How poor is the stimulus? Evaluating hierarchical generalization in neural networks trained on child-directed speech

Aditya Yedetore[*1], **Tal Linzen**[2], **Robert Frank**[3], **R. Thomas McCoy**[*4]
[1]Boston University, [2]New York University, [3]Yale University, [4]Princeton University
yedetore@bu.edu, linzen@nyu.edu, robert.frank@yale.edu,
tom.mccoy@princeton.edu

Abstract

When acquiring syntax, children consistently choose hierarchical rules over competing non-hierarchical possibilities. Is this preference due to a learning bias for hierarchical structure, or due to more general biases that interact with hierarchical cues in children's linguistic input? We explore these possibilities by training LSTMs and Transformers—two types of neural networks without a hierarchical bias—on data similar in quantity and content to children's linguistic input: text from the CHILDES corpus. We then evaluate what these models have learned about English yes/no questions, a phenomenon for which hierarchical structure is crucial. We find that, though they perform well at capturing the surface statistics of child-directed speech (as measured by perplexity), both model types generalize in a way more consistent with an incorrect linear rule than the correct hierarchical rule. These results suggest that human-like generalization from text alone requires stronger biases than the general sequence-processing biases of standard neural network architectures.

1 Introduction

Syntax is driven by hierarchical structure, yet we typically encounter sentences as linear sequences of words. How do children come to recognize the hierarchical nature of the languages they acquire? Some argue that humans must have a hierarchical inductive bias—an innate predisposition for hierarchical structure (Chomsky, 1965, 1980). An alternative view (e.g., Lewis and Elman, 2001) is that no such bias is necessary: there may be clear evidence for hierarchical structure in children's input, so that children would choose hierarchical rules even without a hierarchical bias.

At first blush, recent work in natural language processing (NLP) may seem to indicate that no hierarchical bias is necessary. Neural networks trained on naturally-occurring text perform impressively on syntactic evaluations even though they have no explicit syntactic structure built into them (e.g., Gulordava et al., 2018; Wilcox et al., 2018; Warstadt et al., 2020a). However, these results do not provide strong evidence about the learning biases required to learn language from the data available to humans because these models receive very different training data than humans do (Warstadt and Bowman, 2022). First, NLP models are typically trained on far more data than children receive, so models have more opportunities to encounter rare syntactic structures (Linzen, 2020). Second, most training sets in NLP are built from Internet text (e.g., Wikipedia), which differs qualitatively from the utterances that children typically hear; e.g., sentences in Wikipedia are on average 25 words long (Yasseri et al., 2012), compared to 5 words for sentences in the North American English subset of the CHILDES corpus of child-directed speech (MacWhinney, 2000).

In this work, to evaluate if neural networks without a hierarchical bias generalize like children do, we train models on text[1] comparable to the sentences in children's linguistic input: English data from CHILDES. We then analyze what they have learned about the relationship between declarative sentences, such as (1a), and their corresponding yes/no questions, such as (1b):

(1) a. Those **are** your checkers.
b. **Are** those your checkers?

Crucially, nearly all naturally-occurring yes/no questions are consistent with two rules: one based

[*] Work done while at Johns Hopkins University.

[1] Section 6.5 discusses other input types (e.g., visual input).

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9370–9393

on hierarchical structure (2), and one based on linear order (3):[2,3]

(2) HIERARCHICALQ: The auxiliary at the start of a yes/no question corresponds to the **main** auxiliary of the corresponding declarative.

(3) LINEARQ: The auxiliary at the start of a yes/no question corresponds to the **first** auxiliary of the corresponding declarative.

Despite the scarcity of evidence disambiguating these rules, children reliably favor HIERARCHICALQ (Crain and Nakayama, 1987), albeit with occasional errors consistent with LINEARQ (Ambridge et al., 2008). Yes/no questions thus are a prime candidate for an aspect of English syntax for which human-like generalization requires a hierarchical bias. We evaluate yes/no question performance in LSTMs and Transformers, two neural-network architectures that have no inherent hierarchical inductive bias (McCoy et al., 2020; Petty and Frank, 2021). These architectures employ different computational mechanisms, so consistent results across both would indicate that our results are not due to idiosyncrasies of one particular architecture.

To investigate if models generalize more consistently with the hierarchical or linear rule, we evaluate them on cases where the rules make different predictions, such as (4): under HIERARCHICALQ, the question that corresponds to (4a) is (4b), whereas under LINEARQ it is (4c).

(4) a. The boy who **has** talked **can** read.
b. **Can** the boy who **has** talked ___ read?
c. ***Has** the boy who ___ talked **can** read?

We find that across several ways of framing the learning task, models fail to learn HIERARCHICALQ. Instead, they generalize in ways that depend on linear order and on the identities of specific words. These results suggest that children's training data, if taken to be words alone, may not contain enough hierarchical cues to encourage hierarchical generalization in a learner without a hierarchical bias. Thus, explaining human acquisition of syntax may require postulating that humans have stronger inductive biases than those of LSTMs and Transformers, or that information other than word sequences plays a crucial role.[4]

2 Background

Though HIERARCHICALQ and LINEARQ often make the same predictions, the evidence in children's input may still favor HIERARCHICALQ. The most straightforward evidence would be utterances that directly disambiguate the rules, such as (4b). Pullum and Scholz (2002) show that disambiguating examples appear in the *Wall Street Journal*, in literature, and arguably in child-directed speech, but direct evidence may still be too rare to robustly support HIERARCHICALQ (Legate and Yang, 2002). Nonetheless, children might conclude that yes/no questions obey HIERARCHICALQ rather than LINEARQ based on *indirect* evidence—evidence that *other* syntactic phenomena are hierarchical (Mulligan et al., 2021).

To test if the cues favoring HIERARCHICALQ render a hierarchical bias unnecessary, we study how well non-hierarchically-biased models acquire English yes/no questions. Several prior papers have used this approach, but their training data differed from children's input in important ways: some used synthetic datasets (Lewis and Elman, 2001; Frank and Mathis, 2007; Clark and Eyraud, 2007; McCoy et al., 2020), others used massive Internet corpora (Lin et al., 2019; Warstadt and Bowman, 2020), and those that used child-directed speech simplified the data by replacing each word with its part of speech (Perfors et al., 2011; Bod et al., 2012). We used training data closer to children's input, namely sentences from CHILDES with word identities preserved, rather than being converted to parts of speech. Two other recent works have also trained neural networks on CHILDES data (Pannitto and Herbelot, 2020; Huebner et al., 2021), but neither investigated yes/no questions.

One particularly important reason for training models on CHILDES is that, in prior work, different types of training data have yielded diverging results: Recent models trained on synthetic data failed to properly acquire yes/no questions (McCoy et al., 2020; Petty and Frank, 2021), whereas ones trained on large Internet corpora scored well on evaluations of yes/no questions (Lin et al., 2019; Warstadt and Bowman, 2020). Given these differing results, it is not clear from past work how these

[2] In past work these rules have been framed as transformations named MOVE-FIRST and MOVE-MAIN (McCoy et al., 2020). We instead follow Berwick et al. (2011) and frame the child's knowledge as a relationship between sentences.

[3] Though these two rules are the most prominent in prior literature, other rules are possible; see Section 5.2.

[4] GitHub repo with data and code: https://github.com/adityayedetore/lm-povstim-with-childes.

models would generalize when faced with the type of data that children receive.

3 Overview of Experimental Setup

We evaluated models on yes/no questions in two ways. First, we used relative acceptability judgments (Experiment 1): We trained neural networks on the task of language modeling (predicting the next word at every point in the sentence) and evaluated whether they assigned a higher probability to sentences consistent with LINEARQ or HIERARCHICALQ. Our second approach was based on text generation (Experiment 2): We trained networks to take in a declarative sentence and output the corresponding question, and tested whether they generalized in a way more consistent with LINEARQ or HIERARCHICALQ. Under both framings, we trained models on data from CHILDES and evaluated them on targeted datasets constructed to differentiate LINEARQ and HIERARCHICALQ.

4 Experiment 1: Relative Acceptability

4.1 Dataset

To train models on data as similar as possible to the sentences children receive, we extracted data from CHILDES (MacWhinney, 2000). We used the North American English portion. We wished to replicate children's *input*, so we excluded the children's own utterances, leaving a 9.6-million-word corpus. We allocated 90% of the data to training, 5% to validation, and 5% to testing. We replaced words that appeared two or fewer times in the training set with <unk>, giving a replacement rate of 0.3%. See Appendix A for more details.

4.2 Task: Next-Word Prediction

We trained models on next-word prediction, also known as language modeling. We chose this task for two reasons. First, it is clear empirically that next-word prediction can teach neural networks a substantial amount about syntax (e.g., Hu et al., 2020). Second, it is plausible that humans perform some version of next-word prediction during sentence processing (Altmann and Kamide, 1999; Hale, 2001; Levy, 2008; Kutas et al., 2011) and that such prediction may play a role in acquisition (Elman, 1991). Thus, while next-word prediction is certainly not the only goal of human language learners, we view this task as a reasonable first step in emulating human language acquisition.

4.3 Architectures

We used two neural network architectures: LSTMs (Hochreiter and Schmidhuber, 1997) and Transformers (Vaswani et al., 2017). We chose these models for two reasons. First, they have been the most successful architectures in NLP. Thus, we have reason to believe that, of the types of low-bias models invented, these two are the ones most likely to discover linguistic regularities in our CHILDES training data. Second, the two architectures process sequences very differently (via recurrence vs. via attention). Thus, if both generalize similarly, we would have evidence that what was learned is strongly evidenced in the data, rather than due to a quirk of one particular architecture.

For our LSTMs, we used 2 layers, a hidden and embedding size of 800, a batch size of 20, a dropout rate of 0.4, and a learning rate of 10. For our Transformers, the corresponding values were 4, 800, 10, 0.2, and 5, and we used 4 attention heads. We chose these values based on a hyperparameter search described in Appendix B. All following results are averaged across 10 runs with different random seeds.

