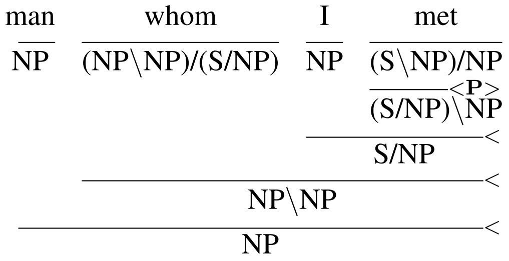
Another operation in GCG is **coordination** (Φ), where 2 constituents with the same categories can be combined into a single one of the same type if they are separated by a conjunction. This is demonstrated in the example below.



We do not use CCG-style type raising, and instead use **permutation** from GCG due to its greater computational tractability. We use cyclic permutation as defined by Briscoe (1997, 2000), where the arguments to functor categories can be cyclically permuted while maintaining their directionality. Formally:

$$(\alpha|\beta_1)...|\beta_n \Rightarrow (\alpha|\beta_n)|\beta_1$$

Permutation (P) is shown below:



We design our ALs based on the application, composition, coordination, and permutation rules defined above. Figure 1 shows an example of the parse of an English-like AL sentence.

3 Dataset

3.1 Overview

We introduce a new set of ALs designed using our GCG framework, which allows us to create a wider range of ALs that reflect different word orders and long-distance dependencies. We generally replicate the experiments of Kuribayashi et al. (2024) and White and Cotterell (2021) on our new datasets and extend these with generalization tests. Because GCGs can, in principle, generate all syntactic patterns observed in natural languages, our framework offers a more comprehensive framework to evaluate neural LMs. We illustrate this flexibility by extending the dataset of White and Cotterell (2021) to include object relative clauses with potentially unbounded dependencies. Our ALs are parameterized by word order parameters (Table 1), similarly to White and Cotterell (2021). Each binary word order parameter controls the order of components within sets of constructions; for example, parameter S changes the order of subject and verb. By setting these parameters independently and exhaustively, we create a set of ALs that differ in word order rules from each other. All parameters except O follow those used by White and Cotterell (2021). The additional O parameter introduces a subject–object ordering rule, which enables us to cover VSO and OSV word orders that were not represented in the ALs created by White and Cotterell (2021), resulting in 96 distinct ALs.

3.2 Lexicon

We first define 11 GCG lexical syntactic categories as shown in Table 2. The directionality of the slashes in each category will be determined once the word order parameters are set (Table 1). Note that we include subject and object markers, and these consistently adopt postpositional case marking, following White and Cotterell (2021).[2] Our lexicon is the same size as that in White and Cotterell (2021), consisting mostly of English words. To simplify the setting, we currently disregard subject-verb number agreement and trained word-level LMs without subword tokenization; that is, phonological and morphological patterns are not

[2] We changed the case marking system to be word-level (*Taro -ga [whom I met]*) rather than phrase-level (*Taro [whom I met] -ga*) adopted in White and Cotterell (2021). Here, *ga* is a nominal case marker.

Param.	Description	0 (head-final)	1 (head-initial)
S	Order of subject and verb	$\mathrm{VI} \rightarrow \mathrm{S}\backslash\mathrm{NP}_{\mathrm{SUBJ}}$ $\mathrm{VT} \rightarrow (\mathrm{S}\backslash\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{NP}_{\mathrm{OBJ}}$ $\mathrm{VCOMP} \rightarrow (\mathrm{S}\backslash\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{SCOMP}$	$\mathrm{VI} \rightarrow \mathrm{S}/\mathrm{NP}_{\mathrm{SUBJ}}$ $\mathrm{VT} \rightarrow (\mathrm{S}/\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{NP}_{\mathrm{OBJ}}$ $\mathrm{VCOMP} \rightarrow (\mathrm{S}/\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{SCOMP}$
VP	Order of object and verb	$\mathrm{VT} \rightarrow (\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})\backslash\mathrm{NP}_{\mathrm{OBJ}}$ $\mathrm{VCOMP} \rightarrow (\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})\backslash\mathrm{SCOMP}$ $\mathrm{REL} \rightarrow (\mathrm{NP}_{\mathrm{SUBJ}}\mid\mathrm{NP}_{\mathrm{SUBJ}})\mid(\mathrm{S}\backslash\mathrm{NP}_{\mathrm{OBJ}})$	$\mathrm{VT} \rightarrow (\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})/\mathrm{NP}_{\mathrm{OBJ}}$ $\mathrm{VCOMP} \rightarrow (\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})/\mathrm{SCOMP}$ $\mathrm{REL} \rightarrow (\mathrm{NP}_{\mathrm{SUBJ}}\mid\mathrm{NP}_{\mathrm{SUBJ}})\mid(\mathrm{S}/\mathrm{NP}_{\mathrm{OBJ}})$
O	Order of subject and object	Subject occurs before the object	Object occurs before the subject
COMP	Position of complementizer	$\mathrm{COMP} \rightarrow \mathrm{SCOMP}\backslash\mathrm{S}$	$\mathrm{COMP} \rightarrow \mathrm{SCOMP}/\mathrm{S}$
PP	Postposition or preposition	$\mathrm{PREP} \rightarrow (\mathrm{NP}\backslash\mathrm{NP})/\mathrm{NP}$	$\mathrm{PREP} \rightarrow (\mathrm{NP}/\mathrm{NP})\backslash\mathrm{NP}$
ADJ	Order of adjective and noun	$\mathrm{ADJ} \rightarrow \mathrm{NP}/\mathrm{NP}$	$\mathrm{ADJ} \rightarrow \mathrm{NP}\backslash\mathrm{NP}$
REL	Position of relativizer	$\mathrm{REL} \rightarrow (\mathrm{NP}_{\mathrm{SUBJ}}/\mathrm{NP}_{\mathrm{SUBJ}})\backslash(\mathrm{S}\mid\mathrm{NP}_{\mathrm{OBJ}})$	$\mathrm{REL} \rightarrow (\mathrm{NP}_{\mathrm{SUBJ}}\backslash\mathrm{NP}_{\mathrm{SUBJ}})/(\mathrm{S}\mid\mathrm{NP}_{\mathrm{OBJ}})$

