# *Trees neural those*: RNNs can learn the hierarchical structure of noun phrases

**Yevgen Matusevych (yevgen.matusevych@ed.ac.uk)**
School of Philosophy, Psychology and Language Sciences; School of Informatics
University of Edinburgh

**Jennifer Culbertson (jennifer.culbertson@ed.ac.uk)**
School of Philosophy, Psychology and Language Sciences
University of Edinburgh

## Abstract

Humans use both linear and hierarchical representations in language processing, and the exact role of each has been debated. One domain where hierarchical processing is important is noun phrases. English noun phrases have a fixed order of prenominal modifiers: demonstratives – numerals – adjectives (*these two green vases*). However, when English speakers learn an artificial language with postnominal modifiers, instead of reproducing this linear order they preserve the distance between each modifier and the noun (*vases green two these*). This has been explained by a hierarchical homomorphism bias. Here, we investigate whether RNNs exhibit this bias. We pre-train one linear and two hierarchical models on English and expose them to a small artificial language. We then test them on noun phrases from a study with humans and find that only the hierarchical models can exhibit the bias, supporting the idea that homomorphic word order preferences arise from hierarchical, and not linear relations.

**Keywords:** hierarchical processing, noun phrase, artificial language learning, neural networks, homomorphism

## Introduction

Do humans represent language primarily in terms of linear or hierarchical structure? Usage-based theories of language (Christiansen & Chater, 2016; Bybee, 2006; Tomasello, 2003) tend to emphasize the local nature of acquisition and processing. This locality principle naturally promotes the linear structure of a language as the defining factor influencing emergent linguistic representations. By contrast, alternative theories of language are based on the fundamental hierarchical nature of linguistic representations (Chomsky, 1957; Adger, 2003). While both local/linear and hierarchical representations may play a role in language, debates about the exact role of each are ongoing (e.g., Brennan et al., 2016; Widmer et al., 2017).

One particularly fruitful testbed in this domain has been the order of elements in a noun phrase—on its face a relatively simple structure, but one where significant hierarchical organization has nevertheless been hypothesized (Adger, 2003; Cinque, 2005). In English, modifiers including adjectives (*green*), demonstrative pronouns (*this*), and numerals (*one*), are all placed in the prenominal position (*green vases*). What happens when speakers of a language like English need to learn a language with *postnominal* modifiers (*vases green*)? If they have never encountered a string of multiple modifiers in this language, one possible strategy would be to reproduce them in their English linear order, but in postnominal position: *vases these two green*. However, a number of studies (Culbertson & Adger, 2014; Martin et al., 2019, 2020; Culbertson et al., 2020) show that learners instead tend to *reverse* the order of modifiers: *vases green two these*. This result has been interpreted as evidence that speakers are making inferences based on hierarchical structure. More specifically, speakers have been argued to demonstrate a cognitive bias for homomorphism—a transparency of mapping between the underlying hierarchical structure of the noun phrase, and the linear order of elements (Martin et al., 2020). In the case of a noun phrase, the underlying structure is associated with semantic scope (Adger, 2003; Culbertson & Adger, 2014; Martin et al., 2020): adjectives form a semantically coherent unit with the noun, thus forming a constituent, numerals then define the quantity of this constituent to create a countable unit, and demonstratives map this unit to the pragmatics of the context (e.g., location in relation to the speaker). Therefore, regardless of whether modifiers are in prenominal or postnominal position, their order relative to the noun reflects this *hierarchy* of meanings. Indeed, the vast majority of the world's languages feature a linear order of nominal modifiers that is homomorphic in this way (Cinque, 2005; Dryer, 2018).