4.4 Results: Language Model Quality

Before testing models on questions, we used perplexity to evaluate how well they captured the basic structure of their training domain. As a baseline, we used a 5-gram model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with KenLM (Heafield, 2011). The test set perplexity for the 5-gram baseline was 24.37, while the average test set perplexity for the LSTMs and Transformers was 20.05 and 19.69, respectively. For perplexity, lower is better. Thus, both neural network types outperformed the strong baseline of a smoothed 5-gram model, showing that they performed well at capturing the basic statistics of their training domain.[5]

4.5 General Syntactic Evaluation

As an additional way to check the validity of our setup, we evaluated our models on the Zorro dataset (Huebner et al., 2021), which is based on BLiMP (Warstadt et al., 2020a). Zorro contains 24 evaluations, each of which targets one syntactic phenomenon (e.g., subject-verb agreement) and involves sentence pairs for which one sentence is grammatical, and the other is minimally different

[5]For an intuitive illustration of our model quality, see the sample text generated by them in Appendix H.

but ungrammatical (e.g., by violating subject verb agreement). A model is said to get a sentence pair correct if it assigns a higher probability to the grammatical sentence than the ungrammatical one. Huebner et al. (2021) showed that Transformers trained on CHILDES data can perform well on many of the Zorro categories, so if our setup is sound, our own models should also perform well on Zorro.

See Appendix D for full results. For each syntactic phenomenon, most model re-runs scored above 0.9, though at least one scored near the chance level of 0.5. For each re-run of each architecture there is at least one phenomenon for which the model scores over 0.97, and many models score 1.00 on some phenomena. Thus, all models score well on at least some syntactic evaluations, attaining results comparable to those of Huebner et al. (2021) and providing additional support for the validity of our setup. We now test whether these models have also successfully learned the specific phenomenon that we focus on, yes/no questions—a phenomenon not included in the Zorro dataset.

4.6 Yes/No Questions

Evaluation Dataset: Forced-Choice Acceptability Judgments As a first way to test whether our models have learned HIERARCHICALQ, we evaluate whether they assign higher probabilities to sentences consistent with HIERARCHICALQ than to minimally different sentences that are ungrammatical. For this purpose, we create an evaluation dataset containing groups of 6 questions, each created by starting with a declarative sentence, such as (5), and then deleting the **first**, **main**, or neither auxiliary, and inserting the **first** or **main** auxiliary at the front of the sentence.[6] For instance, in (6b), the **first** auxiliary has been preposed, and the **main** auxiliary has been deleted.

(5) The dog who **has** seen a boy **did** try.

(6) a. **Has** the dog who seen a boy **did** try?
b. **Has** the dog who **has** seen a boy try?
c. **Has** the dog who **has** seen a boy **did** try ?
d. **Did** the dog who seen a boy **did** try?
e. **Did** the dog who **has** seen a boy try?
f. **Did** the dog who **has** seen a boy **did** try?

[6]It would be possible to also use a 'prepose other' category, where an auxiliary not in the input is inserted (McCoy et al., 2018). We excluded this category because using it would raise complications about which 'other' auxiliary to choose.

Within each group, we evaluate which question the model assigned the highest probability to. If a model has correctly learned HIERARCHICALQ, it should assign the highest probability to the question consistent with this rule, such as (6e).

Several past papers about yes/no questions have used the same general approach (Lewis and Elman, 2001; Reali and Christiansen, 2005). However, these papers considered only pairs of sentences, whereas we consider groups of 6 to allow for a wider range of possible generalizations that a model might have learned.

To generate the declaratives from which we formed groups of 6 questions, we used the context-free grammar (CFG) in Appendix F, which has a vocabulary selected from the most common words in CHILDES. Each declarative generated by the CFG (e.g., (5)) contains two auxiliary verbs: one before the sentence's main verb and one inside a relative clause modifying the subject. One potential problem is that some questions are consistent with both HIERARCHICALQ and LINEARQ. For instance, (7a) can be formed from (7b) with the HIERARCHICALQ-consistent steps PREPOSE-MAIN,DELETE-MAIN, or from (7c) with the LINEARQ-consistent steps PREPOSE-FIRST,DELETE-MAIN.

(7) a. Did the boy who did see the person laugh?
b. The boy who did see the person did laugh.
c. The boy who did see the person can laugh.

To avoid this problem, we required that the auxiliary before the main verb must select for a different verb inflection than the one in the relative clause. For instance in (5), **did** selects for the verb's bare form, while **has** selects for the past participle form. Thus, the auxiliary at the start of the question could only correspond to whichever auxiliary in the declarative has the same selectional properties.[7]

Results: Relative Question Acceptability For each sentence group, we used per-word perplexity to see which of the 6 candidates the models scored most highly.[8] For both LSTMs and Transformers, the correct category (PREPOSE MAIN, DELETE MAIN) was the second-rarest choice, and

[7]A model could succeed on this dataset with a rule that relates the auxiliary at the start of a question with the *last* auxiliary in the declarative form. Since our models fail on this dataset, this consideration is not relevant here.

[8]We also explored evaluation of the models with a more complex measure called SLOR where we additionally normalized scores by word frequency (Pauls and Klein, 2012). Both metrics produced qualitatively similar results, so we only report the simpler metric here. See Appendix C.1.

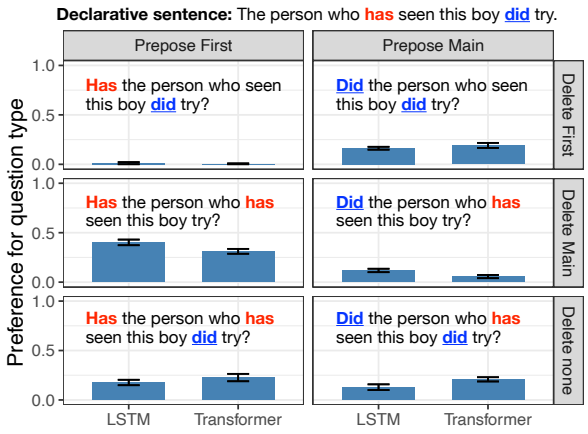

Figure 1: The question types that models prefer when offered a choice between 6 questions. These 6 questions are formed by modifying a declarative with a relative clause on the subject according to 'prepose' and 'delete' rules. The correct category is PREPOSE MAIN, DELETE MAIN. Within each architecture, the proportions across all 6 question types necessarily sum to 1. Each bar shows the average across 10 model re-runs, with single-standard-deviation error bars.

the most frequent preference was for PREPOSE FIRST, DELETE MAIN, a category that is only partially correct because it references linear order in addition to hierarchical structure (Figure 1).

Thus, neither model displays preferences consistent with the correct, fully-hierarchical generalization. The two model types showed similar scores, which may mean that these results are largely driven by the statistics of the training data that both models share, rather than the models' differing inductive biases.

One of the incorrect categories—PREPOSE MAIN, DELETE NONE, such as (6f)—only requires reference to hierarchical structure, so it could be said to capture the hierarchical nature of yes/no questions. Nonetheless, this category was also relatively rare: combining the two fully hierarchical possibilities (PREPOSE MAIN, DELETE MAIN and PREPOSE MAIN, DELETE NONE) accounts for only 26% of LSTM preferences and 27% of Transformer preferences, meaning that both models over 70% of the time favored a sentence generated at least partially based on linear order.

There are two likely reasons for why our models performed so poorly on yes-no questions when they performed well on many of the phenomena in the Zorro dataset (Section 4.5). First, yes/no questions may simply be harder to learn than the other phenomena; indeed, yes/no questions are often singled out as being likely to pose difficulties for a general-purpose learner (Section 1). While this focus in prior literature might simply be a historical coincidence, it is also possible that it points to a true difference in ease of learning. Alternatively, it might be that the six-way evaluation we used for yes/no questions is stricter than the binary judgments used for the Zorro dataset.

5 Experiment 2: Question Formation

The previous experiment was designed to operate entirely in the next-word-prediction paradigm, motivated by arguments from past literature about the strength and relative ecological validity of next-word-prediction as a training objective (see Section 4.2). However, one of this setup's shortcomings is that HIERARCHICALQ describes correspondences between questions and declaratives, but Experiment 1 focused on questions alone, with no consideration of declaratives.

In this second experiment, to better capture that HIERARCHICALQ is defined over sentence pairs, we trained models on a sentence-pair task: transforming a declarative into a question (McCoy et al., 2020). For instance, given *the child did learn* the model must produce *did the child learn ?*

We evaluated models in two ways. First, we checked if the models' predictions fully matched the correct questions. This full-sentence evaluation is demanding, and models might fail this evaluation for reasons unrelated to our core hypotheses. For instance, given *the child did learn* the model might produce *did the baby learn*, which would be marked as incorrect, even though this lexical error is not relevant to HIERARCHICALQ.

As a metric that is less demanding and that also more directly targets HIERARCHICALQ, we measured if the first word of the output question corresponded to the first or main auxiliary of the input. Critically, LINEARQ and HIERARCHICALQ make different predictions for the first word of a question so long as the two auxiliaries are distinct: see (4). Because this framing lets the model freely generate its output (instead of choosing one option from a pre-specified set), we allow for the possibility that the rule learned by models may not be identical to any of our manually-generated hypotheses.

Solely training models to perform this transformation involves the implicit assumption that, when children acquire English yes/no questions, the only evidence they leverage is English yes/no questions.

However, other types of sentences may also provide useful evidence (Pearl and Mis, 2016): e.g., *wh*-questions also illustrate subject-auxiliary inversion (Pullum and Scholz, 2002), while, more generally, many types of sentences could provide evidence that syntax as a whole is hierarchical (Perfors et al., 2011). To explore this possibility, we compared a condition in which models were only trained to perform question formation (the QUESTION FORMATION condition) to another in which models were first pre-trained on next-word prediction with the exact same setup as in Experiment 1 before being further trained to perform question formation (the NEXT-WORD PREDICTION + QUESTION FORMATION condition).

