



# Examining the Inductive Bias of Neural Language Models with Artificial Languages

Jennifer C. White  Ryan Cotterell  

 University of Cambridge,  ETH Zürich

jw2088@cam.ac.uk, ryan.cotterell@inf.ethz.ch

## Abstract

Since language models are used to model a wide variety of languages, it is natural to ask whether the neural architectures used for the task have inductive biases towards modeling particular types of languages. Investigation of these biases has proved complicated due to the many variables that appear in the experimental setup. Languages vary in many typological dimensions, and it is difficult to single out one or two to investigate without the others acting as confounders. We propose a novel method for investigating the inductive biases of language models using artificial languages. These languages are constructed to allow us to create parallel corpora across languages that differ only in the typological feature being investigated, such as word order. We then use them to train and test language models. This constitutes a fully controlled causal framework, and demonstrates how grammar engineering can serve as a useful tool for analyzing neural models. Using this method, we find that commonly used neural architectures exhibit different inductive biases: LSTMs display little preference with respect to word ordering, while transformers display a clear preference for some orderings over others. Further, we find that neither the inductive bias of the LSTM nor that of the transformer appears to reflect any tendencies that we see in attested natural languages.

## 1 Introduction

Modern neural architectures used for language modeling, e.g. Transformer-based language models (Vaswani et al., 2017) and language models based on long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Sundermeyer et al., 2012), are intrinsically black boxes. This makes it difficult to understand whether their structure leads to an inductive bias which results in certain types of language being easier to learn and model. To make this point more plainly, we cannot easily conclude

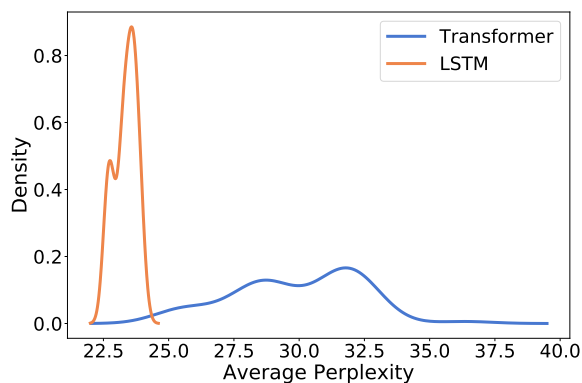


Figure 1: Distribution of average perplexities achieved by transformer- and LSTM-based language models on our artificial languages with varying word order.

much about whether an LSTM language model will perform better on SVO or SOV languages by simply examining its structure. Moreover, satisfactorily investigating the inductive bias of neural models has the potential to yield useful insight into how they work. In this work, we explore whether neural language models exhibit biases towards certain types of languages in a novel causal framework through the use of **artificial languages**.

One of the key problems involved in investigating the effect of typological features on language model performance is the difficulty in isolating only the features being investigated, without influence from other features of the languages being investigated or the data being used. For example, if one were to compare language model performance on English, an SVO language, and Japanese, an SOV language, it would be difficult to directly attribute differences in performance to the difference in word ordering alone. This is because English and Japanese also differ in many other typological dimensions, such as how subjects are marked, the extent of subject–verb agreement and use of postpositions or prepositions, which could contribute to the difference in performance. Indeed, recent correlational studies have failed to find an effect

between language model performance and typological features (Cotterell et al., 2018; Mielke et al., 2019). Moreover, the sentences used for training and testing may differ in content, style or information density, which could further contribute to differences in performance.

Thus, we offer a study investigating the inductive biases of language models through the construction of artificial languages. Our approach involves creating small context-free grammars resembling subsets of attested languages, which we then use to train and evaluate language models. In an approach inspired by Chomsky’s (1981) framework of principles and parameters, we imbue our grammars with “switches” that indicate how to permute the ordering of the non-terminals in a given production. Through generating grammars with all possible combinations of these switches, we can create artificial languages of differing typological profiles. This experimental paradigm allows us to conduct carefully controlled studies by varying only the typological parameter and make a causal claim.

Using our method, we investigate inductive biases related to the head-directionality of several constructions. We find that LSTM-based architectures show little bias towards any particular ordering, achieving similar average perplexities on all grammar variations tested. This contradicts recent findings by Ravfogel et al. (2019) who find LSTMs have a preference for SVO word order. Conversely, we find that performance of transformer-based architectures varies significantly across our artificial languages; this is visualized in Figure 1. This indicates that some combinations of the switches result in languages with word orders that are harder for the transformer to model than others. Our analysis suggests that neither the performance of the transformer-based architectures nor of the LSTM-based architectures reflects any known tendencies in attested natural languages, with the best performance being achieved on languages with the rarely-attested OVS sentence ordering. Importantly, our method exposes that transformer-based language models and LSTM-based language models have vastly different inductive biases, a result that has not been clearly stated in the NLP literature.