If this homomorphism bias is a property of the human cognitive system, it must be possible to design a learning model from which this bias emerges. In this study, we take the first step in this direction and test three computational models on their ability to exhibit this bias and replicate human-like preferences for noun phrase word order. Specifically, we pre-train three recurrent neural network (RNN) language models—one linear (LSTM) and two hierarchical (Ordered Neurons and RNNG)—on English corpora. Then, following Culbertson & Adger (2014), we expose the models to a small artificial language consisting of noun phrases with a single postnominal modifier and test the models on noun phrases with two postnominal modifiers, using the language models' ability to predict the next word in a sequence. This setup allows us to test whether the models can generalize to unseen grammatical structures, rather than merely reproduce structures from the input. In other words, a model not only has to induce the structure of the English noun phrase during pre-training, but also mirror that structure in the artificial noun phrases with multiple modifiers, without ever seeing such phrases in the input. Our goal is to determine whether the models are indeed able to show the preference for the homomorphic word order consistently observed in human speakers.

Table 1: Three conditions in the experiment of Culbertson & Adger (2014).

| Condition | Parts of speech | Example |
|---|---|---|
| DEM-ADJ-N | demonstrative–adjective–noun | *this green vase* |
| DEM-NUM-N | demonstrative–numeral–noun | *these three keys* |
| NUM-ADJ-N | numeral–adjective–noun | *two big boxes* |

To preview, we find that the linear LSTM model does not show human-like preferences, supporting the idea that these preferences do not arise purely from linear information about word order. By contrast, the hierarchical models fare better: the Ordered Neurons model correctly predicts a homomorphism preference in two out of the three conditions tested, while the RNNG predicts this in the third condition only.

## Background

### The homomorphism bias in artificial languages

Culbertson & Adger (2014) trained and tested English native speakers on an artificial language which consisted of noun phrases with modifiers in the postnominal position (e.g., *vase green*). For simplicity, English words were used in this experiment. During the training, participants only saw phrases with a single modifier: an adjective (*green*), a demonstrative pronoun (*this*), or a numeral (*two*), accompanied by their English 'translations' (i.e., the same noun phrases with the prenominal order of modifiers). In the testing phase participants had to choose translations for English phrases with *two* modifiers in three conditions (see Table 1), even though they did not see such phrases in the artificial language during training. If in this task English-speaking participants rely primarily on the linear structure of their native language, they would produce noun phrases such as *vase this green* (N-DEM-ADJ), simply moving the two prenominal modifiers to the postnominal position in their original order. Instead, participants showed preferences for the opposite, homomorphic word order (e.g., N-ADJ-DEM, *vase green this*) in all the three conditions. In a follow-up experiment with novel (non-English) words (Martin et al., 2020), similar results were obtained, although the preference for the homomorphic order was generally less reliable in the NUM-ADJ-N condition than in the other two. These findings received further support from follow-up studies with Thai speakers learning an artificial language and with English speakers producing spontaneous sequences of gestures (Culbertson et al., 2020). To summarize, there is a robust preference for homomorphic word order in noun phrases.

### Hierarchical biases in neural networks

A number of recent studies have explored the inductive biases of neural architectures. For example, Ravfogel et al. (2019) proposed a paradigm in which neural networks are trained on corpora consisting of artificial versions of English, which differ from natural English only in some typological properties. The models are then evaluated on their ability to predict the same linguistic feature (e.g., subject–verb number agreement) across the artificial languages. Using this method, Ravfogel et al. (2019) show that a bidirectional LSTM has a recency bias, i.e., favoring dependencies with recent elements. This characterizes an LSTM as a first and foremost linear model, even though it is able to capture some hierarchical long-distance dependencies (e.g., Gulordava et al., 2018).

White & Cotterell (2021) use a similar methodology of generating multiple constructed artificial languages to investigate how successfully on average neural models (LSTM and Transformer) learn each language. They find that LSTMs do not have a preference for a particular word order. Transformers have more difficulties with some word orders than others, but their preferences do not correspond to hypothesized human preferences—i.e., based on the distribution of word orders in the world languages.

McCoy et al. (2020) study inductive biases in linear (LSTM, GRU) and tree-based (ON-LSTM, RNNG) models using syntactic tasks that probe the models' capacity to learn hierarchical generalization, namely question formation and tense reinflection. They find that only the tree-based models are capable of making the correct generalizations.