Table 1: Binary word order parameters and their corresponding GCG categories. "$\alpha \rightarrow \beta$" indicates α is expanded to β in the GCG derivation. Some expansion rules interact with multiple word order parameters, e.g., $\mathrm{VT} \rightarrow (\mathrm{S}/\mathrm{NP}_{\mathrm{SUBJ}})|\mathrm{NP}_{\mathrm{OBJ}}$, and non-target directionalities are denoted as "|" representing either forward or backward slashes.

GCG Lexical Syntactic Category	Example
Noun Phrase (NP) – NP	**Kim** ga kissed **Sandy** o
Subject Marker – $\mathrm{NP}_{\mathrm{SUBJ}}\backslash\mathrm{NP}$	Kim **ga** kissed Sandy o
Object Marker – $\mathrm{NP}_{\mathrm{OBJ}}\backslash\mathrm{NP}$	Kim ga kissed Sandy **o**
Adjective (ADJ) – $\mathrm{NP}\mid\mathrm{NP}$	**red** car ga ran
Transitive Verb (VT) – $(\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{NP}_{\mathrm{OBJ}}$	Kim ga **kissed** Sandy o
Intransitive Verb (VI) – $\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}}$	red car ga **ran**
Verb with Complement (VCOMP) – $(\mathrm{S}\mid\mathrm{NP}_{\mathrm{SUBJ}})\mid\mathrm{SCOMP}$	Kim ga **believed** that Sandy lied
Complementizer (COMP) – $\mathrm{SCOMP}\mid\mathrm{S}$	Kim ga believed **that** Sandy lied
Preposition (PREP) – $(\mathrm{NP}\mid\mathrm{NP})\mid\mathrm{NP}$	elf **on** shelf ga laughed
Relativizer (REL) – $(\mathrm{NP}_{\mathrm{SUBJ}}\mid\mathrm{NP}_{\mathrm{SUBJ}})\mid(\mathrm{S}\mid\mathrm{NP}_{\mathrm{OBJ}})$	man ga **whom** I ga met laughed
Conjunction – $\mathrm{Var}\backslash\mathrm{Var}/\mathrm{Var}$	Kim **and** Sandy ga ate

Table 2: Lexical syntactic categories, their derivations, and their examples (colored) supplemented with an English sentence as a context. The vertical bars "|" in the GCG lexical syntactic categories represent either forward or backward slashes, determined by word order parameters listed in Table 1.

modeled by our LMs.

3.3 Generating the Datasets

To test the length generalization of LMs, we create three variations of the AL corpus: (i) SHORT with a length of 3–8 words, (ii) MEDIUM with a length of 9–10 words, and (iii) LONG with a length of 11–20 words. Only the SHORT part is used for LM training, and the models are tested in held-out SHORT, MEDIUM, and LONG test sets.

The datasets are generated over several steps:

1. **Set word order parameters.** We generate the AL corpus for each combination of the seven parameters. Each AL is defined by a unique combination of parameter values, such as 0101101 for "English" which corresponds to settings S=0, VP=1, O=0, COMP=1, PP=1, ADJ=0, and REL=1 (see Table 1).

2. **Generate templates.** Once word order parameters are fixed, to ensure coverage of all valid sentences in each AL, we generate all possible sequences of word categories up to a length of 10 (for SHORT and MEDIUM sets). These category sequences are then parsed using a GCG parser configured according to the respective grammar with the lexical syntactic categories as terminal symbols.[3] A sequence of word categories is treated as grammatical if the parser produces at least one derivation with S as a root.

3. **Sample lexicons.** Once grammatical templates with word categories are generated, we

[3] We modify the NLTK CCGChartParser (Bird et al., 2009) by disabling type raising and incorporating the permutation operation described in Briscoe (1997, 2000), which we use to parse our sentence templates.

build sentences by randomly sampling lexicons for each word category. The number of sampled sentences is adjusted based on some policies. In our case, we sampled sentences to form a uniform distribution of sentence length within the training and test data, e.g., 1K of length-3 sentences, 1K of length-4 sentences, ..., 1K of length-8 sentences. An example of a valid sentence parse is illustrated in Figure 1.

4. **Augument LONG set.** Using the existing templates of lengths 3-10 words (SHORT and MEDIUM), we create the templates for the LONG test set, where the template lengths are 11-20 words. We extend the existing templates in 3 different ways:
   (a) **Concatenation:** 2 valid templates are concatenated end to end.
   (b) **Mid-sentence insertion with a conjunction:** A conjunction and the second template are inserted at different points in the first template.
   (c) **Appending with a conjunction:** Appending a template to another template using a conjunction.

