Structure Here, Bias There: Hierarchical Generalization by Jointly Learning Syntactic Transformations

Karl Mulligan[1] **Robert Frank**[2] **Tal Linzen**[3]
[1]Department of Cognitive Science, Johns Hopkins University
[2]Department of Linguistics, Yale University
[3]Department of Linguistics and Center for Data Science, New York University
karl.mulligan@jhu.edu bob.frank@yale.edu linzen@nyu.edu

Abstract

When learning syntactic transformations, children consistently induce structure-dependent generalizations, even though the primary linguistic data may be consistent with both linear and hierarchical rules. What is the source of this inductive bias? In this paper, we use computational models to investigate the hypothesis that evidence for the structure-sensitivity of one syntactic transformation can bias the acquisition of another transformation in favor of a hierarchical rule. We train sequence-to-sequence models based on artificial neural networks to learn multiple syntactic transformations at the same time in a fragment of English; we hold out cases that disambiguate linear and hierarchical rules for one of those transformations, and then test for hierarchical generalization to these held-out sentence types. Consistent with our hypothesis, we find that multitask learning induces a hierarchical bias for certain combinations of tasks, and that this bias is stronger for transformations that share computational building blocks. At the same time, the bias is in general insufficient to lead the learner to categorically acquire the hierarchical generalization for the target transformation.

1 Introduction

Children learning language are faced with a daunting task. Given a finite, limited set of utterances, children must infer the correct grammar for their language among an infinite number of compatible grammars. Despite the variability in quality and quantity in their linguistic input, children generally select grammars that are consistent with the generalizations that best explain the linguistic competence of their parents. How do children converge to such a grammar?

This question can be viewed as a generalization problem, in which sentences are learning instances, and competing grammars are possible generalizations over them. In the machine learning literature, any reason for picking one generalization consistent with the training data over another is referred to as **inductive bias** (Mitchell, 1980). A prominent hypothesis as to the source of such inductive bias is innate, domain-specific knowledge about grammar (Chomsky, 1965). This hypothesis rests on the argument that if there is insufficient evidence in the linguistic environment to acquire certain features of a given language, successful language acquisition must be attributed to a biological language faculty. The classic instantiations of this argument have to do with the structure-sensitivity of rules: linguistic generalization make crucial reference to the syntactic structure of the sentence, rather than the sequence of words it consists of. This view, often referred to as the poverty of the stimulus argument, has come under scrutiny, with critics indicating that positive evidence that would provide unambiguous evidence for one generalization over another, once assumed to be inaccessible, in fact appears more frequently in natural language than argued by Chomsky (Pullum and Scholz, 2002).

A central source of empirical evidence for the poverty of the stimulus argument comes from child language acquisition experiments. Crain and Nakayama (1987) tested 3 to 5-year-old children on their ability to form questions from declarative sentences, a transformation known as subject-auxiliary inversion. They asked children to form yes-no questions from sentences with a relative clause on the subject (e.g. *The cat that is yawning is on the couch*), a construction presumed by Crain and Nakayama to be unattested in question form in child speech. Participants in the experiment generally fronted the correct, matrix auxiliary in such sentences, and even when they made errors, those errors were never consistent with a non-hierarchical rule. These findings have been

Proceedings of the Society for Computation in Linguistics (SCiL) 2021, pages 125-135.
Held on-line February 14-19, 2021

interpreted by Crain and Nakayama as evidence for an innate structure-sensitivity bias.

The necessity of an innate structure-sensitivity bias has also been addressed by computational modeling work. Work in this area has compared the probabilities assigned by language models to correctly and incorrectly fronted questions using artificial languages (Lewis and Elman, 2001), child-directed speech corpora (Reali and Christiansen, 2005), and raw data used in unsupervised pretraining of BERT (Warstadt and Bowman, 2020), finding in all cases that models assigned higher probability to the hierarchically-generated, correctly fronted questions. Other work followed more closely the approach taken by Crain and Nakayama and investigated models trained to perform the transformation by producing an output sequence. Most relevant to this work, Frank and Mathis (2007) trained a neural network language model to output a question after observing a declarative sentence, often producing the correct auxiliary but failing to generalize on held-out cases. This paradigm has recently been extended to contemporary sequence-to-sequence networks (McCoy et al., 2020a), the architecture we use in this paper, with similar results: their models only generalized hierarchically when they were architecturally constrained to base its output on hierarchical syntactic structure.