Based on these findings, in our experiments below we use two hierarchical models: the ON-LSTM (Shen et al., 2018), which implicitly encodes hierarchical structure in its architecture, and the RNNG (Dyer et al., 2016), which is trained on parse trees and thus explicitly learns the hierarchical structure of a language. We also use a 'vanilla' LSTM as a baseline model. We expect to see human-like homomorphic preferences in noun-phrase word order in the two hierarchical models, but not in an LSTM.

## Methods

Our general approach for all models is to pre-train them on an English corpus (either Wikipedia or Penn Treebank, as described below) and then train them for a small number of epochs on artificial language noun phrases. During the artificial language training, we use the next-word prediction setup to test each model at certain intervals on the test stimuli consisting of noun phrases with two postnominal modifiers. Given a pair of stimuli with homomorphic vs. linear order, we identify each model's preference by measuring which of the two sequences that model considers to be more probable. All models are word-level language models, and no subword tokenization is used.

### Models

**LSTM.** LSTM is a RNN model that has been commonly used for studying human sentence processing (e.g., Gulordava et al., 2018; Marvin & Linzen, 2018; Futrell et al., 2019). It is trained on the task of predicting the next word in a sentence. As explained in the previous section, it primarily relies on local dependencies and has no hierarchical biases encoded in

its architecture. We use a pre-trained model of Gulordava et al. (2018)[1], with 2 stacked hidden layers, 650 units per layer, batch size 128, learning rate 20.0 and dropout rate 0.2. We further use Van Schijndel & Linzen's (2018) implementation to train and test this model on our artificial languages. We use the LSTM as a baseline, to ensure that the preference for the target word order does not arise in a linear model due to unknown properties of English that have nothing to do with homomorphism.

**ON-LSTM.** The Ordered Neurons LSTM (ON-LSTM) model (Shen et al., 2018) adds a hierarchical bias to the 'vanilla' LSTM model. While in the LSTM model all neurons in the hidden layer function in the same way, the ON-LSTM model promotes the differentiation of neurons: higher-order neurons only get updated once the lower-order neurons have been updated. Therefore, the higher a neuron's ranking, the longer it stores information, which results in tree-like hierarchical processing. Thanks to this feature, the model has been used in other studies looking at the hierarchical processing and syntactic generalization in RNNs (McCoy et al., 2020; Hu et al., 2020). We use the original implementation by Shen et al. (2018)[2] and their default hyperparameters: 1150 units in the hidden layer, embedding size 400, batch size 20, dropout and weight dropout 0.45 (0.3 for the recurrent layer, 0.5 for the input layer), and 1000 epochs for pre-training.

**RNNG.** While the two previous models are trained on text data, the Recurrent Neural Network Grammars (RNNG) model (Dyer et al., 2016) is trained on constituency trees to perform a parsing task. Therefore, it explicitly encodes hierarchical relationships in the training data. We use a recent implementation (Noji & Oseki, 2021)[3] based on its efficiency, with the default parameters (Adam optimizer, learning rate 0.001, dropout 0.3, 18 pre-training epochs), but smaller batch size of 128 due to technical limitations.

## English pre-training

We use a publicly available LSTM model pre-trained on a Wikipedia corpus (approximately 80M tokens, Gulordava et al., 2018). For consistency, we use the same corpus for training the ON-LSTM model. The RNNG, however, requires a parsed corpus, and we cannot train it on the Wikipedia data. Instead, we use the Penn Treebank data set (Marcus et al., 1993), as in the original study (Dyer et al., 2016).

The Wikipedia corpus could be directly used for pre-training the LSTM and ON-LSTM, while for pre-training the RNNG we had to introduce additional annotations in the Penn Treebank. Specifically, all noun phrases in the corpus were represented as shallow linear structures, without any hierarchy. If we trained the RNNG on this data, it would not be able to use its capacity as a hierarchical model in the noun phrase. Therefore, we augmented the Penn Treebank data with additional noun phrase annotations (cf. examples 1 vs. 2), to introduce the hierarchical structure in the English noun phrases.

(1) (NP (DT a) (JJ possible) (NN acquirer))

(2) (NP (DT a) (NP (JJ possible) (NP (NN acquirer))))

An alternative would be to use a corpus with hierarchical noun structures, but to our knowledge such corpora are not readily available. We use the standard train–development–test split of the Penn Treebank to pre-train the RNNG.