   We filter the valid extended templates by parsing them using the GCG parsers for all 96 ALs, as previously done for the templates of length 3-10.

5. **Sample lexicons for templates of LONG set.** We randomly sample 20,000 unique valid templates for each of the 96 ALs and, for each template, we sample one sentence from the lexicon. For each AL, we end up with 20,000 unique sentences of length 11-20.

Appendix A shows further details on the GCG parser configuration, as well as the statistics of the data we generated, including the template numbers for each length.

4 Experimental Settings

Models. We evaluate three variants of neural LMs: simple RNN (Elman, 1990), LSTM (Hochreiter and Schmidhuber, 1997), and Transformer (Vaswani et al., 2017). These models are trained using the Fairseq toolkit (Ott et al., 2019). We quantify their inductive bias on what kind of word order they are good at for productive generalization. See Appendix B.1 for more details on the models.

Training. The training set consists of 80K sentences of lengths 3-8 words (SHORT training set). The sentence length is equally distributed, and in each length, templates are also uniformly sampled.[4] We stop the training based on an early-stopping criterion with a patience of five epochs (i.e., the training stops when validation loss does not decrease in five consecutive epochs) and a maximum of 10,000 update steps.

Evaluation. We use different evaluation metrics in different experiments, but they are all focused on generalization for longer sentences (≥ 9; MEDIUM and LONG sets) than those in the training data. We trained three LMs with different seeds for model initialization, and reported scores are the average of three runs.

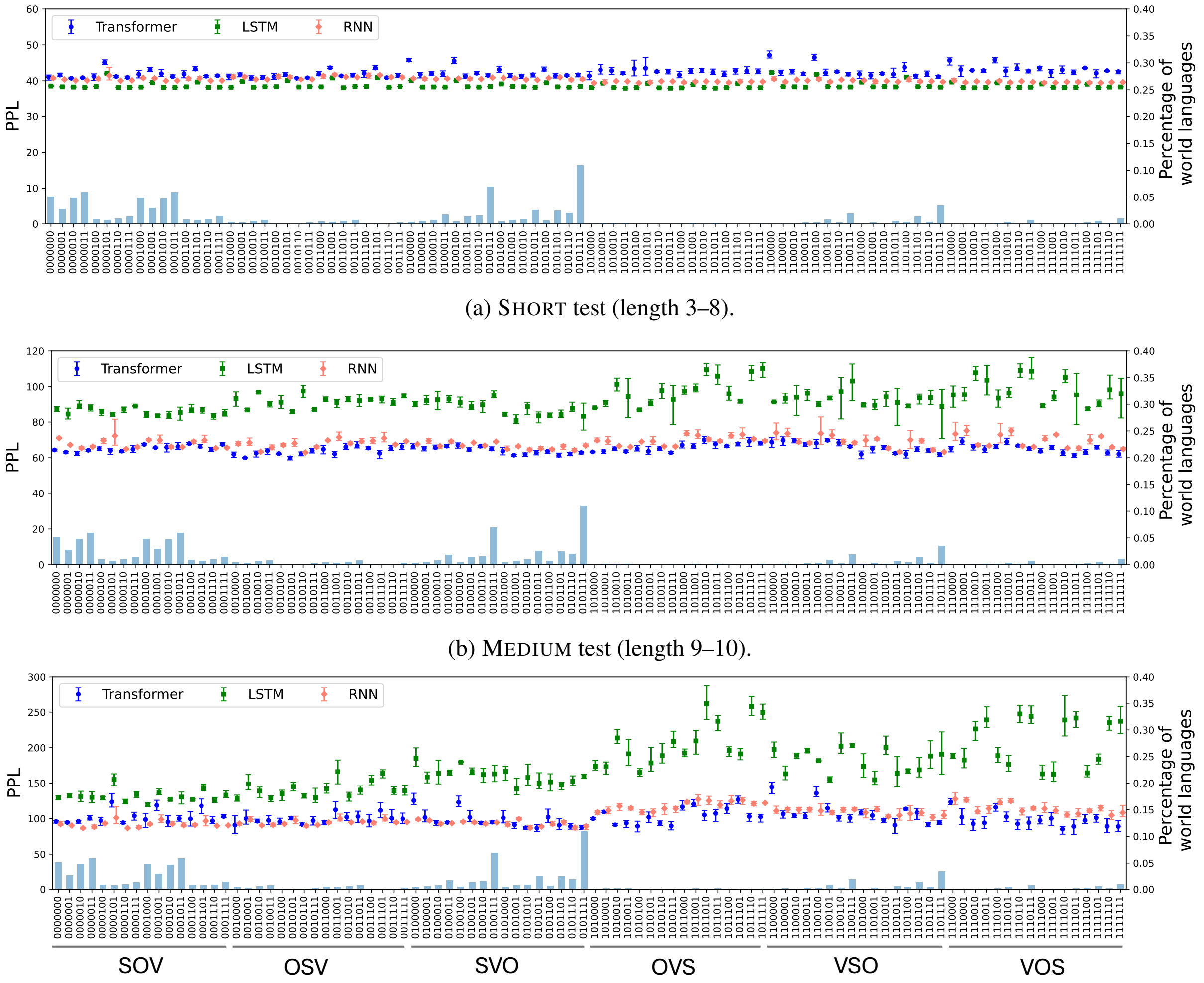
Typological alignment (TA). We measure perplexity (PPL), the geometric mean of word probabilities across sequences, in each language. That is, we obtained the PPL distribution over the 96 languages we used. Following Kuribayashi et al. (2024), we report Pearson's correlation coefficients between PPL and the percentage of respective word order in the world. The typological distribution is based on the percentage of languages that adopt the respective word order estimated with WALS (Dryer and Haspelmath, 2013) and Grambank (Skirgård et al., 2023).[5] Lower negative correlation indicates that learnability is better aligned with typological commonality, i.e., the more common the word order is, the more easily the model learns and productively generalizes it. Appendix B.2 includes details on these databases.