5.1 Dataset

Training Set Our question formation dataset consisted of the yes/no questions in the CHILDES Treebank (Pearl and Sprouse, 2013a,b), a parsed subset of CHILDES containing 189,359 sentences. We used these parses to extract all yes/no questions from the CHILDES Treebank and derive their corresponding declarative forms. The resulting declarative was concatenated with the question. An example declarative/question pair is:

(8) `you can spell your name . can you spell your name ?`

The training set consisted of 10,870 declarative/question pairs, the validation set 1,360 pairs, and the test set 1,358 pairs (we will call this test set the *randomly-partitioned test set* to distinguish it from two other evaluation sets discussed below). We trained models to perform next-word prediction on such concatenated sentence pairs.

The first-word accuracy of the trained model was then computed based on the model's prediction for the word after the period in each test example, while the full-sentence accuracy was computed based on its predictions for all tokens after the period. All questions in the randomly-partitioned test set were withheld from both the question-formation training set and the next-word-prediction training set. Thus, models had not seen these test examples in their training, even in the NEXT-WORD PREDICTION + QUESTION FORMATION condition in which they were trained on both tasks.

Evaluation Sets In addition to the randomly-partitioned test set, we used CFGs to generate two targeted evaluation sets. As in Experiment 1, we selected the CFGs' vocabulary from common words in our CHILDES data. In sentences generated from the first CFG, the sentence's first auxiliary was also its main auxiliary, so LINEARQ and HIERARCHICALQ make the same predictions. (9a) exemplifies the type of declarative-question pair in this dataset. We call this dataset FIRST-AUX = MAIN-AUX. For sentences generated by the second CFG, the main auxiliary was the *second* auxiliary in the sentence; thus, these examples disambiguate LINEARQ and HIERARCHICALQ. Example (9b) is a declarative-question pair from this evaluation set. We call this dataset FIRST-AUX $\neq$ MAIN-AUX. See Appendix F for the CFGs used.

(9) a. `a girl was playing . was a girl playing ?`
    b. `a boy who is playing can try . can a boy who is playing try ?`

5.2 Results

Randomly-Partitioned Test Set The LSTMs and Transformers in the QUESTION FORMATION condition performed well on the randomly-partitioned test set, with a full-question accuracy of 0.68 ± 0.014 and 0.87 ± 0.005 (averaged across 10 reruns with margins indicating one standard deviation). The models in the NEXT-WORD PREDICTION + QUESTION FORMATION condition performed similarly well, with a full-question accuracy of 0.66 ± 0.008 for the LSTMs and 0.93 ± 0.004 for the Transformers. For both model types, the first-word accuracy for the question was nearly 1.00 across re-runs. We suspect that Transformers have a stronger full-question accuracy because producing the question requires copying all words from the declarative (but in a different order). Copying is likely easy for Transformers because they can attend to specific words in the prior context, while our LSTMs must compress the entire context into a fixed-size vector, which may degrade the individual word representations. Because both model types achieved near-perfect performance on the crucial first-word accuracy metric, we conclude that our models have successfully learned how to handle the types of declarative/question pairs that we extracted from the CHILDES Treebank.

Targeted Evaluation Sets On our targeted evaluation sets, models seldom produced the complete question correctly. On the more lenient measure of first-word accuracy, for cases where LINEARQ and HIERARCHICALQ predict the same first output word (FIRST-AUX = MAIN-AUX), the Transformer trained only on question formation per-

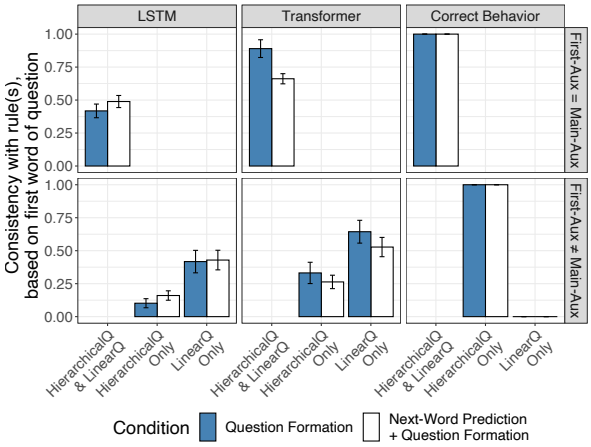

Figure 2: Proportion of model-produced or target questions that were consistent with the linear rule LINEARQ and/or the hierarchical rule HIERARCHICALQ. In the FIRST-AUX = MAIN-AUX dataset, the first auxiliary is the main auxiliary, so both LINEARQ and HIERARCHICALQ produce the correct question string. The FIRST-AUX ≠ MAIN-AUX dataset disambiguates the two rules. Each bar in the LSTM and Transformer facets shows the average across 10 model re-runs, with error bars showing one standard deviation. The Correct Behavior column shows how a model would perform if it had perfectly learned the correct generalization.

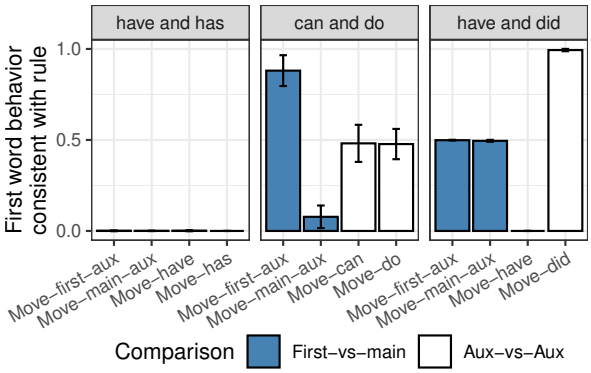

Figure 3: Lexical specificity in model behavior. Each facet considers only the evaluation examples containing the two auxiliaries in the facet heading; e.g., the *can and do* facet includes, for example, the inputs *the children who **can** play **do** learn* and *the children who **do** play **can** learn*. The bars show the proportion of model predictions for the first word of the output that are consistent with four potential movement rules, averaged across 10 model re-runs and with error bars showing one standard deviation above and below the mean. This plot only shows an illustrative subset of auxiliary pairs for one model type (Transformers in the NEXT-WORD PREDICTION + QUESTION FORMATION condition); see Appendix G for the full results.

formed strongly, while the Transformer trained on both tasks, and both LSTMs, performed decently (Figure 2; note chance performance is 1/vocabulary size, which is near 0.00). For cases that disambiguate the two rules (FIRST-AUX ≠ MAIN-AUX), both models in both conditions performed more consistently with LINEARQ than HIERARCHICALQ. Training on next-word prediction before question formation had inconsistent effects: it modestly increased the chance of hierarchical behavior in LSTMs, and decreased it in Transformers.

Lexical Specificity In Appendix G, we further break down the FIRST-AUX ≠ MAIN-AUX results based the auxiliaries' identity. The generalization pattern varied considerably across auxiliary pairs. For some auxiliary pairs, the auxiliary chosen to begin the question was usually neither auxiliary in the input (Figure 3, left facet). For other pairs, models usually chose the first auxiliary, regardless of lexical identity (Figure 3, middle facet). Finally, for some pairs, the auxiliary chosen was usually the same one, regardless of whether it was the first or main auxiliary (Figure 3, right facet).

Generalization based on lexical identity is rarely considered in past discussions of English yes/no question acquisition. Of the papers on this phenomenon (see Clark and Lappin (2010), Lasnik and Lidz (2017), and Pearl (2021) for overviews), the only one to our knowledge that discusses lexical specificity is Frank and Mathis (2007), which studied models trained on synthetic data. Our results highlight the importance of testing for a broad range of generalizations: Lexically-specific hypotheses appear attractive for our low-bias learners, so an account of what biases can yield human-like learning should rule out these lexically-specific hypotheses along with linear ones.

6 Discussion

We have found that, when trained on child-directed speech, two types of standard neural networks performed reasonably well at capturing the statistical properties of the dataset, yet their handling of English yes/no questions was more consistent with a linear rule LINEARQ than the correct hierarchical rule HIERARCHICALQ. These results support the hypothesis that a learner requires a hierarchical bias to consistently learn hierarchical rules when learning from the linguistic data children receive.

6.1 Takeaways for LSTMs and Transformers

When trained on massive corpora, LSTMs and Transformers perform impressively on some syn-

tactic evaluations. Based on such results, it is tempting to conclude that the general-purpose biases of these architectures suffice to yield human-like syntax acquisition. Our results caution against this interpretation: When we trained the same architectures on data more similar to children's input, they failed to learn the structure of English yes/no questions. Thus, at least when learning from text alone, LSTMs and Transformers do not display human-like language learning—they do not generalize as humans do *from the data that humans receive*.

6.2 Takeaways for the Poverty of the Stimulus Debate

Below we specify four possible positions in the poverty-of-the-stimulus debate about the adequacy of children's input for inducing hierarchical rules in low-bias learners, arranged from assuming the most limited to the most expansive innate component:

(10) **Any inductive biases:** Any learner trained on CHILDES will generalize like humans do.

(11) **Any inductive biases that enable in-distribution learning:** Any learner that captures the statistical patterns of the training distribution will generalize to HIERARCHICALQ.

(12) **Some non-hierarchical inductive biases:** Some general-purpose learners will generalize as humans do, but others will not.

(13) **Only a hierarchical inductive bias:** No general-purpose learners will generalize as humans do: hierarchical biases are necessary.

Position (10) is clearly false: many learners cannot learn certain aspects of syntax, no matter their training data (e.g., bigram models cannot capture long-distance dependencies). Our work shows that position (11) is also false: Though our models performed well on the in-distribution test sets of Experiments 1 and 2, they did not generalize in human-like ways. This leaves positions (12) and (13), which our existing results cannot differentiate. It is possible that only learners with hierarchical inductive biases can demonstrate human-like language learning (position (13)), but also that some learners without this bias can succeed (position (12))—just not the learners we tested. For further discussion of how computational modeling can bear on learnability arguments, see Wilcox et al. (2022).