Previous computational work has focused on inductive bias arising from the learning architecture, which would correspond to the innate biases often considered in linguistics. In this work, by contrast, we assess whether a structure-sensitivity bias for a particular transformation can arise from learning the target transformation jointly with other syntactic transformations. We hypothesize that, by observing hierarchical structure in other constructions and situations, learners may show a preference for hierarchical structure when tested on new constructions where the training data is ambiguous between generalizations.

To test this hypothesis, we use the **multitask learning** paradigm, which leverages training signals from related tasks to induce a bias for a target task. We train models to perform various syntactic transformations and find that some, but not all, combinations of tasks lead to improved hierarchical generalization, and that the magnitude of this effect depends on the nature of the transformations involved. Our findings suggest that some of the hierarchical bias exhibited in syntactic transformation tasks may emerge through indirect evidence made accessible by learning aspects of hierarchical structure in other parts of the linguistic input, but this evidence alone is insufficient for learning categorically structure-dependent rules. This indicates that, even given evidence that other transformations in the language are structure-sensitive, architectural structure-sensitivity bias is still required to obtain robust hierarchical generalization in a transformation for which such evidence is absent, at least in the simplified setting we investigate.

2 Background: Multitask Learning

Inductive bias can arise not only from model architecture but also from the training regimes in which it is employed. One such regime used for introducing bias is **multitask learning** (Caruana, 1998). In this setup, generalization performance on a **target task** is improved by leveraging training signals from related *side tasks*, which the model is trained to perform in parallel with the target task. In a neural network, for example, a weight update that improves the performance of the network on a side task might result in an inductive bias that is beneficial for the target task. One example Caruana gives is autonomous driving: to improve performance on the target task, determining the direction in which the car should be streered, the learner might be taught a series of side tasks, such as predicting the location of the center of the road, the intensity of the road surface, and so on. Caruana shows that when tested for generalization on unseen roadways, the multitask model outperforms the single-task model on the same quantity of within-task data.

Multitask learning has been shown to be effective in natural language tasks. For instance, the various similarities underlying all human languages have made multitask learning with multiple languages a useful approach to machine translation. Dong et al. (2015) designed encoder-decoder networks that learn to decode into many languages in parallel, where the shared encoder learns a more useful syntactic representation for translating into the target language, whereas Firat et al. (2016) used separate encoders and decoders for each language but shared the attention mechanism across languages. Johnson et al. (2017) performed multilingual translation using a single en-

coder and decoder by appending an artificial token in the input which indicates the language to be translated into. In this setting, they were able to perform zero-shot translation, i.e., translating between language pairs for which the model never received parallel data; instead, it relied on a shared representation learned via multitask learning.

The present work adopts the architecture used by Johnson et al. (2017), with a shared encoder and a shared decoder. Instead of translation *between* different languages, we deploy this architecture to learn different syntactic transformations *within* a single language (a fragment of English), and assess whether the process of learning the transformations can induce a bias for hierarchical generalization in other syntactic transformations.

3 Methods

We performed a series of multitask learning experiments.[1] In these experiments, one transformation task was always designated as the **target task**, and the others as **side tasks**. The training data for the target task were ambiguous as to whether they were generated using the hierarchical or linear rule; in other words, applying either of the rules to the inputs included in the training set resulted in identical outputs. By contrast, the training data for the side task provided unambiguous evidence for either a **hierarchical** or **linear** rule through the inclusion of key **disambiguating examples**, for which different rules produce different outputs.

3.1 Tasks

We used three syntactic transformations: passivization, question formation, and tense reinflection. In English, all of these transformations are governed by hierarchical structure. For each of these transformations, we also constructed a rule stated in terms of *linear* order of the words in the sentence, such that the rule produces identical outputs for the sentence types included in the training set. The tasks and their corresponding hierarchical and linear rules are described below. Table 1 gives examples of sentences on which the rules yield distinct outputs.

Passivization In this task, an active sentence is transformed into a passive sentence using MOVE-OBJECT or MOVE-SECOND. Disambiguating inputs are those with a relative clause (RC) or prepositional phrase (PP) on the subject, i.e., sentences in which the subject contains two different nouns.