## Artificial language training

We try to keep our computational simulations as close as possible to the original experiment of Culbertson & Adger (2014). Following their setup, we consider the same 30 nouns, 10 adjectives, 10 numerals, and 4 demonstratives. Using this vocabulary, we generate 10 different training sets, where each training set consists of 30 noun phrases with a single postnominal modifier, 10 of each kind: N-ADJ (*scarf blue*), N-DEM (*car that*), and N-NUM (*shirts six*).

While all words from Culbertson & Adger's artificial language data appeared in the Wikipedia training corpus, some of them were missing from the Penn Treebank, a situation that could not arise in the original experiment where all the native English speakers were familiar with the words in the artificial language. Therefore, we extended the vocabulary using words from the Penn Treebank corpus. Only 13 nouns appeared in the Penn Treebank, and one option would be to choose 17 extra nouns to have 30 in total, as in the original experiment. Instead, we used additional 13 high-frequency and 13 low-frequency nouns to keep the size of the noun classes (original, low- and high-frequency) balanced, so that the artificial language training data for the RNNG was somewhat larger, with each training set including 39 instead of 30 items. We also used 4 frequent adjectives from the Penn Treebank, in addition to the 6 adjectives from the original set.

Importantly, because all models are pre-trained on full English sentences, training them on isolated noun phrases from an artificial language could potentially introduce an additional confound, the form of the training stimuli. Instead, we converted all the training data sets described above into full sentences that consist of a subject and a transitive verb, followed by the target noun phrase. To form these sentences, we randomly choose one of the four personal pronouns (*I, they, she, he*) and one of the two verbs (*see, want*) in the correct grammatical form, as shown in (3). This construction was chosen thanks to its high degree of abstraction: more specific (and less frequent) constructions could bias the model towards reproducing patterns seen in the training data.

(3) She wants shirts six.

Note that for training the RNNG on the artificial language, we converted these sentences into parsed sequences, using the standard Penn Treebank annotation augmented with the hierarchical noun structure as described above.

[1] https://github.com/facebookresearch/colorlessgreenRNNs
[2] https://github.com/yikangshen/Ordered-Neurons
[3] https://github.com/aistairc/rnng-pytorch

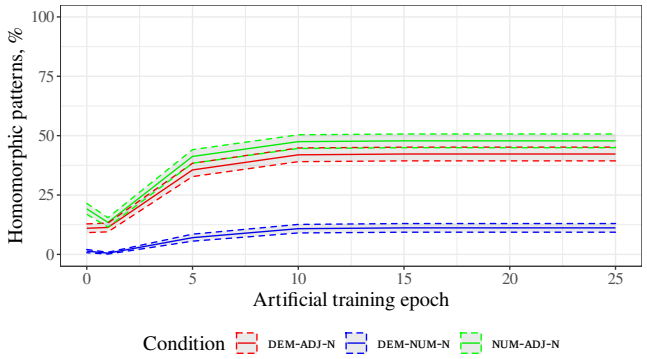



Figure 1: Percentage of sentence pairs for which the LSTM model shows preference for the homomorphic word order. Results are averaged over training/test sets and test items, error bands show the standard error of the mean over different test items.

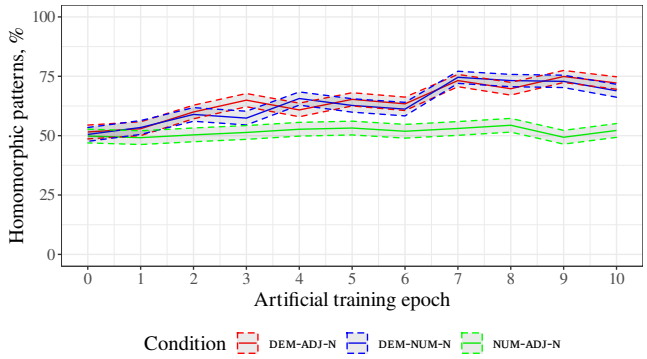



Figure 2: Percentage of sentence pairs for which the ON-LSTM model shows preference for the homomorphic word order. Results are averaged over training/test sets and test items, error bands show the standard error of the mean over different test items.