One potential solution supporting position (12) would be that learners leverage the hierarchical structure of some syntactic phenomenon to help conclude that other, impoverished phenomena are hierarchical (Perfors et al., 2011; Mulligan et al., 2021). However, our results from Experiment 2 show that giving learners access to a wider range of phenomena does not automatically improve hierarchical generalization: Models' performance on question formation was not substantially improved (and in some cases was even harmed) when they were trained not just on question formation but also on next-word prediction on the entire CHILDES corpus. Thus, although training on text that contains many linguistic phenomena can give models a hierarchical inductive bias when the training is done over large Internet corpora (Warstadt and Bowman, 2020; Mueller et al., 2022), our results provide evidence that this conclusion does not extend to models trained on child-directed speech.

Though both (12) and (13) remain as possibilities, we believe that our results more strongly support (13). Of all currently available general-purpose learners, LSTMs and Transformers are the best at modeling the probabilistic structure of linguistic data. Therefore, if child-directed speech contains clear evidence for the hierarchical nature of yes/no questions—evidence so clear that at least some general-purpose learners could recognize it—it is likely that LSTMs and Transformers would be among the set of general-purpose learners that could use this evidence to make hierarchical generalizations in our experiments. The fact that these architectures instead predominantly favored linear generalizations therefore supports position (13).

6.3 How to test for HIERARCHICALQ

We have argued that an ideal simulation of the acquisition of English yes/no questions would have the following properties:

(14) The training data should be similar to children's linguistic input.

(15) The training task should be ecologically valid.

(16) The evaluation method should focus on correspondences between pairs of sentences rather than the acceptability of individual sentences.

Property (14) motivated our use of text from CHILDES as the training data. We are not aware of a single experimental setup that fully satisfies both Property (15) and Property (16), so we instead used two experiments, each one focusing on one property at the cost of satisfying the other one less

well. Experiment 1 works entirely in the context of the relatively ecologically valid task of next-word prediction, motivated by Property (15), but its evaluation is only based on the acceptability of individual sentences, failing to satisfy Property (16). Experiment 2 fully satisfies Property (16) by using an evaluation based on sentence pairs, at the cost of including a less ecologically-valid training component based on sentence transformations. Both experiments yielded qualitatively similar conclusions (failure of models to learn HIERARCHICALQ).

6.4 Quantity of Training Data

The size of our training set was plausibly in the range from which children can acquire HIERARCHICALQ. Crain and Nakayama (1987) found that children between ages 3 and 5 behaved much more consistently with HIERARCHICALQ than LINEARQ. Though these children made many errors, their errors were usually compatible with a hierarchical rule (e.g., PREPOSE MAIN, DELETE NONE errors: see Section 4.6). By age 3, American children receive approximately 10 to 33 million words of input (Hart and Risley, 1995), and the 8.5 million of our training set is near the lower end of that range. Thus, while we cannot be completely certain, it is reasonable to suppose that a learner that generalizes as children do would favor HIERARCHICALQ after being trained on our training set. Our models, in contrast, preferred sentences generated in ways based on linear order (Figures 1 and 2), a error category very rare in children (Crain and Nakayama, 1987; Ambridge et al., 2008).

In order to give our models the strongest chance of generalizing correctly, it would have been ideal to provide a quantity of data closer to 33 million words, the high end of Hart and Risley's range. Our data source did not contain enough text to make this possible, but future work could investigate ways to augment the data using other sources.

6.5 Type of Training Data

Our training set was both qualitatively and quantitatively closer to children's input than the massive Internet corpora standardly used to train models in NLP (Linzen, 2020). This difference is important: Lin et al. (2019), Warstadt and Bowman (2020), and Mueller et al. (2022) all found evidence that models trained on large Internet corpora performed well on yes/no questions evaluations, whereas our models trained on CHILDES performed poorly—though we cannot be certain the differences in results are solely due to differences in the training data, since these prior papers used different model architectures, training tasks, and evaluation setups.

Though our training data are more similar to children's input than massive Internet corpora are, differences remain. Our experiments omit several aspects of a child's experience that might help them acquire syntax, such as prosody (Morgan and Demuth, 1996), visual information (Shi et al., 2019), meaning (Fitz and Chang, 2017; Abend et al., 2017), and social interaction (Kuhl et al., 2003; Rowe and Weisleder, 2020), all of which involve information that might correlate with syntactic structure and thus provide cues to the correct hierarchical generalization. On the other hand, our dataset might present an easier learning scenario than children are faced with, because children must learn to segment the speech stream into words (Lakhotia et al., 2021), while our models do not need to. Further, though real-world grounding could provide helpful information, learners might struggle to leverage this information due to difficulty determining what is being discussed in the physical world (Gleitman et al., 2005).

7 Conclusion

In this work, we trained two types of neural networks (LSTMs and Transformers) on sentences of the types available to children and then analyzed what they had learned about English yes/no questions. Across several evaluation paradigms, these models failed to generalize in human-like ways: Humans display hierarchical generalization, while the models' generalization was instead based on linear order and individual words' identities. Our results support the hypothesis that human-like linguistic generalization requires biases stronger than those of LSTMs and Transformers. Future work should investigate what inductive biases enable successful generalization. One approach would be to test architectures with built-in hierarchical structure; past work has shown that such architectures have a hierarchical bias (McCoy et al., 2020) and generalize better on the hierarchical phenomenon of subject-verb agreement (Kuncoro et al., 2018; Lepori et al., 2020), so they may also generalize better on English yes/no questions. A final direction would be to expand the input beyond words alone so that learners can leverage hierarchical structure that is present in other modalities, such as hierarchical structure in visual scenes.

Ethics Statement

Use of human data: While we did not collect any new human data ourselves, many of our analyses involved the use of prior datasets within the CHILDES database. All of these datasets were collected in accordance with IRB policies at the institutions of the data collectors, and all followed standard practices in obtaining informed consent and deidentifying data.[9]

Limitations

We view strong performance on our evaluation datasets as necessary but not sufficient to demonstrate human-like learning. Thus, if models perform poorly on our datasets (as the models we evaluated did), then we have strong reason to conclude that models are not learning in human-like ways. If future models perform better, such results would be consistent with human-like learning but would not conclusively establish that models learn as humans do, as they might instead be using some shallow heuristic that is not controlled for in our datasets. In other words, a criterion that is necessary but not sufficient facilitates strong conclusions about failure but does not facilitate strong conclusions about success. If future papers are faced with models that are more successful, such papers would ideally supplement results based on our datasets with analyses of models' internal strategies in order to more conclusively establish that what they have learned is not a spurious heuristic.

Thus an important risk of our proposed analyses is that future work using the same analyses might draw overly strong conclusions based on increased model performance, leading to overestimates of model strength. Such overestimates are an issue because they can lead users to place more trust in a model than is warranted.

Acknowledgments

For helpful comments and discussion, we are grateful to Najoung Kim, An Nguyen, Grusha Prasad, Paul Smolensky, Paul Soulos, and the NYU Computation and Psycholinguistics Lab. Any errors are our own. We are also grateful to the Maryland Advanced Research Computing Center (MARCC) for providing the computing resources used in our experiments.

Portions of this research were supported by the National Science Foundation (NSF) under grants BCS-2114505, BCS-1919321, and Graduate Research Fellowship Program grant no. 1746891. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[9] https://talkbank.org/share/irb/

References

Omri Abend, Tom Kwiatkowski, Nathaniel J Smith, Sharon Goldwater, and Mark Steedman. 2017. Bootstrapping language acquisition. *Cognition*, 164:116–143.

Gerry TM Altmann and Yuki Kamide. 1999. Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264.

Ben Ambridge, Caroline F Rowland, and Julian M Pine. 2008. Is structure dependence an innate constraint? New experimental evidence from children's complex-question production. *Cognitive Science*, 32(1):222–255.

Robert Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive science*, 35:1207–42.

Rens Bod, Margaux Smets, et al. 2012. Empiricist solutions to nativist problems using tree-substitution grammars. *Workshop on Computational Models of Language Acquisition and Loss: EACL*.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press.

Noam Chomsky. 1980. *Rules and Representations*. Columbia University Press.

Alexander Clark and Rémi Eyraud. 2007. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8(8).

Alexander Clark and Shalom Lappin. 2010. *Linguistic Nativism and the Poverty of the Stimulus*. John Wiley & Sons.

Stephen Crain and Mineharu Nakayama. 1987. Structure dependence in grammar formation. *Language*, pages 522–543.

Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2):195–225.

Hartmut Fitz and Franklin Chang. 2017. Meaningful questions: The acquisition of auxiliary inversion in a connectionist model of sentence production. *Cognition*, 166:225–250.

Robert Frank and Donald Mathis. 2007. Transformational networks. *Models of Human Language Acquisition*, pages 22–27.

Lila R Gleitman, Kimberly Cassidy, Rebecca Nappa, Anna Papafragou, and John C Trueswell. 2005. Hard words. *Language learning and development*, 1(1):23–64.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Betty Hart and Todd R Risley. 1995. *Meaningful differences in the everyday experience of young American children.* Paul H Brookes Publishing.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

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

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Cynthia Fisher, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of CoNLL.*

Xuân-Nga Cao Kam, Iglika Stoyneshka, Lidiya Tornyova, Janet D Fodor, and William G Sakas. 2008. Bigrams and the richness of the stimulus. *Cognitive Science*, 32(4):771–787.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1:181–184 vol.1.

Patricia Kuhl, Feng-Ming Tsao, and Huei-Mei Liu. 2003. Foreign-language experience in infancy: Effects of short-term exposure and social interaction on phonetic learning. *Proceedings of the National Academy of Sciences*, 100:9096–101.

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 (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Marta Kutas, Katherine A DeLong, and Nathaniel J Smith. 2011. A look around at what lies ahead: Prediction and predictability in language processing. In *Predictions in the brain: Using our past to generate a future*, pages 190–207. Oxford University Press.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Howard Lasnik and Jeffrey L Lidz. 2017. The argument from the poverty of the stimulus. *The Oxford handbook of universal grammar*, pages 221–248.