> MOVE-OBJECT (hierarchical): Delete the subject noun phrase, move the **object noun phrase** to the subject position, and inflect the matrix verb to agree with the former object.
>
> MOVE-SECOND (linear): Delete the linearly first noun phrase, move the **linearly second noun phrase** to the front of the sentence, and inflect the linearly first verb to agree with the formerly second noun phrase.

Question Formation In this task, a declarative sentence is transformed into an interrogative sentence using MOVE-MAIN or MOVE-FIRST. Disambiguating inputs are those with an RC on the subject, i.e., sentences with an auxiliary that linearly precedes the matrix auxiliary.

> MOVE-MAIN (hierarchical): Move the **auxiliary modifying the main (matrix) verb** to the front of the sentence.
>
> MOVE-FIRST (linear): Move the **linearly first auxiliary** to the front of the sentence.

Tense Reinflection In this task, a past tense sentence is transformed into a present tense sentence using INFLECT-SUBJECT or INFLECT-RECENT. In English, past tense verbs have no number morphology (*did*), while present tense verbs inflect for the third-person singular (*does* vs. *do*). Disambiguating inputs are those with an RC or PP on the subject that contains a noun of a different number from the head of the subject; this noun therefore intervenes between the head of the subject and the matrix auxiliary.

> INFLECT-SUBJECT (hierarchical): Change the inflection of the main verb to match the number of the head of the **subject noun phrase**.
>
> INFLECT-RECENT (linear): Change the inflection of the main verb to match the number of the **most recently processed noun**.

Unlike question formation and passivization, tense reinflection as it is framed here is not a linguistically natural task: declarative and active sentences can easily be argued to be base forms, while

[1] Code and data for the experiments described in this paper are available at the following link: https://github.com/karlmulligan/mtl-transformations.

Input	Output (Hierarchical)	Output (Linear)
our handsome yak upon our fantastic grotesque newt does amuse her walrus . PASSIVE	**her walrus** is amused .	**our fantastic grotesque newt** is amused .
the yaks who don't amuse her agreeable quail do applaud my determined handsome vulture . QUEST	**do** the yaks who don't amuse her agreeable quail applaud my determined handsome vulture ?	**don't** the yaks who amuse her agreeable quail do applaud my determined handsome vulture ?
some newts upon my courageous zebra did admire my bewildered exuberant zebras . PRESENT	some newts upon my courageous zebra **do** admire my bewildered exuberant zebras .	some newts upon my courageous zebra **does** admire my bewildered exuberant zebras .

Table 1: Examples of disambiguating constructions for passivization, question formation, and tense reinflection, for which hierarchical and linear rules give distinct outputs.

no framework to our knowledge would treat the past tense as a base form and the present as a transformation over it; instead, a more plausible input form would be a lemma underspecified for tense. Instead of using an abstract lemma form, we use already-inflected forms for consistency across sequence-to-sequence tasks, with a view towards using these datasets to test any model that accepts English sentences as inputs. We emphasize that tense reinflection as it is defined here achieves our main goal of targeting the same fundamental features as the other syntactic tasks, namely, long-distance agreement and hierarchical structure.

3.1.1 Notation

As a shorthand for describing the combinations of tasks used in each experiment, we write the target task followed by the side task and a superscript that indicates whether the rule used to generate the disambiguating examples for the side task was hierarchical or linear. For instance, we write

$$\text{PASSIVE} \mid \text{QUESTION}^{\text{H}}$$

to indicate that passivization was the target task and question formation, with examples disambiguating the transformation in favor of the hierarchical rule (MOVE-MAIN), was the side task. We use the following abbreviations: PASSIVE = passivization; QUESTION = question formation; TENSE = tense reinflection; H = hierarchical; and L = linear.

3.1.2 Computational Building Blocks

The multitask approach for inducing a bias relies on the interrelatedness of the target task and the side task. In the case of syntactic transformations, we hypothesize that two tasks are related insofar as they have in common certain **computational building blocks**. For instance, both question formation and passivization involve MOVEMENT, more specifically fronting to the beginning of the sentence.[2] Table 2 shows the computational building blocks shared among tasks. While some of this terminology originates with the theory of Transformational Grammar (Chomsky, 1965), the level of description at which these subprocesses are compared is abstract and independent of any particular syntactic theory or framework.