5 Experiment 1: PPLs

Evaluation settings. We first measure the PPLs on MEDIUM and LONG (out-of-domain) test sets with 20K longer sentences for each set. For a comparison, we also measure PPLs on the 20K SHORT (in-domain) set with the same length distribution, i.e., length of 3–8, as training data. We ensured that there is no overlap between each test set and training set, and all the vocabulary in

[4] Longer sentences have a larger number of grammatically possible templates; thus, this uniform sampling automatically introduces the tendency that shorter templates are more frequently selected.

[5] We basically used the WALS database (Dryer and Haspelmath, 2013) to count the frequency of word orders, following the same procedure as (Kuribayashi et al., 2024), and the COMP statistics are supplemented with Grambank (Skirgård et al., 2023) as the COMP statistics are not recorded in WALS.



(a) SHORT test (length 3–8).

(b) MEDIUM test (length 9–10).

(c) LONG test (length 11–20).

Figure 2: Distributions of perplexities and typological plausibility across languages. The error bars indicate max and min PPLs within three runs.

the test sets is used at least once in the training set.

Results. Figure 2 illustrates the PPL and typological distributions, and Table 3 summarizes the results. First, **the PPLs in longer test sets have larger variance** (Figure 2), and thus are more informative about which word order is easier to learn and generalize for a particular LM. The flat PPL distribution in SHORT set, particularly for LSTMs, was also reported in White and Cotterell (2021). Second, Table 3 shows that **the TA score and PPLs for each word order group substantially change between in-domain (SHORT) and out-of-domain (MEDIUM and LONG) evaluation.** In particular, the TA scores in the out-of-domain evaluation are consistently negative, while not in the in-domain evaluation.[6] These suggest that typologically plausible word orders tend to be easier for LMs to productively generalize, in contrast to just fitting to the in-domain data. Third, we also have some architecture-dependent differences between in-domain and out-of-domain PPLs; specifically, **the TA scores for LSTM and RNN drastically improved in the out-of-domain evaluation, and RNN with LONG test set achieved the best (lowest) correlation in all settings**. In contrast, Transformer yielded a good correlation only with the in-domain test, which diminished in out-of-domain tests. The out-of-domain results are somewhat intuitive if one believes that typologically common patterns are a consequence of human limited working

[6] This negative result in in-domain data seems to contradict Kuribayashi et al. (2024), where length generalization is not considered, but the advantage of working memory limitation is reported. This might be due to the difference in the length distribution of the training data (i.e., our study uses much shorter sentences than them), and we clarify that their results do not always hold if the training/evaluation domain is limited.

	SHORT							MEDIUM							LONG						
Model	SOV	OSV	SVO	OVS	VSO	VOS	TA ↓	SOV	OSV	SVO	OVS	VSO	VOS	TA ↓	SOV	OSV	SVO	OVS	VSO	VOS	TA ↓
Transformer (PPL ↓)	41.8	41.6	42.3	42.6	42.7	43.3	−**27.7**†	65.2	63.5	64.2	65.9	66.1	65.0	−**10.4**	102.3	99.4	97.9	104.0	107.6	97.9	−**19.2**
LSTM (PPL ↓)	38.7	38.8	38.7	38.4	39.1	38.5	−**14.2**	85.9	91.7	88.0	97.5	92.9	97.9	−**31.0**†	131.9	141.5	160.7	205.5	180.9	207.5	−**33.4**†
RNN (PPL ↓)	40.4	41.0	40.6	39.7	40.1	39.7	13.0	67.8	67.9	66.7	69.6	69.0	69.4	−**17.4**	91.8	94.6	93.2	118.0	109.0	114.2	−**43.1**†
Natural Lang. (Prob. ↑)	0.54	0.04	0.23	0.01	0.12	0.05	-	0.54	0.04	0.23	0.01	0.12	0.05	-	0.54	0.04	0.23	0.01	0.12	0.05	-

Table 3: Average PPLs within each base word order group as well as Pearson's correlation coefficient between PPL and the frequency of respective word order in the world. Negative TA (typological alignment) scores are highlighted in bold. Statistical significance of correlation coefficient ($p<0.05$) is marked with †.

Language	Recursive Relative Clauses	Embedded Relative Clause
0000000	John ga promised which pasta ga nibbles which fruits ga wall o received	John ga pasta ga nibbles that said which fruits ga wall o received
0101101 (English)	fruits ga which pasta ga which John ga promised nibbles received wall o	fruits ga which John ga said that pasta ga nibbles received wall o
1111111	received wall o fruits ga which nibbles pasta ga which promised John ga	received wall o fruits ga which said that nibbles pasta ga John ga

Table 4: Examples in challenging test sets. The examples with the 0101101 word order parameters follow the basic English word order.