## 2 Why Artificial Languages?

### 2.1 Previous Work

Artificial languages have previously been used to investigate the ability of neural architectures with

respect to specific phenomenon, such as their ability to acquire hierarchical generalizations (McCoy et al., 2018) and whether they can use systematic composition skills to make generalizations (Lake and Baroni, 2018). Bowman et al. (2015) also used artificial languages to investigate the ability of LSTMs to learn compositional structure, and compare their ability to that of tree-structured models.

The work most closely related to ours is that of Ravfogel et al. (2019). Taking methodological inspiration from Wang and Eisner (2016), they create artificial versions of English with modified word order and case systems, including a version with object–verb agreement. They use the task of predicting the number of the subject and object of a missing verb to examine language model performance across these variations. They find that the models perform better on this task for the language with SVO word order. What they leave unchanged in their experiment, however, is the original English ordering within the constituents, e.g. the adjective–noun ordering in a noun phrase. However, constituent order correlates with ordering of other grammatical constituents typologically (Greenberg, 1963), and this could lead to unwarranted preferences for the original English ordering. Our work addresses this problem by using fully artificial languages rather than modifying English sentences. This allows for our experiment to be more controlled by eliminating possible confounders.

Other work conducted on the topic of inductive biases of language models has tended to focus on correlational studies investigating the relationship between typological features extracted from the World Atlas of Language Structures (WALS; Dryer and Haspelmath, 2013), which have only found negative results (Cotterell et al., 2018; Mielke et al., 2019). Since this work looked exclusively at the features of attested natural languages, it is difficult to control for the multiple typological dimensions along which any two natural languages differ. Further, given the large number of typological features exhibited among the world’s languages, there are simply not enough attested languages to make strong correlational claims. Mielke et al. (2019) ultimately concluded with a negative result; this negative result, in part, motivates our study.

### 2.2 The Necessity of Artificial Languages

We suggest that properly investigating the inductive biases of language models will likely

require artificial languages. Choosing languages to investigate the inductive bias of a language model requires a trade-off between the experiment being realistic and being controlled. Using attested natural languages gives us the most realistic representation of natural language and all its complexities, but this also reduces the level of control and makes it difficult to disentangle the various typological variables that differ between languages. Indeed, this was the conclusion of Mielke et al. (2019). Work such as Ravfogel et al. (2019) finds some mid-point by using artificial languages which have been modified from English. This means that the language is less natural and more controlled, but does not maximize either.

In our experiments, we have chosen to maximize the level of control. This means that our grammars are simple and do not necessarily cover all possible constructions that one would expect to see in a natural language. However, our reward for this sacrifice is that we can precisely control and understand how two languages tested differ from one another. We argue that this provides a good base for the exploration of inductive bias, as when differences are observed under these conditions we may now make a causal claim about their origin. In future work, the base grammars could be changed and extended as much as necessary to test additional hypotheses.

### 3 Constructing Controlled Languages

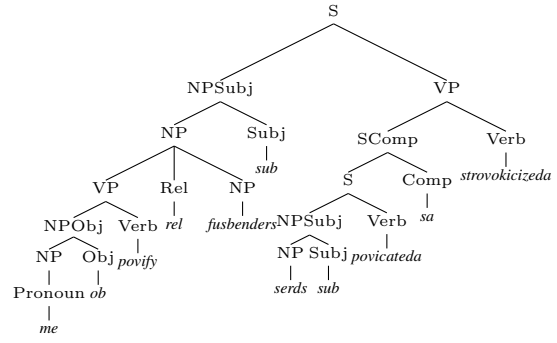
#### 3.1 A Fully Controlled Experiment

A context-free grammar (CFG) is a quadruple  $(\mathcal{N}, S, \Sigma, \mathcal{R})$  where  $\mathcal{N}$  is a set of non-terminals,  $S \in \mathcal{N}$  is a distinguished start non-terminal,  $\Sigma$  is an alphabet and  $\mathcal{R}$  is a set of production rules. An element  $r \in \mathcal{R}$  takes the form  $N \rightarrow \alpha$  where  $\alpha \in (\mathcal{N} \cup \Sigma)^*$ . A CFG defines a subset of  $\Sigma^*$ .