Task	AGR	MOVE	DEL
PASSIVE	✓	✓	✓
QUESTION		✓	
TENSE	✓		

Table 2: A characterization of syntactic transformations in terms of coarsely defined computational building blocks: AGR = NUMBER AGREEMENT, MOVE = MOVEMENT, and DEL = DELETION.

3.2 Data

Each of the training datasets was generated using a probabilistic context-free grammar (PCFG) described below. The grammar was based on the one created by McCoy et al. (2018), with multiple modifications; we describe the grammar below. Each training example was created by sampling a sentence from the grammar and then applying a transformation to that sentence. While the

[2] Although the exact types of movement (word-level V-MOVEMENT and phrase-level NP-MOVEMENT) are different and involve different grammatical categories, they are fundamentally functionally related.

sentences generated from the PCFG were grammatical in English, the output of the transformation was ungrammatical in the conditions where the side task was disambiguated in favor of a linear rule (see the rightmost column of Table 1). We omitted from the training sets examples of the target task with constructions that constituted disambiguating evidence for one rule over another (e.g., questions that are compatible with having been generated using MOVE-MAIN but not MOVE-FIRST for question formation). As baselines, we also experimented with single-task training sets, which only included examples from the target task. Finally, since our task inventory included three tasks, we also conducted experiments with two side tasks.

The training set for each experiment included 100,000 examples of each transformation, with the single-task baseline datasets consisting only of examples that are ambiguous with respect to the rule that generated them (linear or hierarchical). Each multitask training set ($N = t * 100{,}000$, where t is the number of tasks) was created by concatenating and shuffling ambiguous examples of the target task with examples of one or more side tasks containing disambiguating evidence. The development set, used to determine early stopping, consisted of $t*1{,}000$ examples sampled from the same distribution as the training set.

Grammar All sentences generated from the grammar followed a basic subject-verb-object structure. A third of the subject and object noun phrases were modified by prepositional phrases (PP), and a third were modified by relative clauses (RC); the remaining noun phrases were unmodified. Noun phrase modification was not recursive: it was limited to one PP or RC per noun phrase. Each noun was modified by up to three adjectives, with the aim of varying sentence length and discouraging the potential use of position-based linear heuristics. The grammar used explicit auxiliaries rather than inflected verbs (e.g. *does giggle* instead of *giggles*) in order to facilitate transfer of representations across tasks, since all examples of question formation had explicit auxiliary verbs in the input. Finally, it only included transitive verbs, since passivization requires an object noun phrase in each sentence.

Each example consisted of an input sentence paired with a output sentence. Half of the examples were instances of syntactic transformations, in which the input sentence was followed by a token describing which transformation task to perform, such as `PASSIVE` (active to passive) or `QUEST` (declarative to question) or `PRESENT` (past tense to present). The other half of the examples were `IDENT` (identity) tasks, which consisted of simply reproducing the input sequence. The `IDENT` examples included instances of held-out constructions (e.g. subject RCs in QUESTION), but did not reveal the outputs of the transformations for these instances. The identity examples were included to familiarize the encoder with sentences of that type, so that the the model would be able to produce viable representations for such inputs when asked to apply transformations to those sentences at test time.

Each model was then evaluated on both an in-distribution **test set** ($N = 10{,}000$), featuring the same sentence constructions as those encountered in training, and also an out-of-distribution **generalization set** ($N = 10{,}000$) consisting only of the disambiguating constructions withheld during training (see Table 1 for examples); critically, on these sentences, the hierarchical and non-hierarchical rule make different predictions. The test and generalizations sets for the target task only contained examples that were consistent with the relevant hierarchical rule (and were therefore grammatical in English), in all conditions.

3.3 Models

All experiments used Gated Recurrent Networks (GRU) (Cho et al., 2014) with attention (Bahdanau et al., 2015). The output was generated using greedy decoding, that is, the highest probability word was generated at each time step. All models had a hidden state size of 256 units and were optimized using stochastic gradient descent with a learning rate of 0.001. These parameters were based on the best combination of model architecture and hyperparameters for generalization on question formation as reported by McCoy et al. (2020a). For each dataset configuration, we trained 10 models with different sets of randomly generated initial weights.