Julie Anne Legate and Charles D Yang. 2002. Empirical re-assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):151–162.

Michael Lepori, Tal Linzen, and R. Thomas McCoy. 2020. Representations of syntax [MASK] useful: Effects of constituency and dependency structure in recursive LSTMs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3306–3316, Online. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

John Lewis and Jeffrey Elman. 2001. Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. *Proceedings of the 26th Annual Boston University Conference on Language Development*, 1.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates.

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 Meeting of the Cognitive Science Society*, Madison, WI.

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.

James L. Morgan and Katherine Demuth. 1996. *Signal to syntax: Bootstrapping from speech to grammar in early acquisition*. Psychology Press.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.

Karl Mulligan, Robert Frank, and Tal Linzen. 2021. Structure here, bias there: Hierarchical generalization by jointly learning syntactic transformations. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 125–135, Online. Association for Computational Linguistics.

Ludovica Pannitto and Aurélie Herbelot. 2020. Recurrent babbling: Evaluating the acquisition of grammar from limited input data. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 165–176, Online. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. Association for Computational Linguistics.

Lisa Pearl. 2021. Poverty of the stimulus without tears. *Language Learning and Development*, pages 1–40.

Lisa Pearl and Benjamin Mis. 2016. The role of indirect positive evidence in syntactic acquisition: A look at anaphoric one. *Language*, 92:1–30.

Lisa Pearl and Jon Sprouse. 2013a. Computational models of acquisition for islands. In *Experimental syntax and island effects*, pages 109–131. Cambridge University Press.

Lisa Pearl and Jon Sprouse. 2013b. Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, 20(1):23–68.

Andrew Perfors, Josh Tenenbaum, and Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition*, 118:306–338.

Jackson Petty and Robert Frank. 2021. Transformers generalize linearly. ArXiv:2109.12036.

Geoffrey K. Pullum and Barbara C. Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 18(1-2):9–50.

Florencia Reali and Morten H. Christiansen. 2005. Uncovering the richness of the stimulus: Structure dependence and indirect statistical evidence. *Cognitive Science*, 29(6):1007–1028.

Meredith L. Rowe and Adriana Weisleder. 2020. Language development in context. *Annual Review of Developmental Psychology*, 2(1):201–223.

Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. 2019. Visually grounded neural syntax acquisition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1842–1861, Florence, Italy. Association for Computational Linguistics.

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

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? *Proceedings of the 42nd Annual Conference of the Cognitive Science Society.*

Alex Warstadt and Samuel R. Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In Shalom Lappin and Jean-Philippe Bernardy, editors, *Algebraic Structures in Natural Language*. Taylor and Francis.

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

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Ethan Wilcox, Richard Futrell, and Roger Levy. 2022. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–88.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Taha Yasseri, András Kornai, and János Kertész. 2012. A practical approach to language complexity: A Wikipedia case study. *PLoS ONE*, 7(11):e48386.

A CHILDES preprocessing details

The train, test, and validation split kept each document in the corpora intact to allow for learning of context. Since a document roughly correspond to a single recording session, and the sentence order within each document was not randomized, the networks could utilize cross sentence context while predicting the next word.

Generally, we kept the data as close to the actual input that the child receives as possible. However, in some cases we modified tokenization to match the CHILDES Treebank, a syntactically parsed subset of the CHILDES corpora. For instance, contractions were split, e.g. we replaced *don't* with *do n't*,

The ages of the children vary by corpus, ranging from six months to twelve years. Almost 95% (49/52) of the corpora consist of transcriptions with children between one and six years of age.

Note that for Experiment 2, we used the same vocabulary as we used in Experiment 1, which means that the words that were not present in Experiment 1's vocabulary were replaced with <unk> tokens.

The unprocessed CHILDES datasets were downloaded in XML format from the online XML version[10] of the CHILDES database (MacWhinney, 2000).[11] A modified NLTK CHILDESCorpusReader[12] was used to parse the XML into plain text for training.

The CHILDES dataset is licensed for use under a CC BY-NC-SA 3.0 license[13]. Under the terms of this license, the data can be freely used and adapted, as long as it is not used for commercial purposes and as long as attribution is provided.[14] Our usage fits these criteria.

Though CHILDES contains many corpora of many languages, we use only corpora from the North American English subset of CHILDES, which contains child-directed speech with many different North American children. See the CHILDES database for more details.

By the CHILDES rules for data citation,[15] research that relies on more than 6 of the corpora need only cite the overall database, not each individual corpus.

All the data on CHILDES must adhere to IRB guidelines,[16] including a requirement for anonymity.

The final dataset is included in our GitHub repository. This dataset is not intended for commercial use.

CHILDES corpora included The CHILDES corpora that we used were: Bates, Bernstein, Bliss, Bloom70, Bloom73, Bohannon, Braunwald, Brent, Brown, Carterette, Clark, Cornell, Demetras1, Demetras2, EllisWeismer, Evans, Feldman, Garvey, Gathercole, Gelman, Gillam, Gleason, HSLLD, Haggerty, Hall, Higginson, Kuczaj, MacWhinney, McCune, McMillan, Morisset, NH, Nelson, NewEngland, NewmanRatner, Normal, POLER, Peters, Post, Rollins, Sachs, Sawyer, Snow, Soderstrom, Sprott, Suppes, Tardif, Valian, VanHouten, VanKleeck, Warren, Weist.

B Hyperparameter Search and Model Implementation

We conducted a hyperparameter search for each of the architectures we investigated (LSTMs and Transformers). Our broad goal in this paper is to investigate the extent to which capturing the statistical properties of the CHILDES dataset naturally leads a learner to capture the structure of yes/no questions. Therefore, we sought to find the hyperparameter settings that made models most effective at capturing the statistical properties of CHILDES data, a goal which we operationalized as finding the model with the lowest perplexity.

B.1 Hyperparameter search

LSTMs For LSTMs we explored the following hyper-parameters via a grid search for a total of 144 models.

1. layers: 2
2. hidden and embedding size: 200, 800
3. batch size: 20, 80
4. dropout rate: 0.0, 0.2, 0.4, 0.6
5. learning rate: 5.0, 10.0, 20.0

[10] https://childes.talkbank.org/data-xml/
[11] https://childes.talkbank.org
[12] https://www.nltk.org/howto/childes.html
[13] https://talkbank.org/share/rules.html
[14] https://creativecommons.org/licenses/by-nc-sa/3.0/
[15] https://talkbank.org/share/citation.html
[16] https://talkbank.org/share/irb/

6. random seed: 3 per parameter combination, unique for each LSTM

The LSTM model with the lowest perplexity on the validation set after training had 2 layers, a hidden and embedding size of 800, a batch size of 20, a dropout rate of 0.4, and a learning rate of 10.[17] A LSTM model with these hyperparameters has 37,620,294 parameters.

Transformers For the Transformers we performed a hyperparameter sweep over the following hyper-parameters for a total of 84 models.

1. layers: 2, 4, 8, 16
2. context size: 50, 100, 500
3. hidden and embedding size: 200, 800, 1600
4. heads: 2, 4, 8, 16
5. batch size: 20, 80, 160
6. dropout rate: 0.0, 0.2, 0.4, 0.6
7. learning rate: 0.5, 1.0, 5.0, 10.0, 20.0
8. random seed: 3 per parameter combination

The Transformer model with the lowest perplexities after training had 4 layers, a context size of 500, a hidden size of 800, a batch size of 10, 4 heads, a dropout rate of 0.2, and a learning rate of 5.0. A Transformer model with these parameters has 42,759,494 parameters.

We did not include a warmup period in our training procedure. In informal experiments, we tried including a warmup period for both LSTMs and Transformers, but we found that this did not meaningfully affect the perplexity of the trained models in our setting.

B.2 Comment on model size

Although neural networks generally perform better as they increase in size, the best-performing models that we found were not the largest ones. This result is consistent with the finding of Warstadt et al. (2020b) that, for small training sets, smaller language models sometimes outperform larger ones. Thus, it is unlikely that scaling up models beyond the range we investigated would have yielded better CHILDES language models than the ones we trained.

[17]The hyperparameters we explored for the LSTMs were those of Gulordava et al. (2018), the code for which can be found at https://github.com/facebookresearch/colorlessgreenRNNs

B.3 Implementation

All models were implemented in PyTorch by building on code from https://github.com/facebookresearch/colorlessgreenRNNs and https://github.com/pytorch/examples/tree/main/word_language_model, and trained using Nvidia k80 GPUs. The final models are included in our GitHub repository. These models are not intended for commercial use.

C PREPOSE-ONE&DELETE-ONE Full Results

See Table 1 and Table 2 for these results.

LSTMs	Prepose First	Prepose Main
Delete First	0.01	0.14
Delete Main	0.39	0.12
Delete None	0.20	0.14

Table 1: Numerical results for LSTMs' preference for questions consistent with combinations of 'prepose' and 'delete' rules. Within each architecture, the proportion preferences across all 6 question types necessarily sum to 1.

Transformers	Prepose First	Prepose Main
Delete First	0.01	0.16
Delete Main	0.31	0.06
Delete None	0.25	0.21

Table 2: Numerical results for Transformers' preference for questions consistent with combinations of 'prepose' and 'delete' rules. Within each architecture, the proportion preferences across all 6 question types necessarily sum to 1.

C.1 Results using SLOR

See Table 3 and Table 4 for these results.

LSTMs	Prepose First	Prepose Main
Delete First	0.01	0.14
Delete Main	0.33	0.08
Delete None	0.26	0.18

Table 3: Analysis of LSTMs' preference for questions consistent with combinations of 'prepose' and 'delete' rules, evaluated using SLOR. Within each architecture, the proportion preferences across all 6 question types necessarily sum to 1.

Transformers	Prepose First	Prepose Main
Delete First	0.01	0.15
Delete Main	0.27	0.04
Delete None	0.29	0.24

Table 4: Analysis of Transformers' preference for questions consistent with combinations of 'prepose' and 'delete' rules, evaluated using SLOR. Within each architecture, the proportion preferences across all 6 question types necessarily sum to 1.

