Structure-(in)dependent Interpretation of Phrases in Humans and LSTMs

Cas W. Coopmans
MPI for Psycholinguistics
Centre for Language Studies
cas.coopmans@mpi.nl

Helen de Hoop
Centre for Language Studies
h.dehoop@let.ru.nl

Karthikeya Kaushik
MPI for Psycholinguistics
karthikeya.kaushik@gmail.com

Peter Hagoort
MPI for Psycholinguistics
Donders Institute
peter.hagoort@mpi.nl

Andrea E. Martin
MPI for Psycholinguistics
Donders Institute
andrea.martin@mpi.nl

Abstract

In this study, we compared the performance of a long short-term memory (LSTM) neural network to the behavior of human participants on a language task that requires hierarchically structured knowledge. We show that humans interpret ambiguous phrases, such as *second blue ball*, in line with their hierarchical constituent structure. LSTMs, instead, only do so after unambiguous training data, and they do not systematically generalize to novel items. Overall, the results of our simulations indicate that a model can behave hierarchically without relying on hierarchical constituent structure.

1 Introduction

It has long been recognized that phrases and sentences are interpreted in line with their underlying hierarchical constituent structure (Chomsky, 1957; Everaert et al., 2015). Contrasting with these arguments in theoretical linguistics, however, it has been argued in (computational) psycholinguistics that language *use* is fundamentally sequential (Frank et al., 2012). This claim is strengthened by recent findings from natural language processing, which show that computational models of language learn structure-dependent phenomena, seemingly without invoking hierarchical structure (Linzen et al., 2016).

In response to the claim that language use is fundamentally sequential, we present experimental evidence which suggests that language interpretation might in fact be biased towards hierarchy. Specifically, we used an experimental paradigm based on Hamburger and Crain (1984) to examine whether participants interpret ambiguous noun phrases, such as *second blue ball*, as a hierarchical structure or as a linear string. On the hierarchical interpretation (Figure 1B), the phrase refers to 'the second among blue balls' (fourth ball in Figure 1C). On the linear interpretation (Figure 1A), instead,

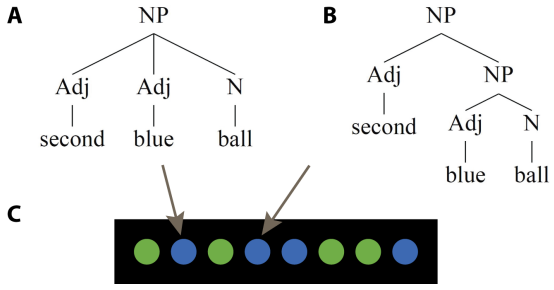

Figure 1: Linear (A) and hierarchical (B) representations for the noun phrase *second blue ball*. (C) presents a picture from the behavioral experiment.

'second' and 'blue' are interpreted conjunctively, referring to 'the ball that is blue *and* second' (second ball in Figure 1C). In a behavioral experiment with such phrases (ordinal, color, shape) and pictures in which the hierarchical and linear interpretations were always both present (e.g. Figure 1C), participants overwhelmingly interpreted these phrases hierarchically.

Having shown a strong bias for hierarchical constituent structure in human language interpretation, we then trained and tested a recurrent neural network (RNN) on a computational version of our task. We tested whether the model would be able to reproduce such 'hierarchical' behavior, and evaluated whether its performance varied as a function of the data on which it was trained. The model received either purely hierarchical training or fully ambiguous training, from which both the hierarchical and the linear interpretations are possible inductions. The ambiguous training-test regime indirectly models language acquisition, for which the input is also compatible with many possible generalizations. The fact that humans consistently arrive at the same generalizations, despite the fact that these generalizations are underdetermined by the input, reflects the "poverty of the stimulus" problem (Chomsky, 1980).

Proceedings of the Society for Computation in Linguistics (SCiL) 2021, pages 459-463.
Held on-line February 14-19, 2021

Recent modeling studies have adopted a poverty-of-the-stimulus approach to look at the inductive biases of RNNs. McCoy et al. (2020), for instance, compared sequential and tree-based RNNs on their ability to learn structure-dependent syntactic phenomena, such as question formation (i.e. converting a declarative into an interrogative sentence). The training data presented to these models were consistent with two generalizations, one of which was based on hierarchical structure (move *main* verb), the other was based on linear order (move *first* verb). The models were then tested on sentences for which these generalizations make different predictions, such as complex sentences with a subject-relative clause, for which the first verb is not the main verb. The only model which showed a bias towards the hierarchical generalization on all syntactic phenomena was the tree-structured model, suggesting that human-like syntactic generalization requires an explicit reference to hierarchical syntactic structure.