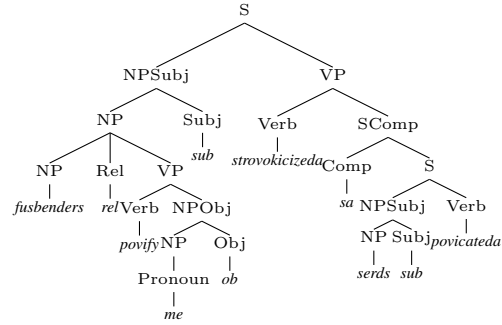
Probabilistic context-free grammars (PCFG) are a probabilistic generalization of CFGs. Rather than simply defining a subset of  $\Sigma^*$ , a PCFG gives us a probability distribution of  $\Sigma^*$  where the structure of the grammar gives us the structural zeros of the distribution. Given a PCFG, we can take samples from it in order to generate sentences.

We set out to construct a set of PCFGs to expose the inductive bias of neural language models. These grammars are parametrized by several “switches”, which determine the ordering of constituents within the grammar. The “switches” used are described in more detail in §3.3.

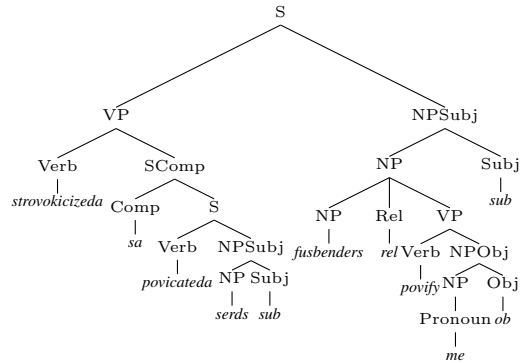
We write an initial base PCFG in which



(a) Grammar 000000: me ob povify rel fusbenders sub serds sub povicateda sa strovokicizeda .



(b) Grammar 011101: fusbenders rel povify me ob sub strovokicizeda sa serds sub povicateda .



(c) Grammar 111111: strovokicizeda sa povicateda serds sub fusbenders rel povify me ob sub .

Figure 2: Trees showing the structure of parallel sentences across 3 of our artificial languages

productions are written to correspond with the ordering obtained when all switches are “off”.<sup>1</sup> In this base PCFG, the rules which are affected by the toggling of each switch are marked. From this, sentences are sampled. On generation, each production in these sentences is marked with the switch it is associated with. We then work through every combination of switches, replicating this same set of generated sentences and reversing productions as required by the switches, to produce

<sup>1</sup>The choice of which permutation is “on” or “off” is arbitrary. In this case, “off” switches correspond to head-final orderings.

multiple parallel corpora, identical in their content up to a reordering of constituents.

This experimental set-up allows us to ensure that sentences in the corpus for each of our artificial languages differ only in the configuration of the switches. In this way we can be confident in attributing any differences in performance to a causal difference in these switches rather than any differences caused by confounders, e.g. content, style or complexity of the sentences.

### 3.2 Our Context-Free Grammar

Now we describe the construction of the PCFG with which we experiment in this work. Example sentences from several of our generated languages are shown in Figure 2. The base grammar and the scripts for sampling from it and generating corpora for all switch configurations will be released at <https://github.com/rycolab/artificial-languages>.

**The Alphabet  $\Sigma$ .** Open-class words were taken from a list of phonotactically plausible English pseudowords (Kharkwal, 2014). These pseudowords included verbs, nouns and adjectives. We inflected the nouns manually for English plurality (adding *s* or *es*) depending on what English phonotactics requires. We conjugated the verbs for present and past tense, again, using the rules of English. Additional morphological markers that are not present in English, e.g. subject and object markers and an additional marker to denote a plural past tense verb form, were obtained by randomly sampling two-letters slices from the list of morphological plausible words.<sup>2</sup> Pronouns and prepositions were also obtained in this fashion.

**The Non-Terminals  $\mathcal{N}$ .** Our grammar has a single distinguished start symbol *S*. It describes verb phrases (VP), containing transitive and intransitive verbs, as well as verbs that take a sentential complement (complementizers are denoted *Comp*). Nouns are marked as being objects or subjects using a particle (denoted *Obj* or *Subj*). Verbs in our grammar have two tenses (past and present). Noun phrases (NP), including those modified by adjectives (*Adj*), relative clauses (where relativizers are denoted *Rel*) and prepositional phrases (PP), are described in our grammar.

<sup>2</sup>This sampling occurred only once, and markers used were the same for all words.