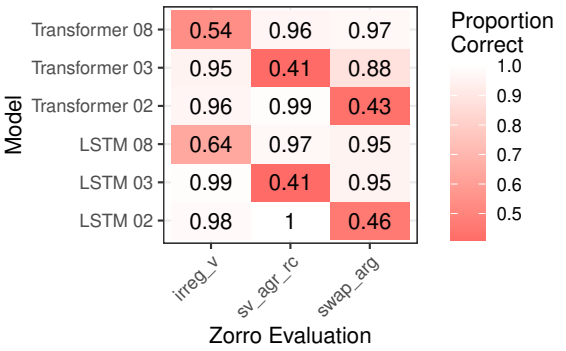

Figure 4: The performance of a selected subset of model re-runs on a selected subset of the Zorro evaluations. Each Zorro evaluation targets a specific syntactic phenomenon—in the cases shown here, irregular verbs, subject-verb agreement across relative clauses, and correct argument ordering.

D BabyBERTa dataset evaluation

For an illustrative subset of the results on the Zorro evaluation dataset (discussed in Section 4.5), see Figure 4. For the full results, see Figure 5.

E Move-One Dataset Results

One approach used in several past papers (e.g., Lewis and Elman (2001) and Reali and Christiansen (2005)) is to evaluate models using pairs of sentences that can be formed by starting with a declarative sentence (e.g., (17)) and moving one of its auxiliaries to the front of the sentence. The first sentence in each pair (e.g., (18a)) follows HIERARCHICALQ, because the *main* auxiliary is moved, while the second (e.g., (18b)), follows LINEARQ because the *first* auxiliary is moved.

(17) The children who **are** talking **are** sleeping.

(18) a. **Are** the children who **are** talking sleeping?
b. **Are** the children who talking **are** sleeping?

If a model assigns a higher probability to (18a) than (18b), that is evidence that the models favors HIERARCHICALQ over LINEARQ. While this preference is a necessary component of correctly learning HIERARCHICALQ, it is by no means sufficient: indeed, Kam et al. (2008) showed that models can prefer sentences consistent with HIERARCHICALQ over sentences consistent with LINEARQ due to shallow n-gram statistics rather than due to knowledge of hierarchical structure. More generally, there are infinitely many other incorrect hypotheses besides LINEARQ, and demonstrating successful learning of HIERARCHICALQ would require ruling out all of them. Investigating all possibilities is intractable, but we can at least investigate a few additional plausible ones. Thus, in the main paper we depart from prior work by considering a greater number of candidate sentences than just the pairs of sentences used in prior work.

To create the MOVE-ONE dataset, we randomly sampled 10,000 declarative sentences from our CFGs for which the first and main auxiliary were identical and then modified them to give 10,000 sentence pairs. To create the PREPOSE-ONE&DELETE-ONE dataset, we randomly sampled a different 10,000 declarative sentences from our CFGs for which the first and main auxiliary were different and then we modified them to give 10,000 6-tuples of sentences. See Appendix F for more details about the CFGs.

F Context Free Grammars

Figure 6 contains the context-free grammar used for the analyses in Section 4.6. Figures 7 and 8 contain the context-free grammars used for the targeted evaluation sets in Section 5.2; for each of these evaluation sets, we sampled 10,000 declarative sentences from these grammars and transformed them into questions according to HIERARCHICALQ. Figure 9 contains the vocabulary used for all of these datasets.

G Breakdown by lexical identity

Here we further break down models' predictions for the FIRST-AUX $\neq$ MAIN-AUX evaluation set based on the identities of the two auxiliaries in the input sentence. Figure 10 gives the results for the LSTM in the QUESTION FORMATION condition; Figure 11 for the LSTM in the NEXT-WORD PREDICTION + QUESTION FORMATION condition; Figure 12 for the Transformer in the QUESTION FORMATION condition; and Figure 13 for the for the Transformer in the NEXT-WORD PREDICTION + QUESTION FORMATION condition.

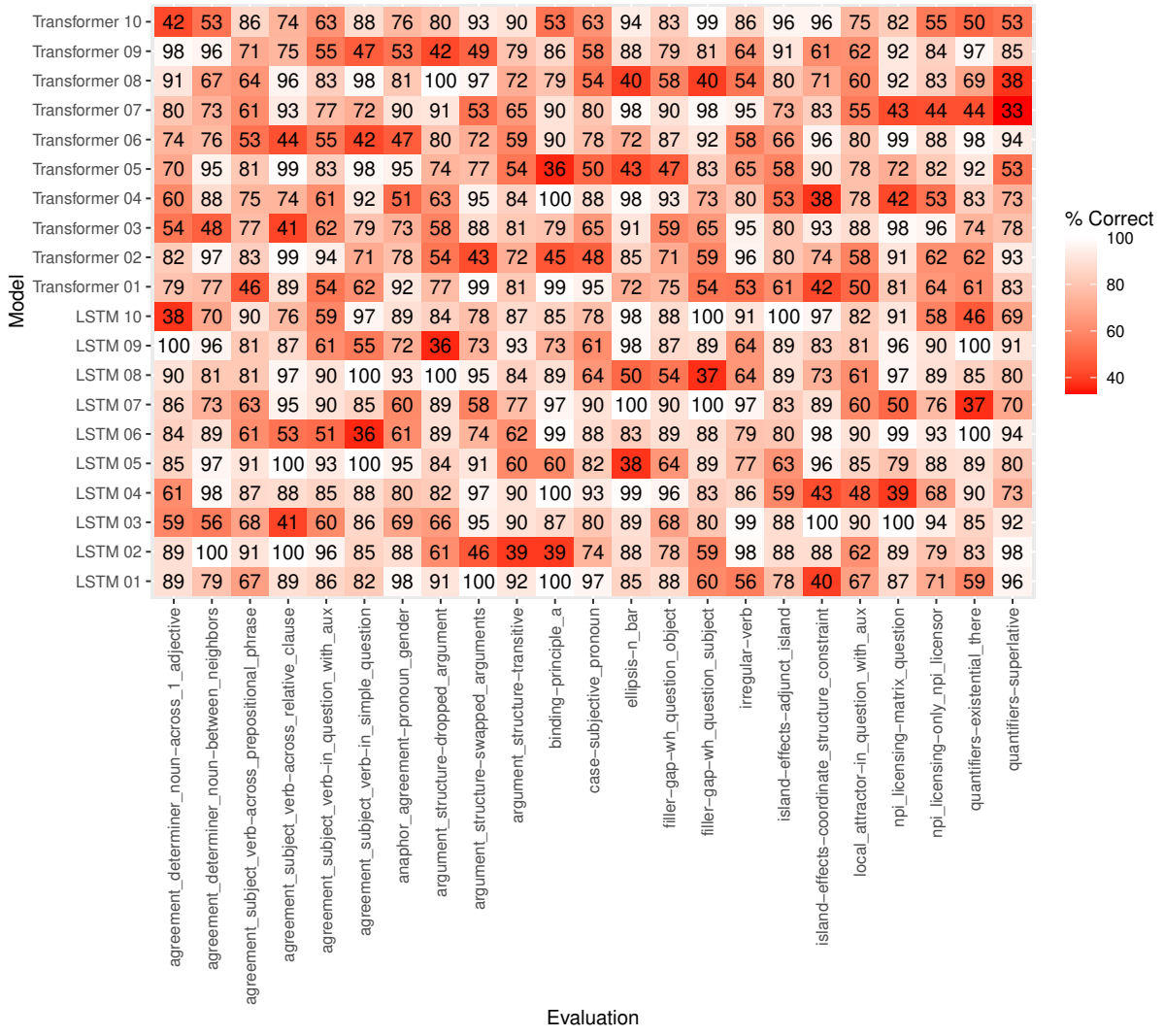

Figure 5: Results on the targeted syntactic evaluations in Huebner et al. (2021) in percent accuracy. Evaluation names in Figure 4 were shortened.

H Example generated text

Figure 14 gives some example text generated by our models. Models trained on next-word prediction produce their predictions as a probability distribution over the vocabulary. To use such models to generate text, we sample a word from this distribution then use that word as the model's input for the next time step.

S	→ {NP_S RC_S_BARE MAIN-AUX VP_S_PAST}
S	→ {NP_S RC_S_PAST MAIN-AUX VP_S_BARE}
S	→ {NP_S RC_S_BARE MAIN-AUX VP_S_PROG}
S	→ {NP_S RC_S_PROG MAIN-AUX VP_S_BARE}
S	→ {NP_S RC_S_PAST MAIN-AUX VP_S_PROG}
S	→ {NP_S RC_S_PROG MAIN-AUX VP_S_PAST}
S	→ {NP_P RC_P_BARE MAIN-AUX VP_P_PAST}
S	→ {NP_P RC_P_PAST MAIN-AUX VP_P_BARE}
S	→ {NP_P RC_P_BARE MAIN-AUX VP_P_PROG}
S	→ {NP_P RC_P_PROG MAIN-AUX VP_P_BARE}
S	→ {NP_P RC_P_PAST MAIN-AUX VP_P_PROG}
S	→ {NP_P RC_P_PROG MAIN-AUX VP_P_PAST}
NP_S	→ {Det_S N_S}
NP_P	→ {Det_P N_P}
NP_O	→ {Det_S N_S | Det_P N_P | Det_S N_S Prep Det_S N_S | Det_S N_S Prep Det_P N_P | Det_P N_P Prep Det_S N_S | Det_P N_P Prep Det_P N_P}
VP_S_BARE	→ {Aux_S IV }
VP_S_BARE	→ {Aux_S TV NP_O}
VP_S_PROG	→ {Aux_S_BE IV_IS}
VP_S_PROG	→ {Aux_S_BE TV_IS NP_O}
VP_S_PAST	→ {Aux_S_HAS IV_HAS}
VP_S_PAST	→ {Aux_S_HAS TV_HAS NP_O}
VP_P_BARE	→ {Aux_P IV}
VP_P_BARE	→ {Aux_P TV NP_O}
VP_P_PROG	→ {Aux_P_BE IV_IS}
VP_P_PROG	→ {Aux_P_BE TV_IS NP_O}
VP_P_PAST	→ {Aux_P_HAS IV_HAS}
VP_P_PAST	→ {Aux_P_HAS TV_HAS NP_O}
RC_S_BARE	→ {Rel Aux_S IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_S TV Det_S N_S | Rel Aux_S TV Det_P N_P}
RC_S_PROG	→ {Rel Aux_S_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_S_BE TV_IS Det_S N_S | Rel Aux_S_BE TV_IS Det_P N_P}
RC_S_PAST	→ {Rel Aux_S_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_S_HAS TV_HAS Det_S N_S | Rel Aux_S_HAS TV_HAS Det_P N_P}
RC_P_BARE	→ {Rel Aux_P IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_P TV Det_S N_S | Rel Aux_P TV Det_P N_P}
RC_P_PROG	→ {Rel Aux_P_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_P_BE TV_IS Det_S N_S | Rel Aux_P_BE TV_IS Det_P N_P}
RC_P_PAST	→ {Rel Aux_P_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_P_HAS TV_HAS Det_S N_S | Rel Aux_P_HAS TV_HAS Det_P N_P}