### Artificial language testing

Analogously, we generated 10 different test sets, each of them including only nouns that do not appear in the corresponding training set. Each test set included 30 pairs of noun phrases with two postnominal modifiers, 10 for each of the three conditions: DEM-ADJ-N, DEM-NUM-N, and NUM-ADJ-N). (For the RNNG, there were 13 pairs in each condition, 39 in total.) Each pair consisted of two alternatives: homomorphic (e.g., *pears purple those*) and non-homomorphic (*pears those purple*). We tested how probable the model finds each alternative, using the commonly adopted approach of measuring average sentence surprisal (Goodkind & Bicknell, 2018): the alternative with a lower surprisal value indicated the model's preference. We report all the results averaged over the 10 training–test set combinations.

## Results

### LSTM

We first look at the LSTM, our baseline model. Recall that it has no hierarchical bias of any kind, and therefore it would be surprising to see a preference for homomorphic word order in this model. The results in Figure 1 support this intuition. Before any exposure to the artificial language, the model shows a clear preference for the linear word order in all three conditions. This pattern suggests that the model readily transfers its knowledge about the English linear order of modifiers (DEM-NUM-ADJ-N) from the pre-training data. Because the LSTM mostly relies on local transitional probabilities, patterns such as DEM-ADJ, DEM-NUM, and NUM-ADJ are more likely than their inverse counterparts.

Interestingly, the preference for the linear order gets noticeably smaller with more training in the DEM-ADJ-N and NUM-ADJ-N conditions (the red and the green lines approach the 50% chance level, although the red DEM-ADJ-N line is still somewhat lower, indicating a small preference for the linear order). This is not the case for the DEM-NUM-N condition. Our simple analysis of bigram frequencies in the English pre-training data suggests that this result can be explained by the much higher frequencies of DEM-NUM bigrams (e.g., *that one*, *those six*; average frequency 161.1), compared to DEM-ADJ (e.g., *that blue*; average frequency 2.5) and NUM-ADJ bigrams (e.g., *three wooden*; average frequency 0.9) in the training data. The model is more likely to 'memorize' the high-frequency patterns, so that even after relatively large amounts of exposure to the artificial language, the DEM-NUM order is still transferred to the artificial language data to a larger extent, followed by DEM-ADJ and NUM-ADJ.

To ensure the reported results are statistically significant, we fit a series of mixed-effects logistic regression models (one per condition) to the data, with fixed effect of epoch, and random intercepts over stimulus, train–test subset, and subject–verb combination. Because there is a clear change in the model's preferences for all conditions between epochs 5 and 10, we fit separate regressions for epochs 0–5 and 10-25. The results suggest that early on during the artificial language learning the model's preferences for the non-homomorphic order are significantly different from chance in all three conditions: the intercepts are $-4.3$, $-18.8$, and $-3.1$ in the DEM-ADJ-N, DEM-NUM-N, and NUM-ADJ-N conditions, respectively, with all $p < .001$. Later, this preference is only significant in the DEM-ADJ-N and DEM-NUM-N (intercepts are $-13.3$ and $-25.8$, respectively, both $p < .001$), but not in the NUM-ADJ-N condition (intercept $-3.9$, $p = 0.248$), which supports our earlier observations.

To summarize, the LSTM model shows a preference for non-homomorphic word order, as expected. This suggests that the homomorphism preference observed by Culbertson & Adger (2014) does not simply arise in any learning model from unknown properties of training data. We now proceed with the results for the two hierarchical models.