In the following paragraphs, we will describe our study, which has a similar poverty-of-the-stimulus logic, but rather than looking at generalization between syntactic forms (i.e. form-only generalization), we look at semantic interpretation (i.e. form-meaning generalization), for which the solution to the learnability problem is similarly underdetermined (Gleitman and Gleitman, 1992).

2 Methods

2.1 Model

We trained and tested an LSTM (Hochreiter and Schmidhuber, 1997) on a computational version of our experimental task. The model's task was to take the phrase and the picture as input, and provide as output the position of the target. The input to the model consisted of four vectors, which were sequentially presented in four time steps (Figure 2). These vectors were one-hot representations of respectively the ordinal, color, and shape of the target, and the picture. Each input vector had a length of 170, where the first 10 elements were reserved for the phrase (elements 1-6 represented the ordinals second through seventh, 7 and 8 represented the colors blue and green, 9 and 10 represented the shapes ball and triangle) and the last 160 elements were reserved for the eight-element picture, wherein each element had a color and a shape. The picture was normalized to make sure that its net content is 1, in line with the other one-hot vectors.

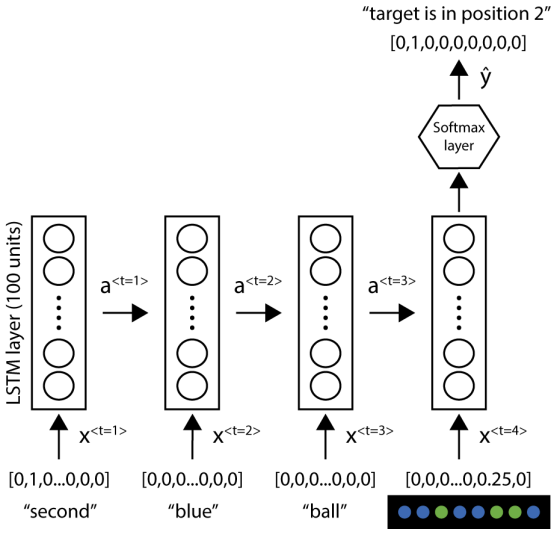

Figure 2: Experimental setup. The LSTM receives four inputs, sequentially presented in time, where $x^{<t>}$ represents the input at time step t and $a^{<t-1>}$ the activation state of the model after the previous time step.

To give an example of an input vector, the word 'blue' would be represented as a 170-element vector which has a one in position 7 and zeros everywhere else.

The hidden layer consisted of 100 units, whose activation function at the last time step was forwarded to a softmax layer, which provided the output of the network. The output was a nine-element one-hot vector which had a one at the position of the target (positions 1-8) on target-present trials or a one at position 9 to indicate that the target was absent from the picture.

We trained the LSTM in a supervised manner on datasets of different types (see Section 2.2) and different sizes (100-1000 trials), in 50 epochs (100 steps per epoch) using the optimizer "Adam" and the categorical-crossentropy loss function. For each dataset, the model was evaluated on 100 test trials, and this training-test evaluation was simulated 100 times.

2.2 Training

We created different training and test datasets, which contained both target-present and target-absent trials. The "hierarchical" training and test set consisted of target-present trials in which only the hierarchical interpretation of the phrase was present (e.g. for the phrase *second blue ball*, the second ball in the figure would not be blue), and target-absent trials in which it was absent. The "ambiguous" training set was fully ambiguous between

the hierarchical and linear interpretations of the target phrase, both on target-present and target-absent trials. For instance, on target-present *training* trials the first two balls were blue (Figure 2), such that the ball that was blue and in second position (linear) was also the second of the set of blue balls (hierarchical). On target-present *test* trials the linear and hierarchical interpretation did not converge on the same target (e.g. Figure 1C). The training data are thus equally compatible with the hierarchical and linear interpretations, but as these two generalizations make different predictions for the test trials, the model's answers on these test trials can reveal its inductive bias in this setup.

3 Results

3.1 Induction of hierarchy

When the model is trained on unambiguously hierarchical data, it learns to give hierarchical answers. Its performance starts at 39% correct on target-present trials after 100 training trials, steadily increases with increasing training size up to 700 trials, and stabilizes around 97-100% correct (Figure 3A). After "ambiguous" training, however, the model mainly gives linear answers on target-present trials (M = 81.1%, SD = 5.51%), and never gives a hierarchical answer (Figure 3B). Moreover, when the same "ambiguous" model is tested on hierarchical test trials, in which the linear answer is not even present, it still never gives a hierarchical answer.