Model	RECURSIVE (TA ↓)	EMBEDDED (TA ↓)
Transformer	−5.1	−23.5†
LSTM	9.2	−3.7
RNN	12.9	−18.1†

Table 5: Correlation between PPL in the targeted evaluation set for each language and typological plausibility. Statistical significance of correlation coefficient ($p<0.05$) is marked with †.

memory, and the LSTM and RNN with recurrent model architecture have such cognitively plausible constraints, at least compared to the Transformer architecture.

6 Experiment 2: PPLs in Targeted Generalization Sets

PPLs reported in § 5 are a holistic measure of out-of-domain generalization, given that the data is less focused on specific linguistic phenomena.

Evaluation settings. As a complementary evaluation, we introduce additional challenging out-of-domain test sets that focus on unbounded dependency constructions: (i) recursive relative clauses, where two relative clauses are used in a nested; and (ii) embedded relative clauses, where the relative clause is in another subordinate clause, such as "he said" (Table 4). We refer to these test sets as the RECURSIVE and EMBEDDED test sets, respectively. All the sentences have the same construction as shown in Table 4, and lexicons are randomly sampled, resulting in 500 test sentences. Note that these constructions are successfully regarded as grammatical under our GCG-based framework with the permutation operation, and are not included in the training set as they exceed the length of 8. We report the TA score, i.e., correlation between PPL and typological distribution, on these challenging test sets.

Results. Table 5 shows the TA scores for each challenging set and model. In the RECURSIVE set, the correlations are not statistically significant. The ease of such generalization was not related to the typological plausibility of word order, and possibly LMs simply failed to learn such a complex structure. In the EMBEDDED set, the correlations tend to be negative, and Transformer and RNN exhibited statistically significant correlations. This result in EMBEDDED set is overall consistent with the previous finding that typologically common ALs are easier to generalize for LMs. That is, we found that, when the evaluation is extended to specific complex constructions, the results are somewhat phenomenon-dependent and require further investigation with broader-coverage targeted evaluations.

7 Experiment 3: Grammaticality Judgment Accuracy

Lastly, we also perform grammaticality judgment evaluation as orthogonal to PPL evaluation, following the widely adopted minimal pair grammatical-

Language	Case Type Judgment	Verb Type Judgment
0000000	fluffy soft and intelligent mango ga owl o controls *fluffy soft and intelligent mango o owl o controls	green machine ga escorts which scooter ga walk *green machine ga evolves which scooter ga walk
0101101 (English)	fluffy soft and intelligent mango ga controls owl o *fluffy soft and intelligent mango o controls owl o	scooter ga which green machine ga escorts walk *scooter ga which green machine ga evolves walk
1111111	controls owl o mango fluffy soft and intelligent ga *controls owl o mango fluffy soft and intelligent o	walk scooter ga which escorts machine green ga *walk scooter ga which evolves machine green ga

Table 6: Examples in grammatical judgment tests. Ungrammatical sentences are marked with *. The examples with the 0101101 word order parameters follow the basic English word order.

	Case Type		Verb Type	
Model	Corr.↑	Avg. Acc.	Corr.↑	Avg. Acc.
Transformer	0.14	97.7±1.5	$0.27^\dagger$	81.0±14.7
LSTM	0.03	97.2±1.4	$0.28^\dagger$	85.1±9.6
RNN	$0.21^\dagger$	97.4±1.4	$0.23^\dagger$	77.4±15.5

Table 7: Correlation between accuracy and typological plausibility distribution, along with average and standard deviation of accuracy. Statistical significance of correlation coefficient ($p<0.05$) is marked with †.

ity judgment paradigm (Warstadt et al., 2020).

Evaluation settings. As a case study, we selected two simple test cases: (i) case type accuracy, and (ii) verb type selection accuracy (Table 6). Accuracy is measured based on whether a model could assign a high sentence probability to a grammatical sentence, given a pair of grammatical and ungrammatical ones. 500 sentences are first sampled from the MEDIUM set used in § 5 as out-of-domain grammatical sentences. For each grammatical sentence, in the case type accuracy data, an ungrammatical option is created by replacing a case marker with a grammatically incorrect one (i.e., ga→o or o→ga). In the verb type data, a transitive verb in the original sentence is wrongly replaced with an intransitive verb as an ungrammatical option (e.g., escorts→evolves). The target token to be replaced is randomly selected if there are several candidates in a sentence. Note that, in our tests, the sentence length is aligned between the two options, and thus we simply calculated and compared the accumulated sentence probability $p(s) = \prod_{w_i \in s} p(w_i|\boldsymbol{w}_{<i})$ without any length normalization. The grammatical judgment accuracy is measured in each word order configuration, and the correlation between the accuracies and typological plausibilities over 96 languages is reported (noted as Corr.). This correlation should be positive if typologically common ALs are easier to learn.