## ON-LSTM

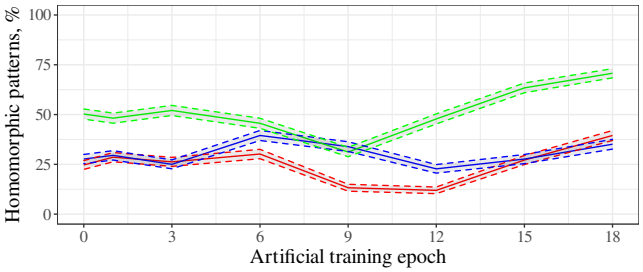
Figure 2 shows the results for the ON-LSTM model. After pretraining (epoch 0) the model's preferences are approximately at chance level in all three conditions. Later in training the preferences diverge: in the NUM-ADJ-N condition (see the green line), the preference stays close to chance level, while in the other two conditions (the red and blue lines) we observe a preference for the homomorphic word order (up to 74.9%).

As in the previous section, we fit a series of mixed-effects logistic regressions to the data, this time separately for epoch 0 and epochs 7–10. This analysis supports the observed patterns: initially the model's preferences are not significantly different from chance (DEM-ADJ-N: intercept is 0.03, $p = .815$; DEM-NUM-N: intercept is 0.03, $p = .863$; NUM-ADJ-N: intercept is $-0.01$, $p = .935$), but later during training the preference for the homomorphic word order is significantly higher than chance in the DEM-ADJ-N and DEM-NUM-N (intercepts are 8.74 and 11.9, respectively; both $p < .001$), but not the NUM-ADJ-N condition (intercept 1.1, $p = .227$). This mirrors the findings of Martin et al. (2020): as we mentioned in the Background section, in their experiments the preference for the homomorphic word order was less reliable in the NUM-ADJ-N condition, compared to the other two. At the same time, it is unclear exactly what causes the differences across conditions in our model.

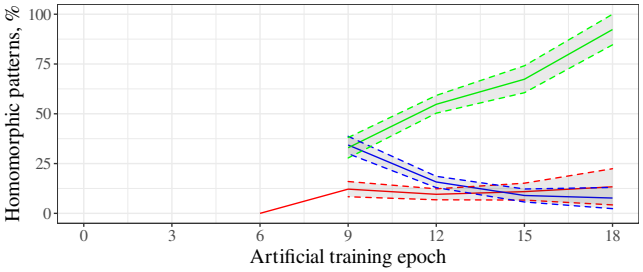
## RNNG

Figure 3a shows the results for the RNNG model. As with the previous models, we again observe differences across the three conditions, but the pattern of preferences is not the same as either in the LSTM or the ON-LSTM. In the NUM-ADJ-N condition (green line), we initially observe no preference, but later in learning the RNNG develops a preference for the homomorphic word order. In contrast, in the DEM-ADJ-N and DEM-NUM-N conditions (red and blue lines, respectively) we see preferences for the non-homomorphic word order throughout the learning. As in the previous sections, mixed-effects logistic regressions with the same predictors fitted to the data from the later epochs (after epoch 12) show that the model's preferences at the later stages of learning differ from chance, and these differences are statistically significant (the intercepts are: $-2.6$ for DEM-ADJ-N, $-2.3$ for DEM-NUM-N, and 1.9 for NUM-ADJ-N, note the different sign in the last condition).

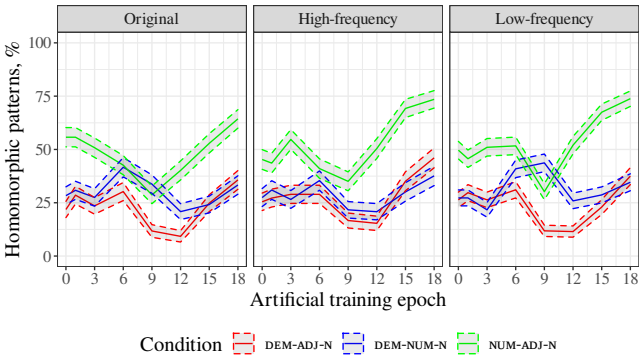
Recall that this computational model is trained to parse the data, and it may be the case that our result is merely an artifact of the model being unable to correctly parse the test sentences. To ensure this is not the case, we ran an additional analysis in which we only considered the RNNG's responses with noun phrases parsed correctly in both sentences. Figure 3b shows the results for this subset of the data: note that many data points are missing from the early epochs, suggesting that the RNNG has not learned to correctly parse the postnominal noun phrases yet. Later in learning, we see that the main qualitative patterns of results for each condition stay the same as for all test sentences, although the preferences are more extreme.