Rule for each switch value		
Switch	0	1
<b>S</b>	$S \rightarrow NP VP$	$S \rightarrow VP NP$
<b>VP</b>	$VP \rightarrow NP VP$	$VP \rightarrow VP NP$
<b>Comp</b>	$S_{Comp} \rightarrow S Comp$	$S_{Comp} \rightarrow Comp S$
<b>NP</b>	$NP \rightarrow PP NP$	$NP \rightarrow NP PP$
<b>PP</b>	$PP \rightarrow NP Prep$	$PP \rightarrow Prep NP$
<b>NP</b>	$NP \rightarrow Adj NP$	$NP \rightarrow NP Adj$
<b>Rel</b>	$NP \rightarrow VP Rel Noun$	$NP \rightarrow Noun Rel VP$

Table 1: Rules that are switchable in our grammar. Subscripts for tense and number agreement are not shown for simplicity.

**The Production Rules  $\mathcal{R}$ .** Our production rules  $\mathcal{R}$  cover several common productions seen in natural language. We list the production rules which are subject to switching in our experiment in Table 1.

**Modeling Morphological Agreement.** Our grammar models a simple form of morphological agreement: verbs agree with their subjects in number (singular or plural). This introduces an element of long-term dependencies into our languages – if a language model is to correctly predict a verb form, it must carry information about the number of the subject. In order to enforce this agreement in our grammar, non-terminals are subscripted with their number (where applicable).

**Assigning Probabilities.** Weights given to each production were chosen manually through experimentation. Some principles for choosing weights for a grammar in this manner are described by Eisner and Smith (2008). An automated method of assigning weights could be explored in future work.

### 3.3 Controlled Typological Variation

Our end goal is to construct a grammar parameterized by a binary vector of  $K$  switches. We denote such a vector of switches  $\mathbf{b} \in \{0, 1\}^K$ . Toggling an individual switch in the grammar reverses the order of the right-hand sides of a set of production rules. For example, the switch that we term the *S* switch reverses the order of the production  $S \rightarrow NP VP$  to create  $S \rightarrow VP NP$ .<sup>3</sup>  $2^K$  different grammars are possible from  $K$  binary switches. In the following paragraphs, we describe each of the switches we consider in this work.

**Position of subject in sentence (S Switch).** This switch determines the order in which a subject and its verb phrase appear within a sentence. If the

<sup>3</sup>Details of all switches are shown in Table 1.



Japanese			English		Spanish	
Switch	Value	Example	Value	Example	Value	Example
S	0	猫が食べる。	0	The cat eats.	0	El gato come.
VP	0	猫がネズミを食べる。	1	The cat eats the mouse.	1	El gato come el ratón.
Comp	0	猫が食べると思う。	1	I think that the cat eats.	1	Pienso que el gato come.
PP	0	テーブルの上の猫が食べる。	1	The cat on the table eats.	1	El gato sobre la mesa come.
NP	0	小さな猫が食べる。	0	The small cat eats.	1	El gato pequeño come.
Rel	0	ミルクを飲む猫が食べる。	1	The cat that drinks milk eats.	1	El gato que bebe leche come.

Table 2: Demonstration of the orders of the switch constituents in Japanese, English and Spanish

switch has a value of 0, the rule  $S \rightarrow NP VP$  is used, which is the order used in the vast majority of the world's languages, including SVO languages such as English and SOV languages such as Japanese. If the switch has a value of 1, the rule becomes  $S \rightarrow VP NP$ . This order is rare among attested natural languages, but can be seen in VOS languages such as Malagasy and OVS languages such as Hixkaryana.

**Position of verb in verb phrase (VP Switch).** This switch determines whether a direct object precedes or follows its verb. If the switch has a value of 0, we use the head-final order, with the object preceding the verb. This is seen in languages such as Japanese and Turkish. If the switch has a value of 1, the head-initial order is used, with the object following the verb. This is seen in languages such as English and Chinese. This switch, in combination with the S switch, determines the overall ordering of subject, object and verb within a sentence. If the values of these switches are (0, 0), the language will have SOV word order, like Japanese and Turkish. If they are (1, 1), the language will have VOS order, which is rare but can be seen in languages such as Malagasy. SVO languages such as English correspond to (0, 1). (1, 0) corresponds to OVS order, which is attested in only a very small number of human languages.

**Position of complementizer in sentential complement (Comp switch).** This switch determines whether a complementizer begins or ends a sentential complement. If the switch has a value of 0, the complementizer appears in head-final position, at the end of the complement. This is the order seen in Japanese. If the switch has a value of 1, the complementizer appears in head-initial position, at the beginning of the complement. This is the order seen in English.