These data suggest that the model does not have a bias to generalize in a hierarchical way, in contrast to what has been argued for humans. One could argue, however, that our ambiguous training regime is not representative of the language input children receive, and therefore does not provide an adequate test of the poverty of the stimulus argument. To make our simulations more representative of natural language acquisition, we ran 100 simulations in which the model was trained on a dataset which was half ambiguous and half unambiguous. On unambiguous training trials, the output is only compatible with the hierarchical interpretation, so overall, the hierarchical interpretation is the only generalization fully compatible with these mixed training data. When tested on unambiguous test trials, the model did give some hierarchical answers (M = 12.9%, SD = 5.66%), yet the majority of its answers were still linear (M = 58.8%, SD = 8.36%). This again suggests that the model can learn to answer "hierarchically", but that it needs unambiguous trials to overcome its non-hierarchical bias (cf. McCoy et al., 2018).

3.2 Generalization to novel items

We then investigated what the model has learned after the hierarchical training regime by evaluating its ability to generalize to completely novel items that were not observed during training, such as *third red ball*. Specifically, we looked the model's response to phrases that included the word 'red' when the training data did not contain red at all (i.e. 'extrapolation'), or only in combination with specific ordinals. In the latter case, the model was trained on all combinations of features except the combination of 'third' with 'red' and 'ball', and was then tested on 'third red ball'. While these features are observed during training, their combination is new and therefore requires 'interpolation'.

As representations of the words in our vocabulary we used one-hot vectors as well as 300-dimensional word embeddings from word2vec (Mikolov et al., 2013) and a dimensionality-reduced version of these word embeddings. We added these word embeddings because they might enhance the model's generalization ability. That is, the one-hot vectors [0,1...0,0] and [0,0...1,0] are fully independent, although they would have

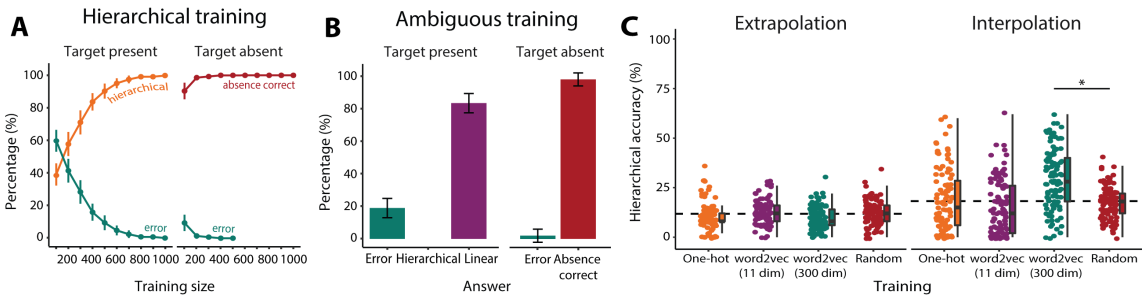

Figure 3: The model's performance on test trials after hierarchical training (A) and ambiguous training (B). Panel (C) represents the percentage of hierarchical responses on both generalization tests after different types of training.

to be related if they are to represent the words 'blue' and 'red'. Word embeddings, instead, do capture this similarity and might therefore aid the model in generalizing (in particular, interpolating). For each type of input vector (one-hot, full and dimensionality-reduced word embeddings) and each generalization test (extrapolation and interpolation), we ran 100 simulations in which the model was trained on 500 training trials and evaluated on 100 test trials, which were identical in each simulation run. The model's overall accuracy was compared to chance level, defined as the accuracy of the model when it was trained on pseudorandom input-output mappings.