3.4 Evaluation

We used two evaluation metrics: full-sentence accuracy and partial credit. For **full-sentence** accuracy, we marked a sentence as correct when the sequence generated by the model matched the

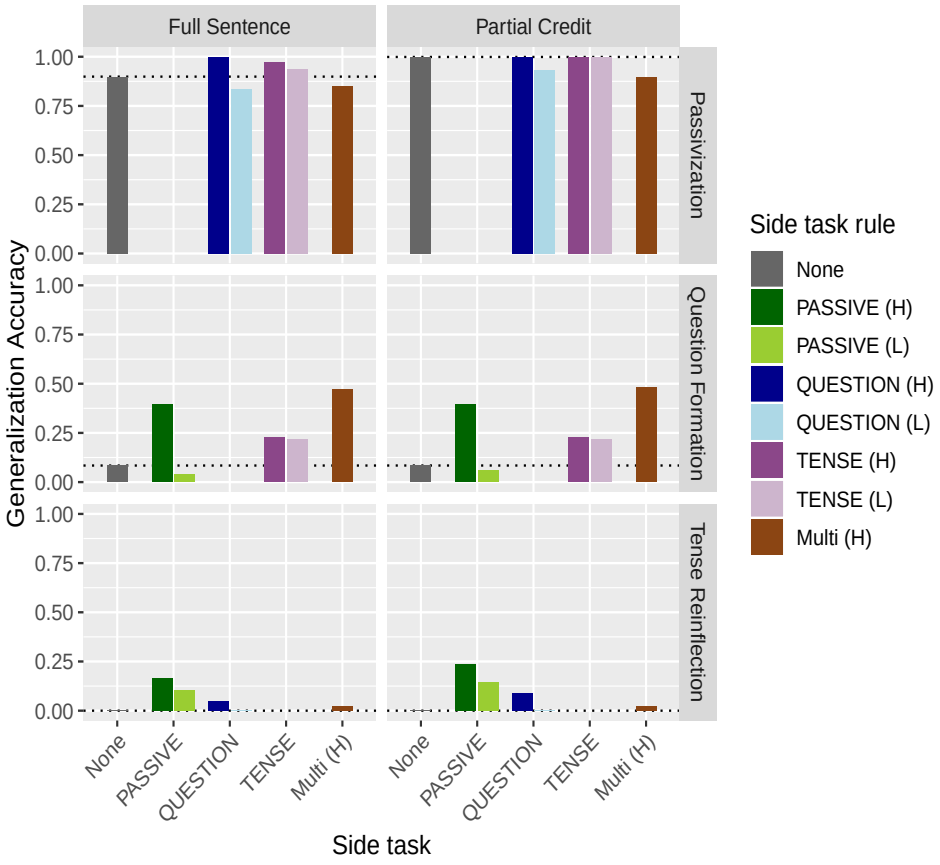

Figure 1: Mean generalization set accuracy. Models with mean accuracies above the baseline, single-task model generalization accuracy (dotted line) exhibits a multitask advantage. Models whose accuracies are higher for a hierarchical side task (darker) than their linear side task counterpart (lighter) exhibit a hierarchical advantage. Partial credit metrics are: for passivization, object-NP accuracy; for question formation, first-word accuracy; and for tense reinflection, matrix-auxiliary accuracy.

gold-standard sequence exactly. This is a stringent criterion, but it penalizes the model for errors that may have nothing to do with the syntactic generalization of interest. As such, we also assigned the models **partial credit** when the predicted sentence contained, in the correct position, the key word or words crucially indicative of sensitivity to hierarchical structure for that transformation. For passivization, we assigned partial credit when the first subsequence matched the object noun phrase (**object-NP accuracy**); for question formation, when the first word matched the correct auxiliary (**first-word accuracy**); and for tense reinflection, when the matrix auxiliary verb was inflected correctly (**matrix-auxiliary accuracy**). Since the crucial elements in each of these cases are of varying lengths, the partial credit measures are not directly comparable to one another: it is more challenging to match an entire noun phrase than a single word.