**Ordering of prepositional phrase (PP Switch).** This switch determines the ordering of a preposi-

tional phrase. If the switch has a value of 0, the prepositional phrase precedes the noun it modifies, and the prepositional phrase ends with a preposition, in head-final order. This order is seen in Japanese. If the switch has a value of 1, the prepositional phrase follows the noun it modifies, and the preposition begins the prepositional phrase, in head-initial order. This order is seen in English.

**Position of adjective in noun phrase (NP Switch).** This switch determines whether an adjective appears before or after the noun it modifies. If the switch is 0, the adjective precedes the noun (as in English and Japanese) and if it is 1, the adjective follows the noun (as in Spanish and Irish).

**Position of relative clause (Rel switch).** This switch determines the position of a relative clause with respect to the noun it modifies. If the switch has a value of 0, a relative clause is followed by a relativizer and then the noun it modifies. This order is seen in Japanese. If the switch has a value of 1, the noun being modified appears first, followed by a relativizer and the clause. This order is seen in French and English.

The unmarked word order of some attested languages can be approximately identified with particular switch vectors.<sup>4</sup> For example, standard English order corresponds approximately to (0, 1, 1, 1, 0, 1), Japanese to (0, 0, 0, 0, 0, 0) and Spanish to (0, 1, 1, 1, 1, 1).<sup>5</sup> This is demonstrated in Table 2. We note that our configurations cannot account for all possible word orders seen in attested languages (VSO languages are not represented, for example), but constitute a subset of possible orders.

<sup>4</sup>This is, of course, a simplification, since word order within a natural language can follow more complex rules, or allow for flexibility.

<sup>5</sup>From this point on, grammars will be referred to by their configuration of switches, sans brackets, e.g. Grammar 011101.

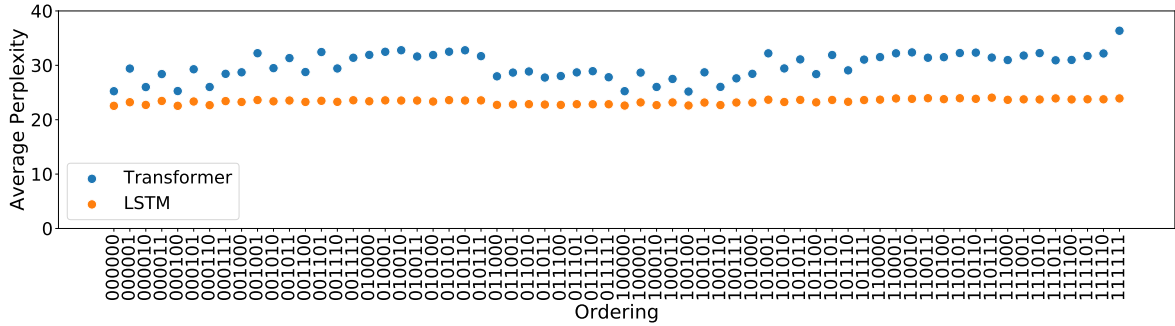


Figure 3: All scores achieved by LSTM- and transformer-based models

## 4 Experiments

**Architectures and Data.** In order to compare inductive biases across architectures, two neural architectures were tested: transformers and LSTMs. We used the implementation available as part of Fairseq (Ott et al., 2019). Our base grammar has  $K = 6$  switches, i.e. 6 binary choice points as described in §3.3. This results in  $2^6 = 64$  possible grammars. For each of these grammars we generated 100,000 sentences, which were divided into 10 splits of 10,000.<sup>6</sup> The sentences generated for each grammar differed only in the designated choice points, i.e. in the ordering of their constituents. This meant that each sentence appeared in an equivalent form in each grammar. As such, for each sentence, we can compare the perplexity of the 64 variants of the sentence as calculated by language models trained on the corresponding grammars. Each split of 10,000 sentences was divided into an 80–10–10 train–dev–test split.<sup>7</sup>

**Procedure.** We trained both a transformer-based and an LSTM-based language model on each train split and the models were evaluated on the test split. This procedure resulted in 10 language models per architecture for each possible grammar, each of which was evaluated on 1,000 sentences in their respective test set. The perplexity achieved on these test sets was averaged across the 10 splits, to give the average perplexity for that grammar. This approach helps to account for the variability between individual training runs.