Accuracy on both generalization tests was defined as the percentage of hierarchical answers. As Figure 3C shows, the model's accuracy was not above chance level on the extrapolation test in any of the conditions. The chance level of 12.5% reflects the probability of each of the eight possible target positions attested during training. Statistical comparison of the groups indicated that the accuracies in the conditions were different ($F(3,396) = 5.12$, $p = .002$), but pairwise follow-up tests showed that none of the conditions scored above chance. On the interpolation test, the accuracies of the groups were again different ($F(3,396) = 20.4$, $p < .001$), and pairwise follow-up tests showed that only the accuracy for the full word embeddings (M = 29.1%, SD = 15.3%) was higher than expected by chance (Tukey test, $p < .001$). These findings show that the inherent similarity between word embeddings that represent related words, in combination with the statistical information that these words occur in the same distributional environments, allows the model to generalize to a novel combination of words. Still, the model's accuracy varies considerably across simulations, plausibly related to the variability in each training set as well as random initializations of the model in each simulation run. Such stochasticity is strikingly different from the systematic behavior of humans on similar generalization tests (Lake and Baroni, 2018).

4 Discussion

In this study, we investigated whether LSTMs are biased to interpret phrases such as *second blue ball* hierarchically, like humans, or linearly, and how this interpretation is affected by the training data. While we show that an LSTM can learn to give hierarchical answers, comparison of the performance of the model and the behavior of the human participants on our task reveals a number of critical differences. First, while the model learned to give hierarchical answers, it only did so when it was explicitly given hierarchical information during supervised training. When the training data were ambiguous with respect to the correct representation underlying the noun phrases, the model had a strongly linear bias, never giving a hierarchical answer during the test phase. This suggests that without an inductive hierarchical bias, the neural network interprets ambiguous input in the way that is most in line with the simplest statistical information computed over sequences of words (i.e. the linear interpretation).

Second, as shown in previous work, the LSTM did not systematically generalize to items that were not observed during training (Lake and Baroni, 2018; Loula et al., 2018; Puebla et al., 2020). In our simulations, only when the model was trained on full word embeddings, it scored higher than chance level on the interpolation test. However, even on this successful test, the model's average accuracy was about 30% and its performance was variable, indicating that it achieved its performance on hierarchical test trials without resorting to truly hierarchical constituent structure, which is commonly assumed to be built from representations with a symbolic format (cf. Figure 1B; Everaert et al., 2015). The inability to generalize thus suggests that the model does not rely on the type of (symbolic) constituent structure we believe underlies the responses of the human participants (Doumas and Martin, 2018; Martin, 2020).

5 Conclusion

The difficulty with which computational models of language acquire hierarchical behavior is at odds with both the ease with which it is acquired by children as well as with the pervasiveness of structure dependence in natural language. To address this issue, we believe that incorporating inductive biases for hierarchy, perhaps in the form of model architecture (Kuncoro et al., 2018; McCoy et al., 2020), is a sensible next step in computational modeling of language. This will make these models more valid as models of human cognition and might also make their performance more human-like (Martin and Doumas, 2017), especially when presented with limited data (Wilcox et al., 2019).

References

Noam Chomsky. 1957. *Syntactic Structures*. Mouton de Gruyter, The Hague.

Noam Chomsky. 1980. *Rules and Representations*. Columbia University Press, New York, NY.

Leonidas A. A. Doumas and Andrea E. Martin. 2018. Learning structured representations from experience. *Psychology of Learning and Motivation*, 69:165–203.

Martin B. H. Everaert, Marinus A. C. Huybregts, Noam Chomsky, Robert C. Berwick, and Johan J. Bolhuis. 2015. Structures, not strings: Linguistics as part of the cognitive sciences. *Trends in Cognitive Sciences*, 19(12):729–743.

Stefan L. Frank, Rens Bod, and Morten H. Christiansen. 2012. How hierarchical is language use? *Proceedings of the Royal Society B: Biological Sciences*, 279(1747):4522–4531.

Lila R. Gleitman and Henry Gleitman. 1992. A picture is worth a thousand words, but that's the problem: The role of syntax in vocabulary acquisition. *Current Directions in Psychological Science*, 1(1):31–35.

Henry Hamburger and Stephen Crain. 1984. Acquisition of cognitive compiling. *Cognition*, 17(2):85–136.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1426–1436.

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*, pages 2873–2882.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

João Loula, Marco Baroni, and Brenden M. Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114.

Andrea E. Martin. 2020. A compositional neural architecture for language. *Journal of Cognitive Neuroscience*, 1:1–20.

Andrea E. Martin and Leonidas A. A. Doumas. 2017. A mechanism for the cortical computation of hierarchical linguistic structure. *PLOS Biology*, 15(3):e2000663.

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.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Guillermo Puebla, Andrea E. Martin, and Leonidas A. A. Doumas. 2020. The relational processing limits of classic and contemporary neural network models of language processing. *Language, Cognition and Neuroscience*, 1:1–15.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3302–3312.