We use the following terminology to discuss the efficacy of multi-task learning. We say that a model shows a **multitask advantage** if, when it is trained on any side task at all, it generalizes according to the hierarchical rule more often than the single-task baseline (for example, if the generalization accuracy for PASSIVE | QUESTION$^{\text{H}}$ is higher than for PASSIVE alone). We further say that a pair of multitask models shows a **hierarchical advantage** if the model with a hierarchical side task performs better on the target task than a model with the linear version of that side task (e.g. generalization accuracy is greater for PASSIVE | QUESTION$^{\text{H}}$ than PASSIVE | QUESTION$^{\text{L}}$).

4 Results

In-distribution accuracy On test sentences drawn from the same distribution as the training

set, all models, single-task and multitask models alike, averaged 97% full-sentence accuracy across tasks and 99% partial credit. Accuracy on most tasks was at ceiling, but there was a multitask advantage for all target task PASSIVE models on full-sentence accuracy.

Generalization accuracy On the out-of-distribution generalization set, performance was generally high for PASSIVE, and much lower for QUESTION and TENSE. In contrast to prior work, the full-sentence and partial credit measures are very similar, thus indicating that, for example, in question formation, whenever the model failed to output the entire sentence correctly it also failed to produce the correct first word. Figure 1 shows the generalization accuracy for all task combinations.

Breakdown by task combination The task combinations that showed the most prominent effects on full-sentence accuracy were PASSIVE | QUESTION^{H} (+10% multitask advantage, +17% hierarchical advantage) and QUESTION | PASSIVE^{H} (+31% m.a., +35% h.a.). There are also smaller effects for TENSE | PASSIVE^{H} (+16% m.a., +6% h.a.) and TENSE | QUESTION^{H} (+4% m.a., +4% h.a.). Combinations with a multitask advantage, but no hierarchical advantage include QUESTION | TENSE (+14% m.a.). None of the models showed a hierarchical *disadvantage*: the hierarchical side tasks were always at least as effective as the linear ones in inducing a hierarchical bias. At the same time, the fact that even the linear side tasks tended to increased the likelihood of hierarchical generalization, compared to the single-task baseline, suggests that the larger set of syntactic constructions included in the multitask training sets, most of it ambiguous as to the rule that governs the transformations, contained considerable cues to hierarchy, regardless of how the critical examples were disambiguated.

5 Adding a Small Number of Disambiguating Examples

In the previous experiments, the evaluation on the generalization set was a test of **zero-shot generalization**: the models were asked to perform a transformation on a syntactic structure for which they have received no examples of the transformation of interest. The zero-shot assumption may be unrealistically strict if children are indeed exposed to a small number of disambiguating examples, as argued by Pullum and Scholz (2002). Even if a particular multitask learning configuration did not show an effect in the zero-shot setting, then, it may still impart to the models a bias for hierarchical generalization that would manifest in faster learning of the correct generalization; that is, it could learn it from fewer disambiguating examples than needed for a model without such a bias (McCoy et al., 2020b).

To empirically determine whether multitask learning can facilitate such **few-shot learning**, we added to the training sets described in Section 4 a few examples of the target transformations applied to the disambiguating construction. If the model normally requires n examples to generalize properly, we expect a hierarchical inductive bias to lead the model to require m examples in the multitask setup to reach the same generalization performance, where $m < n$.

5.1 Methods

For each target task, we removed 5, 10, 50, 100, 500, or 1000 examples from the generalization set, and inserted them into the training set (the disambiguating examples therefore constituted 0.00025%, 0.005%, 0.0025%, 0.05%, 0.025%, or 0.5% of the training data, respectively). The models were then trained and evaluated as before, again with 10 different weight initialization for each combination of multitask configuration and value of n.

5.2 Results

As before, full-sentence accuracy on the in-distribution test set was near perfect for all models. On the generalization sets, for both single-task and multitask models, accuracy for a small number of disambiguating examples was generally similar to $n = 0$, and increased to about 90% partial credit accuracy for most tasks at $n = 1000$.

Multitask advantages gradually disappeared as n increased (Figure 3 in the Appendix). Figure 2 shows a similar pattern for the hierarchical advantage: for greater n, the initial difference between the hierarchical side task and linear side task gets progressively smaller as more within-distribution disambiguating evidence is introduced. In other words, then, the more evidence for the hierarchical rule for the target transformation, the weaker the impact of the bias from other sources.