<sup>6</sup>10,000 sentences may sound like a relatively small number, but we note that our artificial languages are simple with small vocabularies, so we consider this number to be sufficient.

<sup>7</sup>Equivalent sentences across grammars were assured to be in the equivalent splits for each grammar, so train, dev and test sets across grammars contained the same sentences up to reordering of constituents.

## 5 Results and Analysis

### 5.1 Perplexity Evaluation

The average perplexity on the test set was measured for each grammar. This measures how well a language model explains the held-out test set. The lower the perplexity the better the language model fits the held-out data. Average perplexity achieved across all grammars by the transformer- and LSTM-based models are shown in Figure 3.<sup>8</sup>

### 5.2 Mixed-Effects Modeling

We use a linear mixed-effects model to investigate the effects of each choice point in the grammar. This allows us to model the effect of each switch in the grammar, and first-order interaction terms between them, on the perplexity of a sentence, while controlling for the fact that perplexities for parallel sentences across grammars are related (by using a random intersect per sentence grouping). This model is explained in detail below.

Assume we have  $N$  paired sentences from each of our  $2^K$  grammars. Let  $\mathbf{L} \in \mathbb{R}_{\geq 0}^{N \times 2^K}$  be a non-negative real matrix of the perplexity obtained for every test sentence across every grammar. Specifically, we have that  $L_{nk}$  is the perplexity for the  $n^{\text{th}}$  sentence under the  $k^{\text{th}}$  grammar. Furthermore, let  $\mathbf{S} \in \{0, 1\}^{2^K \times \left(\frac{K(K-1)}{2} + K\right)}$  be the binary matrix containing the configuration of switches and the  $\frac{K(K-1)}{2} + K$  switch–switch interactions for each of the  $2^K$  grammars in contrast coding (Wu, 2009). Thus, we have that the column vector  $\mathbf{S}_{k\bullet}$  is a binary vector of length  $\frac{K(K-1)}{2} + K$ . Let  $\boldsymbol{\beta} \in \mathbb{R}^{\frac{K(K-1)}{2} + K}$  be a vector of real coefficients to be estimated describing the effect of each switch and their interactions. Let  $u_n \sim \mathcal{N}(0, \sigma_{\text{dif}}^2)$  be

<sup>8</sup>Error bars are omitted, but across grammars the error on each measurement is generally between 0.25 and 0.5.

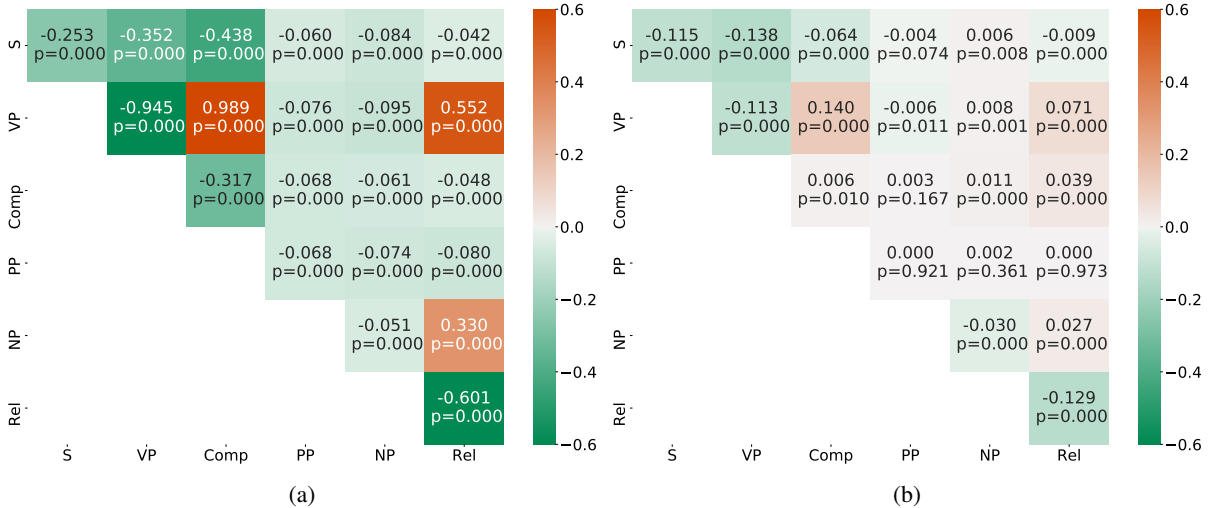


Figure 4: Heat maps showing the coefficients obtained for a mixed-effects model for perplexity as predicted by (a) transformers and (b) LSTMs.