Results. Table 7 shows the correlation scores, as well as average accuracy for each setting. **All the correlations are positive, but only the RNN showed a statistically significant correlation in both settings.** These results are in line with the findings in § 5 that typologically frequent word order facilitates grammar acquisition, and a model with limited working memory yields better typological alignment. To sum up all the experiments, the RNN exhibited superior typological alignment in length generalization, at least compared to the Transformer, especially in § 5 and § 7 (and somewhat comparable results in § 6). Given that RNN has the most limited working memory, as it does not have the gate mechanism of the LSTM or attention-based context access of a Transformer, this suggests that working memory limits create inductive bias predicting typological word order distributions.

8 Conclusion

In this paper, we create an AL framework inspired by White and Cotterell (2021) to assess LM inductive biases towards different word orders. We extend their framework from a PCFG to a GCG, and use 96 ALs to evaluate simple RNN, LSTM and Transformer LMs. We calculate perplexity (PPL) on short, medium and long sentences, and observe a moderate alignment between PPL and frequency of word order in attested NLs, particularly in the out-of-domain evaluation with more complex linguistic constructions. Overall, we observe that the performance of recurrent models, especially RNNs, provides good correlation with typological distributions, indicating that they may be the most typologically aligned models that generalize effectively on typologically frequent word order patterns. In contrast, Transformers seem to be the least aligned when evaluated on generalization to longer sentences.

Limitations

While our artificial language (AL) framework provides a controlled environment for evaluating language models (LMs), it does not fully capture the richness and variability of natural languages. The ALs used in this study are simplified and do not, for example, differentiate between verb tenses or include subject-verb agreement. We also do not explore ambiguity, and ensure that each word in the lexicon belongs to exactly one category, unlike in NLs. Future work is needed to systematically investigate a broader range of linguistic phenomena within this framework.

In the future, there are different avenues that we aim to explore. We would like to explore how different training methods can affect model learning and generalization. Another potential future direction to explore is to investigate model learning and behavior when we introduce more features found in NLs, for example, subject-verb number agreement, or lexical ambiguity.

Ethical Statement

The data used in this paper is artificially generated data that is based mostly on English. There is no sensitive information in the data, and no security risks in the contents of this paper. We have no ethical concerns with the contents of this paper.

AI Writing/Coding Assistance Policy

We occasionally used writing assistance systems, i.e., Grammarly and ChatGPT, but these are for the purpose of correcting grammatical/spelling errors and adjusting wording. In other words, our use of AI writing assistance falls under the category (a) Assistance purely with the language of the paper, described in ARR.

References

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375(1791):20190307.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Nadav Borenstein, Anej Svete, Robin Chan, Josef Valvoda, Franz Nowak, Isabelle Augenstein, Eleanor Chodroff, and Ryan Cotterell. 2024. What languages are easy to language-model? a perspective from learning probabilistic regular languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15115–15134, Bangkok, Thailand. Association for Computational Linguistics.

Ted Briscoe. 1997. Co-evolution of language and of the language acquisition device. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 418–427, Madrid, Spain. Association for Computational Linguistics.

Ted Briscoe. 2000. Grammatical acquisition: Inductive bias and coevolution of language and the language acquisition device. *Language*, 76(2):245–296.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam chomsky: The false promise of chatgpt. *The New York Times*, 8.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2 (Short Papers), pages 536–541, New Orleans, Louisiana. Association for Computational Linguistics.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and 1 others. 2022. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*.

Matthew S. Dryer. 2013a. Order of adjective and noun (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer. 2013b. Order of adposition and noun phrase (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer. 2013c. Order of object and verb (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer. 2013d. Order of relative clause and noun (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer. 2013e. Order of subject and verb (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer. 2013f. Order of subject, object and verb (v2020.3). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nadine El-Naggar, Tatsuki Kuribayashi, and Ted Briscoe. 2025. GCG-based artificial languages for evaluating inductive biases of neural language models. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 540–556, Vienna, Austria. Association for Computational Linguistics.

Nadine El-Naggar, Pranava Madhyastha, and Tillman Weyde. 2022. Exploring the long-term generalization of counting behavior in RNNs. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*.

Nadine El-Naggar, Pranava Madhyastha, and Tillman Weyde. 2023. Theoretical conditions and empirical failure of bracket counting on long sequences with linear recurrent networks. *EACL 2023*, page 143.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Richard Futrell, Roger P Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Michael Hahn, Judith Degen, and Richard Futrell. 2021. Modeling word and morpheme order in natural language as an efficient trade-off of memory and surprisal. *Psychological Review*, 128(4):726.

John A Hawkins. 1994. *A performance theory of order and constituency*. Cambridge University Press.

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

Tim Hunter. 2025. Kallini et al.(2024) do not compare impossible languages with constituency-based ones. *Computational Linguistics*, 51(2):641–650.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14691–14714. Association for Computational Linguistics.

Keito Kudo, Yoichi Aoki, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi, and Kentaro Inui. 2023. Do deep neural networks capture compositionality in arithmetic reasoning? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1351–1362.

Tatsuki Kuribayashi, Ryo Ueda, Ryo Yoshida, Yohei Oseki, Ted Briscoe, and Timothy Baldwin. 2024. Emergent word order universals from cognitively-motivated language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14522–14543. Association for Computational Linguistics.

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

S. J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4975–4989. Association for Computational Linguistics.

Jeff Mitchell and Jeffrey Bowers. 2020. Priorless recurrent networks learn curiously. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5147–5158, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of rnns with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3532–3542. Association for Computational Linguistics.