(a) All sentences




(b) Sentences with correctly parsed noun phrases




(c) All sentences, per noun type

Figure 3: Percentage of sentence pairs for which the RNNG model shows preference for the homomorphic word order. Results are averaged over training/test sets and test items, error bands show the standard error of the mean over different test items.

Finally, recall that we had three types of nouns in our test stimuli for this model: the original nouns from Culbertson & Adger (2014), high-frequency, and low-frequency nouns. Figure 3c shows the results per noun type. Again, we see no substantial differences across the noun types, except small differences in the middle of the learning for the DEM-NUM-N condition (compare the blue lines across the three panels).

These additional analyses suggest that the emergent word order preferences in the RNNG model do not arise from the lack of parsing ability and are stable across the nouns of different frequency. Surprisingly, this pattern of homomorphism preference only in the NUM-ADJ-N condition is different from that observed in our ON-LSTM model in the previous section and reported in Martin et al. (2020).

## Discussion

In this study we have asked whether computational language models show a homomorphism bias when inferring the order of nominal modifiers in noun phrases. This bias is consistently observed in human speakers and has been argued to explain why homomorphic orders are more common in the world's languages. We focused on the artificial language learning experiments of Culbertson & Adger (2014) in which this bias was first observed. We first tested an LSTM model without any hierarchical biases in the input or in its architecture. As expected, we found either no preference, or a preference for the non-homomorphic order, depending on the condition. This supports the claim made in Culbertson & Adger (2014) that the word order preferences of human learners do not arise from information about the linear word order of noun modifiers.

We then tested two hierarchical models. The ON-LSTM model was trained on text input and only showed a homomorphism bias in two out of the three conditions we tested (phrases with demonstratives and adjectives, and phrases with demonstratives and numerals). Interestingly, the RNNG model, which was explicitly trained to parse tree structures in the input, showed the target bias in the other condition (phrases with numerals and adjectives).

To answer our main question, these results suggest that hierarchical computational models *can* exhibit a homomorphism bias. At the same time, it is unclear yet what causes the differences between the two models, ON-LSTM and RNNG. Our additional analyses of the RNNG's word order preferences in correctly parsed sentences and for different types of nouns did not shed light on the issue. Note that the two models are very different: in the ON-LSTM, hierarchical relations emerge implicitly thanks to the model's architecture; in the RNNG model, explicit information is provided about the tree structure of every input sequence. Further investigation of the models' biases is needed to interpret the reported differences. One option would be to train and test these two models on multiple versions of artificial languages that differ in their word order outside of the noun phrase (as in White & Cotterell, 2021; Ravfogel et al., 2019). Future research could also examine the nature of the homomorphism bias in our models. Recall that this bias reflects the proposed hierarchy of meanings of various noun modifiers, and this hierarchy has been argued to reflect conceptual structure, grounded in statistical properties of the real world (Culbertson et al., 2020). At the same time, semantic representations in language models are not grounded in the real world, but emerge from the distributional properties of language. Therefore, one can study the models' representations in search of the hierarchy of meanings. A presence of such a hierarchy in a model's representation space would indicate a true bias for homomorphism in that model. If the hierarchy is not found, the model's preferences for hierarchical generalizations must be explained by other factors.

Regarding the differences across conditions in each model, we can speculate that these are due to differences in the distributional information of the relevant elements: depending on the condition, test words and their collocations (bigrams and trigrams) can occur more or less frequently in the pre-training input data, and the interplay of such distributional information with architecture-specific properties of each model may lead to meaningful differences in preferred order. Our analysis of the English input corpus showed that bigram frequencies could explain differences across conditions for the LSTM model, but the preference patterns found in the two hierarchical models need to be examined further in future work.