a sentence-specific difficulty term (a random effect) and let  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$  be a sentence-grammar-specific noise term. Now, we model an individual perplexity  $L_{nk}$ , which corresponds to the  $n^{\text{th}}$  sentence and the  $k^{\text{th}}$  grammar, as follows:

$$L_{nk} = \mathbf{S}_{k\bullet} \cdot \boldsymbol{\beta} + u_n + \varepsilon \quad (1)$$

Importantly, we draw one  $u_n$  for each unique sentence. It is in this sense that  $u_n$  acts as a term for modeling sentence difficulty. We may write eq. (1) as the following

$$L_{nk} \sim \mathcal{N}(\mathbf{S}_{k\bullet} \cdot \boldsymbol{\beta}, \sigma_{\text{dif.}}^2 + \sigma^2) \quad (2)$$

which reveals that it is no more than a simple Gaussian model with tied parameters. We estimate  $\boldsymbol{\beta}$ ,  $\sigma_{\text{dif.}}^2$ , and  $\sigma^2$  through maximum-likelihood estimation, which, in Gaussian models, is equivalent to least-squares estimation.

A positive coefficient  $\beta_j$  for a given switch means that models perform worse with head-initial ordering for that switch, while a negative coefficient means the opposite. Since the fixed effects were input using contrast coding, the interaction terms in our model deal with the effects of two constituents *sharing head-directionality*. A positive coefficient for an interaction means that the models perform *worse* when they share head directionality, and a negative coefficient means the opposite. Head-directionality is commonly correlated between sentence constituents in attested natural languages, so if the biases of these architectures reflected human languages, we would expect most interaction terms to be negative. The coefficients ob-

tained for the transformers are shown in Figure 4a. Those for the LSTMs are shown in Figure 4b.

## 6 Discussion

**Differences Between Architectures.** It is clear from Figure 3 that the transformer- and LSTM-based models do not show the same inductive biases with respect to the switches we investigated. Across all possible configurations of the switches, LSTMs achieve very similar average perplexities, suggesting that they have little preference for any particular set of constituent orderings. In contrast, the average perplexities achieved by the transformers vary considerably between grammars. This demonstrates clearly that the two models exhibit distinctly different preferences with regard to orderings of words within in a sentence. Further, the clear contrast between the coefficients obtained by the mixed-effects models for transformers and LSTMs (shown in Figure 4a and Figure 4b, respectively) demonstrates a stark contrast between the two models. None of the switches investigated, or their first-order interactions, appear to have a substantial effect on the scores obtained in the case of the LSTM-based models, whereas the transformer-based models are clearly affected to a much greater degree by the configuration of these switches. Given that these two architectures are both commonly used for similar tasks, such a difference in their inductive biases is noteworthy.

**Correlated Switches.** Figure 4a shows the coefficients obtained by the mixed-effects model employed to investigate the effects of the switches

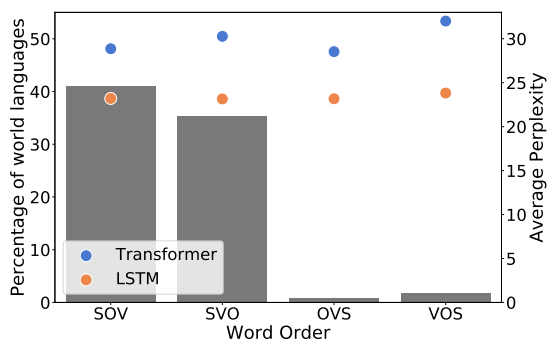


Figure 5: The prevalence of word orders across languages (Dryer, 2013), plotted with the average perplexities achieved on each of these groups of grammars by transformer- and LSTM-based models

on performance for the transformer-based models. The diagonal values (for single switches) are all negative coefficients, which indicates that the model performance is better when these have head-final ordering. Off-diagonal values are the coefficients obtained for the interaction terms between two switches. A positive value here indicates that when these two switches have the same value (either both head-initial or both head-final), the performance of the model is worse. A negative value means that when the two switches have the same value, the performance is better. Most of the off-diagonal elements have small values, with a few exceptions. The coefficients of the cross terms between the S and VP switches and the S and Comp switches are larger negative values, which indicates that when these constituents share their head-directionality the performance of the transformer-based models is better. The coefficients of the cross terms between the VP and Comp, VP and Rel and NP and Rel switches are larger positive values, indicating that the transformers perform worse when these constituents share head-directionality. Generally, attested natural languages tend to exhibit a tendency towards one head-directionality, but the transformer does not seem to have inductive biases that reflect this. The corresponding coefficients for the LSTM-based models, shown in Figure 4b, are all small, further demonstrating that the LSTMs are largely agnostic to word ordering.