Shauli Ravfogel, Francis M Tyers, and Yoav Goldberg. 2018. Can lstm learn to capture agreement? the case of basque. *arXiv preprint arXiv:1809.04022*.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36:3083–3105.

Hedvig Skirgård, Hannah J. Haynie, Harald Hammarström, Damián E. Blasi, Jeremy Collins, Jay Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Noor Karolin Abbas, Sunny Ananth, Daniel Auer, and 100 others. 2023. Grambank v1.0. Dataset.

Taiga Someya, Ryo Yoshida, and Yohei Oseki. 2024. Targeted syntactic evaluation on the chomsky hierarchy. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 15595–15605. ELRA and ICCL.

Mark Steedman. 1996. Surface structure and interpretation. *(No Title)*.

Mirac Suzgun, Sebastian Gehrmann, Yonatan Belinkov, and Stuart M Shieber. 2019. Lstm networks can perform dynamic counting. *arXiv preprint arXiv:1906.03648*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 740–745. Association for Computational Linguistics.

Jennifer C. White and Ryan Cotterell. 2021. Examining the inductive bias of neural language models with artificial languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 454–463. Association for Computational Linguistics.

Mary McGee Wood. 2014. *Categorial grammars (RLE linguistics b: Grammar)*. Routledge.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. 2025. Can language models learn typologically implausible languages? *arXiv preprint arXiv:2502.12317*.

Xiulin Yang, Tatsuya Aoyama, Yuekun Yao, and Ethan Wilcox. 2025. Anything goes? a crosslinguistic study of (im)possible language learning in LMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26058–26077, Vienna, Austria. Association for Computational Linguistics.

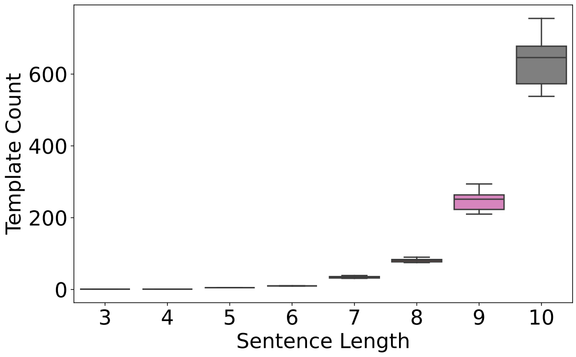


Figure 3: The distribution of the SHORT and MEDIUM template lengths in our ALs (X-axis: template length, Y-axis: template count). The box and error bars present Q1 and Q3 percentiles, respectively.

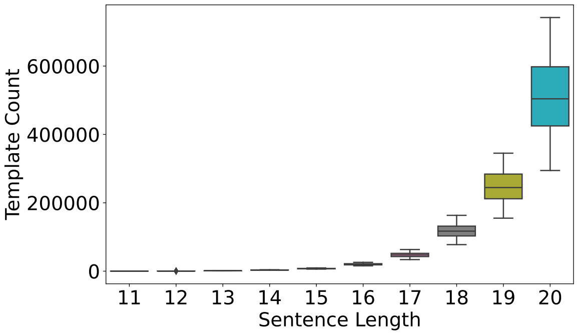


Figure 4: The distribution of all the LONG template lengths in the extended templates created from the SHORT and MEDIUM templates (X-axis: template length, Y-axis: template count). The box and error bars present Q1 and Q3 percentiles, respectively.

A Dataset Details

A.1 Heuristics Applied During SHORT and MEDIUM Template Generation

To improve the efficiency of the template generation process, we apply a set of heuristics to filter out templates that would not produce valid sentences in any of our artificial languages.

We discard templates that meet any of the following criteria:

1. Contain fewer than 3 words (since all grammars require at least 3 words for a valid sentence),
2. Begin with a conjunction,
3. End with a conjunction,
4. Include two consecutive conjunctions,
5. Include two consecutive prepositions,

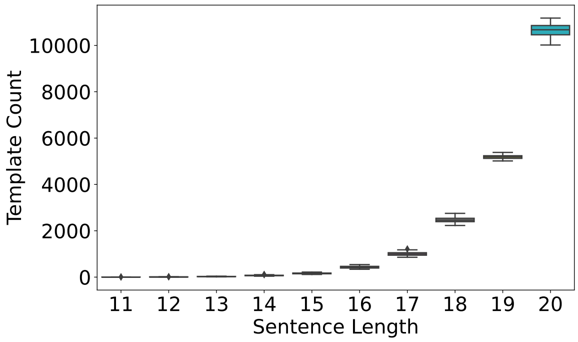


Figure 5: The distribution of the sampled LONG template lengths, which we use to sample the sentences for the LONG test set (X-axis: template length, Y-axis: template count). The box and error bars present Q1 and Q3 percentiles, respectively.

6. Start with subject or object markers,
7. Contain more subject and object markers than NPs,
8. Contain a complementiser without an associated complement verb.

We plot the distribution of the different template lengths across our ALs. We show this in Figure 3 for SHORT and MEDIUM length templates and Figures 4 and 5 for LONG templates. There is a slight variation in the number of templates in each AL, which is attributed to constraints naturally imposed by GCG, e.g., SVO word order can create "S1 V1 and S2 V2 O" as well as "S V1 O1 and V2 O2" structures but SOV word order can only create the "S O1 V1 and O2 V2" structure.