Perhaps surprisingly, even very small n were able to have a large effect on improving partial

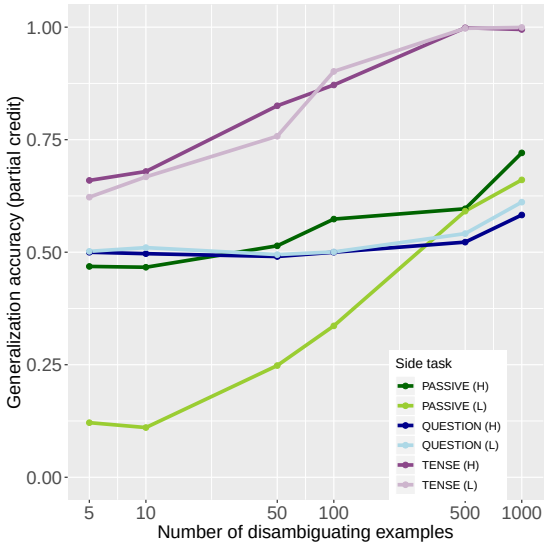

Figure 2: Mean generalization accuracy as a function of number of disambiguating examples of the target task. A log scale is used for the x-axis.

credit generalization accuracy for some tasks. For QUESTION, out of a dataset of 2 million examples, a mere $n = 5$ disambiguating examples were enough to produce a gain from 36% to 47% on partial credit accuracy. However, PASSIVE did not show major improvement over zero-shot models until $n = 50$ and beyond. Table 3 shows improvement over zero-shot models as a function of n.

	Number of disambiguating examples			
Task	0	10	100	1000
PASSIVE	0.76	-0.01	+0.09	+0.19
QUESTION	0.36	+0.19	+0.40	+0.55
TENSE	0.08	+0.02	+0.09	+0.66

Table 3: Improvement in main task generalization accuracy over zero-shot models for different settings of n (partial credit, averaged over all side tasks).

6 Discussion

Our experiments continue a tradition of behavioral and computational work that has sought to illuminate the issue of the poverty of the stimulus in the acquisition of syntax, and has done so by evaluating whether the generalizations that learners acquire for syntactic transformations follow hierarchical principles. In our experiments, we tested the hypothesis that jointly learning multiple types of syntactic transformations could induce a bias for hierarchical generalization. We found that multitask learning indeed resulted in increased hierarchical bias for some, but not all, combinations of syntactic tasks; this increase was often modest and did not lead to categorical hierarchical behavior.

When is one task useful in guiding the learning of another? We hypothesized that tasks that share computational building blocks like MOVEMENT would be more likely to show transfer in a multitask learning setting. In our experiments, we found that PASSIVE was the side task most successful at inducing a bias, whereas QUESTION and TENSE were less mutually informative. We also found that such relationships were often symmetric with respect to whether a task was the target task or side task. For instance, for the models in Figure 1, PASSIVE as a side task induced a multitask advantage for both QUESTION and TENSE, and both those tasks likewise induced a (far smaller) multitask advantage for PASSIVE.

The reason for the difference in *magnitude* of these induced advantages may also be explainable in terms of shared computational building blocks. Because passivization involves many building blocks (DELETION, MOVEMENT, and NUMBER-AGREEMENT), while the other transformations only involve a subset of these, PASSIVE may be more informative as a side task to QUESTION and TENSE than the other way around.

Although the computational building blocks theory can account for some of the variability, the preference for the hierarchical generalization was not categorical: models trained with the hierarchical side tasks did not exhibit behavior consistent with having learned absolute structural *rules*, but rather had a stronger *preference* for correct generalizations than their counterparts trained with a linear side task.

One explanation for the lack of a clearly categorical inductive leap is that different transformations may target different aspects of hierarchical structure; that is, hierarchy could be learned as a single concept, or piecemeal, depending on the aspect of hierarchical structure targeted by a certain pattern of data. Perfors et al. (2011) show that an ideal learner tends toward hierarchical behavior not because of an explicit structural bias, but because the hierarchical rules in their experiments could be expressed more concisely than linear ones; while this is also true of our experiments, the transferability of a learned hierarchical rule is

more complicated and not guaranteed.