Figure 6: CFG used to generate PREPOSE-ONE-AND-DELETE-ONE evaluation dataset

S → {NP_M_S VP_M_S | NP_M_P VP_M_P}
NP_M_S → {Det_S N_S | Det_S N_S Prep Det_S N_S | Det_S N_S Prep Det_P N_P}
NP_M_P → {Det_P N_P | Det_P N_P Prep Det_S N_S | Det_P N_P Prep Det_P N_P}
NP_O → {Det_S N_S | Det_P N_P | Det_S N_S Prep Det_S N_S | Det_S N_S Prep Det_P N_P | Det_P N_P Prep Det_S N_S | Det_P N_P Prep Det_P N_P | Det_S N_S RC_S | Det_P N_P RC_P }
VP_M_S → {Aux_S IV }
VP_M_S → {Aux_S TV NP_O}
VP_M_S → {Aux_S_BE IV_IS}
VP_M_S → {Aux_S_BE TV_IS NP_O}
VP_M_S → {Aux_S_HAS IV_HAS}
VP_M_S → {Aux_S_HAS TV_HAS NP_O}
VP_M_P → {Aux_P IV}
VP_M_P → {Aux_P TV NP_O}
VP_M_P → {Aux_P_BE IV_IS}
VP_M_P → {Aux_P_BE TV_IS NP_O}
VP_M_P → {Aux_P_HAS IV_HAS}
VP_M_P → {Aux_P_HAS TV_HAS NP_O}
RC_S → {Rel Aux_S IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_S TV Det_S N_S | Rel Aux_S TV Det_P N_P}
RC_S → {Rel Aux_S_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_S_BE TV_IS Det_S N_S | Rel Aux_S_BE TV_IS Det_P N_P}
RC_S → {Rel Aux_S_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_S_HAS TV_HAS Det_S N_S | Rel Aux_S_HAS TV_HAS Det_P N_P}
RC_P → {Rel Aux_P IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_P TV Det_S N_S | Rel Aux_P TV Det_P N_P}
RC_P → {Rel Aux_P_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_P_BE TV_IS Det_S N_S | Rel Aux_P_BE TV_IS Det_P N_P}
RC_P → {Rel Aux_P_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_P_HAS TV_HAS Det_S N_S | Rel Aux_P_HAS TV_HAS Det_P N_P}

Figure 7: CFG used to generate FIRST-AUX = MAIN-AUX evaluation dataset

S → {NP_M_S VP_M_S | NP_M_P VP_M_P}
NP_M_S → {Det_S N_S | Det_S N_S Prep Det_S N_S | Det_S N_S Prep Det_P N_P}
NP_M_P → {Det_P N_P | Det_P N_P Prep Det_S N_S | Det_P N_P Prep Det_P N_P}
NP_O → {Det_S N_S | Det_P N_P | Det_S N_S Prep Det_S N_S | Det_S N_S Prep Det_P N_P | Det_P N_P Prep Det_S N_S | Det_P N_P Prep Det_P N_P | Det_S N_S RC_S | Det_P N_P RC_P }
VP_M_S → {Aux_S IV }
VP_M_S → {Aux_S TV NP_O}
VP_M_S → {Aux_S_BE IV_IS}
VP_M_S → {Aux_S_BE TV_IS NP_O}
VP_M_S → {Aux_S_HAS IV_HAS}
VP_M_S → {Aux_S_HAS TV_HAS NP_O}
VP_M_P → {Aux_P IV}
VP_M_P → {Aux_P TV NP_O}
VP_M_P → {Aux_P_BE IV_IS}
VP_M_P → {Aux_P_BE TV_IS NP_O}
VP_M_P → {Aux_P_HAS IV_HAS}
VP_M_P → {Aux_P_HAS TV_HAS NP_O}
RC_S → {Rel Aux_S IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_S TV Det_S N_S | Rel Aux_S TV Det_P N_P}
RC_S → {Rel Aux_S_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_S_BE TV_IS Det_S N_S | Rel Aux_S_BE TV_IS Det_P N_P}
RC_S → {Rel Aux_S_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_S_HAS TV_HAS Det_S N_S | Rel Aux_S_HAS TV_HAS Det_P N_P}
RC_P → {Rel Aux_P IV | Rel Det_S N_S Aux_S TV | Rel Det_P N_P Aux_P TV | Rel Aux_P TV Det_S N_S | Rel Aux_P TV Det_P N_P}
RC_P → {Rel Aux_P_BE IV_IS | Rel Det_S N_S Aux_S_BE TV_IS | Rel Det_P N_P Aux_P_BE TV_IS | Rel Aux_P_BE TV_IS Det_S N_S | Rel Aux_P_BE TV_IS Det_P N_P}
RC_P → {Rel Aux_P_HAS IV_HAS | Rel Det_S N_S Aux_S_HAS TV_HAS | Rel Det_P N_P Aux_P_HAS TV_HAS | Rel Aux_P_HAS TV_HAS Det_S N_S | Rel Aux_P_HAS TV_HAS Det_P N_P}

Figure 8: CFG used to generate FIRST-AUX $\neq$ MAIN-AUX evaluation dataset

Det_S	→ {the | some | this }
Det_P	→ {the | some | those}
N_S	→ {baby | girl | boy | animal | child | person | horse }
N_P	→ {babies | girls | boys | animals | children | people | horses }
IV	→ {play | read | draw | sit | fall | talk | sleep | try | work | walk}
IV_IS	→ {playing | reading | drawing | sitting | falling | talking | sleeping | trying | working | walking}
IV_HAS	→ {played | read | drawn | sat | fallen | talked | slept | tried | worked | walked}
TV	→ {call | see | find | help | feed | know | pick | visit | watch | reach}
TV_IS	→ {calling | seeing | finding | helping | feeding | knowing | picking | visiting | watching | reaching}
TV_HAS	→ {called | seen | found | helped | fed | known | picked | visited | watched | reached}
Aux_P	→ {do | did | can | would | shall}
Aux_S	→ {does | did | can | would | shall}
Aux_S_BE	→ {is | was}
Aux_P_BE	→ {are | were}
Aux_S_HAS	→ {has}
Aux_P_HAS	→ {have}
Prep	→ {by | behind }
Rel	→ {who | that }

Figure 9: Vocabulary used for the PREPOSE-ONE-AND-DELETE-ONE, FIRST-AUX ≠ MAIN-AUX, and FIRST-AUX = MAIN-AUX evaluation datasets

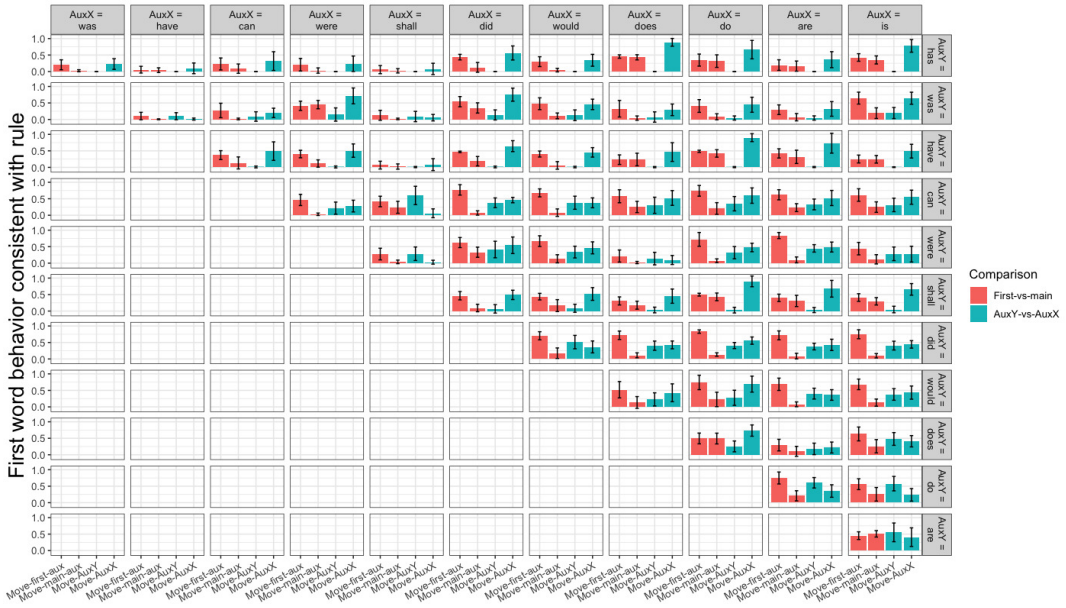

Figure 10: Breakdown by the identities of the two auxiliaries for outputs in the FIRST-AUX ≠ MAIN-AUX evaluation set for LSTMs trained only on question formation. The two leftmost bars in each cell show a First-vs-main comparison, while the two rightmost bars show an AuxY-vs-AuxX comparison.