Category	**Type**
S	Primitive
NP	Primitive
VT	(S\NPSUBJ)/NPOBJ
VI	S\NPSUBJ
VCOMP	(S\NPSUBJ)/SCOMP
COMP	SCOMP/S
PREP	(NP/NP)\NP
ADJ	NP/, NP
REL	(NPSUBJ\NPSUBJ)/(S/NPOBJ)
SUBJ	NPSUBJ\NP
OBJ	NPOBJ\NP
CONJ	var\., @var/., @var

Table 8: GCG grammar for the word order consistent with the English language

Artifact	License	Purpose
NLTK (Bird et al., 2009)	Apache License 2.0	to parse sentences in data generation
Fairseq (Ott et al., 2019)	MIT Linense	to train LMs
White and Cotterell (2021) Data	MIT License	to determine dataset configuration
WALS (Dryer and Haspelmath, 2013)	Creative Commons CC-BY 4.0	to find word order statistics in NLs
Grambank (Skirgård et al., 2023)	Creative Commons CC-BY 4.0	to find word order statistics in NLs

Table 9: Details on artifacts we used in this study

S	82A Order of Subject and Verb (Dryer, 2013e)
VP	83A Order of Object and Verb (Dryer, 2013c)
O	81A Order of Subject, Object and Verb (Dryer, 2013f)
COMP	Feature GB421: Is there a preposed complementizer in complements of verbs of thinking and/or knowing? (Skirgård et al., 2023)
	Feature GB422: Is there a postposed complementizer in complements of verbs of thinking and/or knowing? (Skirgård et al., 2023)
PP	85A Order of Adposition and Noun Phrase (Dryer, 2013b)
ADJ	87A Order of Adjective and Noun (Dryer, 2013a)
REL	90A Order of Relative Clause and Noun (Dryer, 2013d)

Table 10: WALS and Grambank chapters we used

A.2 Parser Configuration

To parse templates and assign them to compatible artificial languages (ALs), we adapt the NLTK CCGChartParser (Bird et al., 2009). We disable type raising, an operation available in Combinatory Categorial Grammar (CCG) (Steedman, 1996), and instead implement the permutation rule described by Briscoe (1997, 2000), which is part of Generalized Categorial Grammar (GCG) (Wood, 2014).

The NLTK CCGChartParser allows us to enforce parsing constraints: placing a comma, period, or underscore before a grammar argument disables composition, crossing, or substitution, respectively. We extend this by introducing a new symbol "@" to block permutation.

In our grammar definitions, we limit permutation to categories that function as verb functors (i.e., those involving S). We also constrain subject and object markers so that they only combine with NPs by disabling composition in the definitions of the NP_{SUBJ} and NP_{OBJ} categories.

GCG enables flexible word orders via permutation, which can cause overlap between word orders, for instance, OSV structures appearing in SOV datasets, or VSO in VOS datasets. To maintain clearer distinctions between word orders, we disable permutation for verbs when parsing templates for OSV, SOV, VOS, and OVS languages, except in cases where a relativizer category (REL) is present.

We provide an example of the SVO grammar that corresponds to English in Table 8.

B Information relevant to responsibility checklist

B.1 Model Details

We used exactly the same model hyperparameters as Kuribayashi et al. (2024) for Transformer, LSTM, and RNNs (see Table 11). These models are trained with the Fairseq toolkit (Ott et al., 2019). We did not apply any subword tokenization, in contrast to Kuribayashi et al. (2024), as we disregard morphological number agreement between subjects and verbs. The whole model training and evaluation will be completed with approximately 150 GPU hours.

B.2 Artifacts

Table 9 shows all the artifacts we used, which follow the original intended use. Specifically, NLTK is used for language analysis, fairseq is used for model training, and the linguistic database is used for accessing language statistics. Table 10 shows the exact chapters/features of linguistic databases we used.

Fairseq model	share-decoder-input-output-embed	True
	embed_dim	128
	ffn_embed_dim	512
	layers	2
	heads	2
	dropout	0.3
	attention_dropout	0.1
	#params.	462K
Optimizer	algorithm	AdamW
	learning rates	5e-4
	betas	(0.9, 0.98)
	weight decay	0.01
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	400
	warmup init learning rate	1e-7
Training	batch size	2,048 tokens
	tokens-per-sample	128 tokens
	sample-break-mode	none
	epochs	10

(a) Transformer.

Fairseq model	share-decoder-input-output-embed	True
	embed_dim	128
	hiden_size	512
	layers	2
	dropout	0.1
	#params.	3,547K
Optimizer	algorithm	AdamW
	learning rates	5e-4
	betas	(0.9, 0.98)
	weight decay	0.01
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	400
	warmup init learning rate	1e-7
Training	batch size	2,048 tokens
	tokens-per-sample	128 tokens
	sample-break-mode	none
	epochs	10

(b) LSTM.

Fairseq model	share-decoder-input-output-embed	True
	embed_dim	64
	hiden_size	64
	layers	2
	dropout	0.1
	#params.	49K
Optimizer	algorithm	AdamW
	learning rates	5e-4
	betas	(0.9, 0.98)
	weight decay	0.01
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	400
	warmup init learning rate	1e-7
Training	batch size	2,048 tokens
	tokens-per-sample	128 tokens
	sample-break-mode	none
	epochs	10

(c) RNN.

Table 11: Hyperparameters of LMs