### Tendencies in Attested Natural Languages.

We wish to consider the question of whether the biases of these models are in any way reflective of word order tendencies that we see across attested natural languages. All word orders are not equally common among natural languages, and it is inter-

esting to consider whether the word orders that these models are able to model more successfully are those which are more commonly seen in natural language. Some have speculated that the skew of word orders in human languages could possibly be reflective of human cognitive biases (Culbertson et al., 2012, 2019), so it would be interesting to see to what extent the inductive biases of these models reflects this skew. Since LSTMs appear to show no preference for any word order over the others, they are clearly not reflective of attested tendencies in word order. To attempt to answer this question for the transformers, we begin by comparing the performance of the models on subsets of grammars with the prevalence of similar languages among humans. In Figure 5, the grammars are grouped by how they order the verb, object and subject of a sentence, and the average perplexities achieved by the language models on each of these groups is shown. On the same figure, we display the estimated prevalence of these orderings among the world’s languages (Dryer, 2013). It is clear that these two things are not correlated, with the transformer performing similarly on SOV languages, the most common among the world’s languages, and OVS languages, which are rarely attested. This shows that the bias exhibited by transformers does not reflect tendencies among attested languages. A further indication of this is the lack of a strong preference for switches sharing head-directionality as shown in Figure 4a. In human languages, the headness of constituents is often correlated (Greenberg, 1963). We would expect to see this through negative coefficients for interaction terms in the mixed-effects model for constituents whose orders commonly correlate. However, we do not observe this for all correlations. For example, we would expect the PP switch to show a strong preference for shared head-directionality with other switches, which we do not observe.

## 7 Conclusion

We propose a novel methodology for the investigation of the inductive bias of language models using the technique of creating carefully controlled artificial languages. This approach allows for the elimination of differences in corpora between languages and means that typological variation between languages can be restricted exclusively to the typological features being investigated. We use this methodology to investigate the inductive bias of two neu-



ral architectures which are commonly used for this task: LSTMs and transformers. We found that these two models have starkly different inductive biases with respect to word order, with the LSTM showing little variation in performance across word order, while the performance of the transformer varied significantly across artificial languages.

## Acknowledgements

We thank Simone Teufel for providing feedback on an early draft.

## Ethical Considerations

The authors foresee no ethical concerns with the research presented in this paper.

## References

- Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. [Tree-structured composition in neural networks without tree-structured architectures](#). In *Proceedings of the 2015 International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches*, volume 1583, page 37–42.
- Noam Chomsky. 1981. *Lectures on Government and Binding: The Pisa Lectures*. Walter de Gruyter.
- 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.
- Jennifer Culbertson, Marieke Schouwstra, and Simon Kirby. 2019. [From the world to word order: the link between conceptual structure and language](#). *PsyArXiv*.
- Jennifer Culbertson, Paul Smolensky, and Géraldine Legendre. 2012. [Learning biases predict a word order universal](#). *Cognition*, 122(3):306–329.
- Matthew S. Dryer. 2013. [Order of subject, object and verb](#). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Jason Eisner and Noah A. Smith. 2008. [Competitive grammar writing](#). In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 97–105, Columbus, Ohio. Association for Computational Linguistics.
- Joseph H. Greenberg. 1963. [Some universals of grammar with particular reference to the order of meaningful elements](#). In Joseph H. Greenberg, editor, *Universals of Human Language*, pages 73–113. MIT Press, Cambridge, Mass.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Gaurav Kharkwal. 2014. *Taming the Jabberwocky: Examining Sentence Processing with Novel Words*. Ph.D. thesis, Rutgers University-Graduate School-New Brunswick.
- Brenden 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*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882, Stockholm, Sweden. PMLR.
- R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. [Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks](#). In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2093–2098.
- Sabrina 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 Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.
- 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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, Volume 1 (Long and Short Papers)*, pages 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. [LSTM neural networks for language modeling](#). In *Thirteenth Annual Conference of the International Speech Communication Association*, pages 194–197.

- 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010.
- Dingquan Wang and Jason Eisner. 2016. [The galactic dependencies treebanks: Getting more data by synthesizing new languages](#). *Transactions of the Association for Computational Linguistics*, 4:491–505.
- Lang Wu. 2009. *Mixed Effects Models for Complex Data*. CRC Press.