It is also worth noting that because our language model has no metalinguistic knowledge, it processes artificial language sentences as if they come from English. One the one hand, this helps us to ensure that the models have encoded some aspects of the meaning of the target words before learning the artificial language. On the other hand, this sometimes yields unrealistic results in our simulations, as the models start forgetting English word order and instead show a preference for one of the two word orders with postnominal noun modifiers, a pattern known as catastrophic forgetting.

Finally, our results suggest that out of the three models, the ON-LSTM shows the pattern of word order preferences most similar to human speakers: first, it shows the target bias in two out of the three conditions, and second, the condition in which it does not show the bias (noun phrases with numerals and adjectives) is the one for which the preference in human speakers was found to be the least reliable (Martin et al., 2020). This result suggests that the ON-LSTM may be better than the RNNG in predicting some human-like biases, which contrasts with the findings of McCoy et al. (2020), who found the presence of explicit tree structures in the input (as in the RNNG model) to be an important condition for the model to show human-like hierarchical biases. This warrants a more rigorous evaluation of the two models in future research.

**Acknowledgments**: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 757643). We thank Shira Tal, Elizabeth Nielsen, and four anonymous reviewers for their helpful feedback.

## References

Adger, D. (2003). *Core syntax*. Oxford University Press.

Brennan, J. R., Stabler, E. P., Van Wagenen, S. E., Luh, W.-M., & Hale, J. T. (2016). Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and Language*, *157*, 81–94.

Bybee, J. (2006). From usage to grammar: The mind's response to repetition. *Language*, 711–733.

Chomsky, N. (1957). *Syntactic structures*. De Gruyter.

Christiansen, M. H., & Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, *39*, e62.

Cinque, G. (2005). Deriving Greenberg's Universal 20 and its exceptions. *Linguistic Inquiry*, *36*, 315–332.

Culbertson, J., & Adger, D. (2014). Language learners privilege structured meaning over surface frequency. *PNAS*, *111*, 5842–5847.

Culbertson, J., Schouwstra, M., & Kirby, S. (2020). From the world to word order: Deriving biases in noun phrase order from statistical properties of the world. *Language*, *96*, 696–717.

Dryer, M. (2018). On the order of demonstrative, numeral, adjective and noun. *Language*, *94*, 798–833.

Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of NAACL*.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., & Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of NAACL-HLT*.

Goodkind, A., & Bicknell, K. (2018). Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of CMCL*.

Gulordava, K., Bojanowski, P., Grave, É., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL-HLT*.

Hu, J., Gauthier, J., Qian, P., Wilcox, E., & Levy, R. P. (2020). A systematic assessment of syntactic generalization in neural language models. In *Proceedings of ACL*.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. In *Proceedings of ACL*.

Martin, A., Holtz, A., Abels, K., Adger, D., & Culbertson, J. (2020). Experimental evidence for the influence of structure and meaning on linear order in the noun phrase. *Glossa*, *5*, 97.

Martin, A., Ratitamkul, T., Abels, K., Adger, D., & Culbertson, J. (2019). Cross-linguistic evidence for cognitive universals in the noun phrase. *Linguistics Vanguard*, *5*.

Marvin, R., & Linzen, T. (2018). Targeted syntactic evaluation of language models. In *Proceedings of EMNLP*.

McCoy, R. T., Frank, R., & Linzen, T. (2020). Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, *8*, 125–140.

Noji, H., & Oseki, Y. (2021). Effective batching for recurrent neural network grammars. In *Proceedings of ACL*.

Ravfogel, S., Goldberg, Y., & Linzen, T. (2019). Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of NAACL-HLT*.

Shen, Y., Tan, S., Sordoni, A., & Courville, A. (2018). Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of ICLR*.

Tomasello, M. (2003). *Constructing a language*. Harvard University Press.

Van Schijndel, M., & Linzen, T. (2018). A neural model of adaptation in reading. In *Proceedings of CogSci*.

White, J. C., & Cotterell, R. (2021). Examining the inductive bias of neural language models with artificial languages. In *Proceedings of ACL*.

Widmer, M., Auderset, S., Nichols, J., Widmer, P., & Bickel, B. (2017). NP recursion over time: Evidence from Indo-European. *Language*, *93*, 799–826.