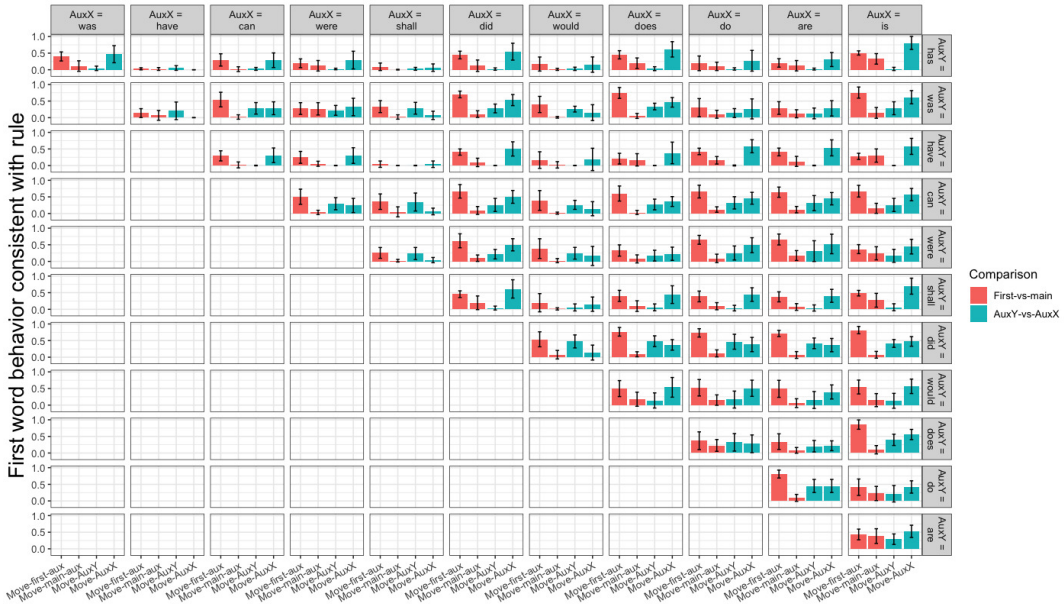

Figure 11: Breakdown by the identities of the two auxiliaries for outputs in the FIRST-AUX ≠ MAIN-AUX evaluation set for LSTMs first trained on next-word prediction and then question formation. The two leftmost bars in each cell show a First-vs-main comparison, while the two rightmost bars show an AuxY-vs-AuxX comparison.

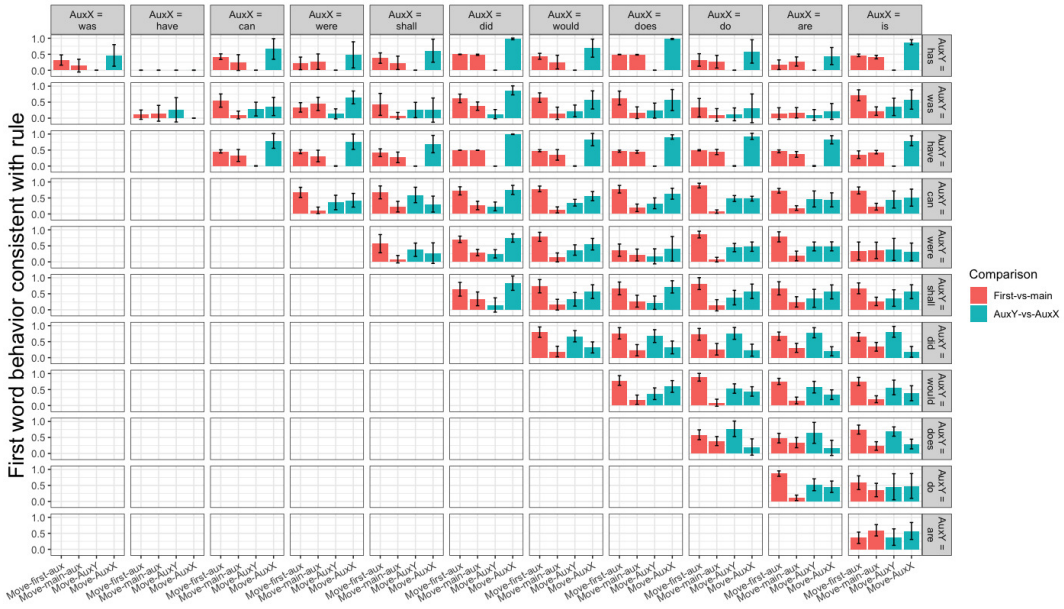

Figure 12: Breakdown by the identities of the two auxiliaries for outputs in the FIRST-AUX ≠ MAIN-AUX evaluation set for Transformers trained only on question formation. The two leftmost bars in each cell show a First-vs-main comparison, while the two rightmost bars show an AuxY-vs-AuxX comparison.

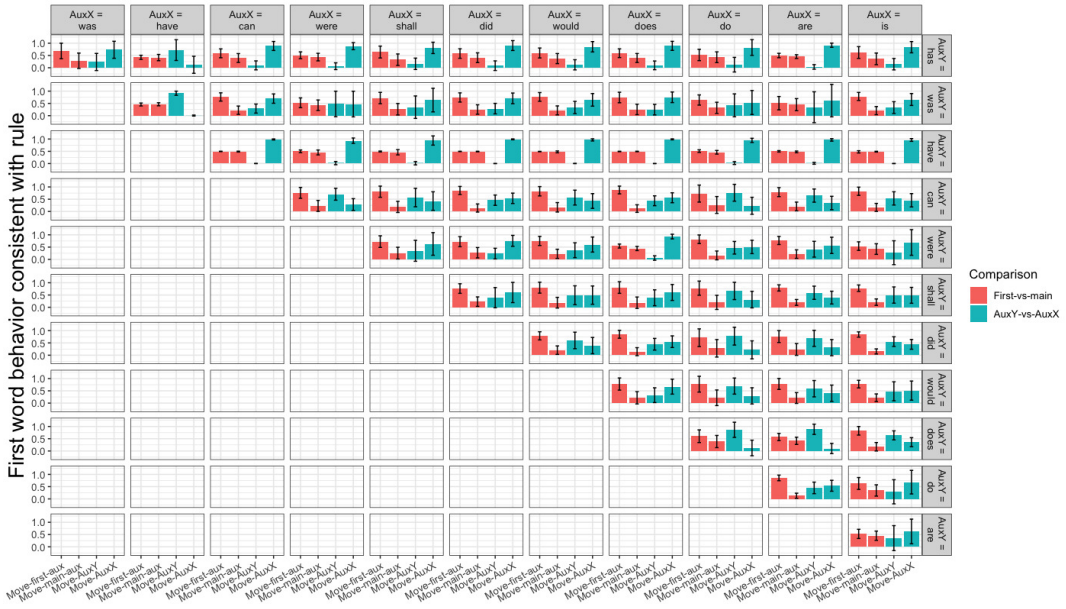

Figure 13: Breakdown by the identities of the two auxiliaries for outputs in the FIRST-AUX ≠ MAIN-AUX evaluation set for Transformers first trained on next-word prediction and then question formation. The two leftmost bars in each cell show a First-vs-main comparison, while the two rightmost bars show an AuxY-vs-AuxX comparison.

Text from CHILDES	do you eat in nursery school too ? oh . it 's nursery school . xxx . what do you get to eat ? what else do you get to eat besides a book ? do n't play with those books . what else do you have ? what do you have in nursery school ? alright - you can take it . why do n't you put the turtle back . well - you 're not going to change his water now . tonight . tonight - you change it about ten times a day . no - you can do it later . forget it now . too bad . too bad . that poor turtle . you do n't let him sit there five minutes . that was five minutes ?
LSTM generated text	do you eat in nursery school too ? oh . it 's nursery school . no ! tomorrow 's Friday . right . yeah - yeah . yeah I know what . I told you that he liked snipsnap that way . well this is address the grayish kings . ok . what kind of bird do you want to play with ? what else do you want to do ? your crayons . got a circle . I 'm going to use these . there . remember ? ok do you want to put all of these there ? ok let 's put everything away . what do you want to do ? you want to play in the box ? it 's just like a ball because it has a ball on it .
Transformer generated text	do you eat in nursery school too ? oh . it 's nursery school . mm-hm . and it looks like a little girl . mm-hm . like a little girl . hm , mm-hm . looks like she 's concerned about baseball this is a picture of a double football player . mm-hm . um , it has a uniform on it . mm-hm . anything more ? no . ok great . got a hat on . and a flower and a little flower . remember that ? do you remember that word ? what do you remember that word ? do you remember what it word says ? it says seven - eight . what 's that word ? it says eight . look at that word . three - four - six ... eight - nine ...

Figure 14: Comparison of text generated by the LSTM and Transformer models with a block of text chosen randomly from the training data. The LSTMs and Transformers were both prompted with the first three sentences of the text taken from CHILDES, which is the underlined in the two model generated texts. Note that neither of the model generated texts were cherry-picked either for quality or to be representative of the models' usual output: rather they were the first things they generated when prompted with the above underlined portion.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
End of Section 6.5 and the Limitations section

☑ A2. Did you discuss any potential risks of your work?
The Limitations section (after Section 7)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3, Section 4, Section 5.

☑ B1. Did you cite the creators of artifacts you used?
Section 1, Appendix A, Appendix B

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix A, Appendix B. In our GitHub repo, we release our data and code under the same license that CHILDES used.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Appendix A.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix A.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4, Section 5.

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We do report number of parameters of the models, and computing infrastructure in Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

and Appendix B, respectively. We do not report total computational budget. When the first author completed the training of the models they were unaware the GPU hours should be tracked. They now recognize how important this is. In the future they will make sure to track this information.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.5, Section 5.2, 6.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.4, Appendix A, Appendix B.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.