Another explanation is that the architecture may impose a bottleneck in the decoder; while the single-task models only need to learn to perform a single transformation, the multitask models must learn to do several with the same number of parameters. While this step encourages parameter sharing, it may also be forcing an excessively compact representation. In future work, this hypothesis can be tested by implementing a separate decoder per task (see Section 2 for examples), and make use of other decoder techniques such as beam search.

6.1 Future Work

The current work serves as a proof of concept for inducing hierarchical bias through multitask learning. However, it remains an open question to what extent joint learning of this kind plays a role in shaping generalization behavior in human language acquisition. In our experiments, passivization was the side task that most reliably induced a hierarchical bias, but in our multitask training sets PASSIVE examples and the target task examples were distributed equally; in spoken language, passives are a far rarer phenomenon (Brown, 1973). In the future, this concern can be addressed, to some extent, by lowering the proportion of passives compared to the primary task data, such that it comprises, say, only 5% of the dataset rather than 50%, and seeing whether the bias persists.

Future work will also attempt to isolate precisely which task properties are responsible for transfer across tasks in a more systematic fashion. For instance, an alternative possibility for why PASSIVE may have been the most informative side task is because it had a higher proportion of disambiguating examples in the side task training sets than QUESTION and TENSE as a result of the PCFG and task properties. We can control for this and other factors by designing more controlled languages and tasks that distill computational building blocks like MOVEMENT down to even simpler computational primitives. By taking inspiration from the notion of a *minimal example*, the goal of future projects will be to define **minimal transformations** based on simple languages with limited vocabularies. By doing so, it will be possible to better understand how syntax learning in multitasking networks takes place.

7 Conclusion

We have shown that multitask learning of syntactic tasks in sequence-to-sequence neural networks can induce a gradient bias for hierarchical generalization in new tasks. This bias manifests more strongly for some task combinations than others; the strength of this effect appears to be related to the number of computational building blocks the tasks have in common. While no side tasks led to the induction of a categorical hierarchical rule for the target task, they were shown to give a significant bias toward correct generalization, though this bias was greatest when there were no or few disambiguating examples, and gradually disappeared as more disambiguating evidence became available.

In natural language, hierarchical structure is everywhere. Our evidence suggests that neural network models of language learning are capable of identifying hierarchical structure and using that information to perform structure-dependent transformations, without necessarily relying on an innate bias. Nonetheless, it remains to be seen whether this model of learning is applicable to human language acquisition, and more work is necessary to understand the nature of the syntactic knowledge contained in these jointly learned representations.

Acknowledgments

We thank Tom McCoy and other members of the JHU Computation and Psycholinguistics lab for their helpful comments. This material is based upon work supported by the National Science Foundation (NSF) grant nos. BCS-1920924 and BCS-1919321. This research project was conducted using computational resources at the Maryland Advanced Research Computing Center (MARCC).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Roger Brown. 1973. *A First Language: The Early Stages.* Harvard U. Press, Oxford, England.

Rich Caruana. 1998. Multitask Learning. In Sebastian Thrun and Lorien Pratt, editors, *Learning to Learn*, pages 95–133. Springer US, Boston, MA.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Number 11 in Massachusetts Institute of Technology (Cambridge, Mass.). Research Laboratory of Electronics. Special Technical Report. M.I.T. Press, Cambridge, Mass.

Stephen Crain and Mineharu Nakayama. 1987. Structure Dependence in Grammar Formation. *Language*, 63(3):522.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-Task Learning for Multiple Language Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

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

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

John D Lewis and Jeffrey L Elman. 2001. Learnability and the Statistical Structure of Language: Poverty of Stimulus Arguments Revisited. In *Proceedings of the 26th Annual Boston University Conference on Language Development*, volume 1, pages 359–370. Cascadilla Press.

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, Austin, TX.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020a. Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*.

R. Thomas McCoy, Erin Grant, Paul Smolensky, Thomas L. Griffiths, and Tal Linzen. 2020b. Universal linguistic inductive biases via meta-learning. In *Proceedings of the 42th Annual Conference of the Cognitive Science Society*.

Tom M Mitchell. 1980. The Need for Biases in Learning Generalizations. Technical Report CBM-TR-117, New Jersey: Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.

Amy Perfors, Joshua B. Tenenbaum, and Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338.

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

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.

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

A Appendices

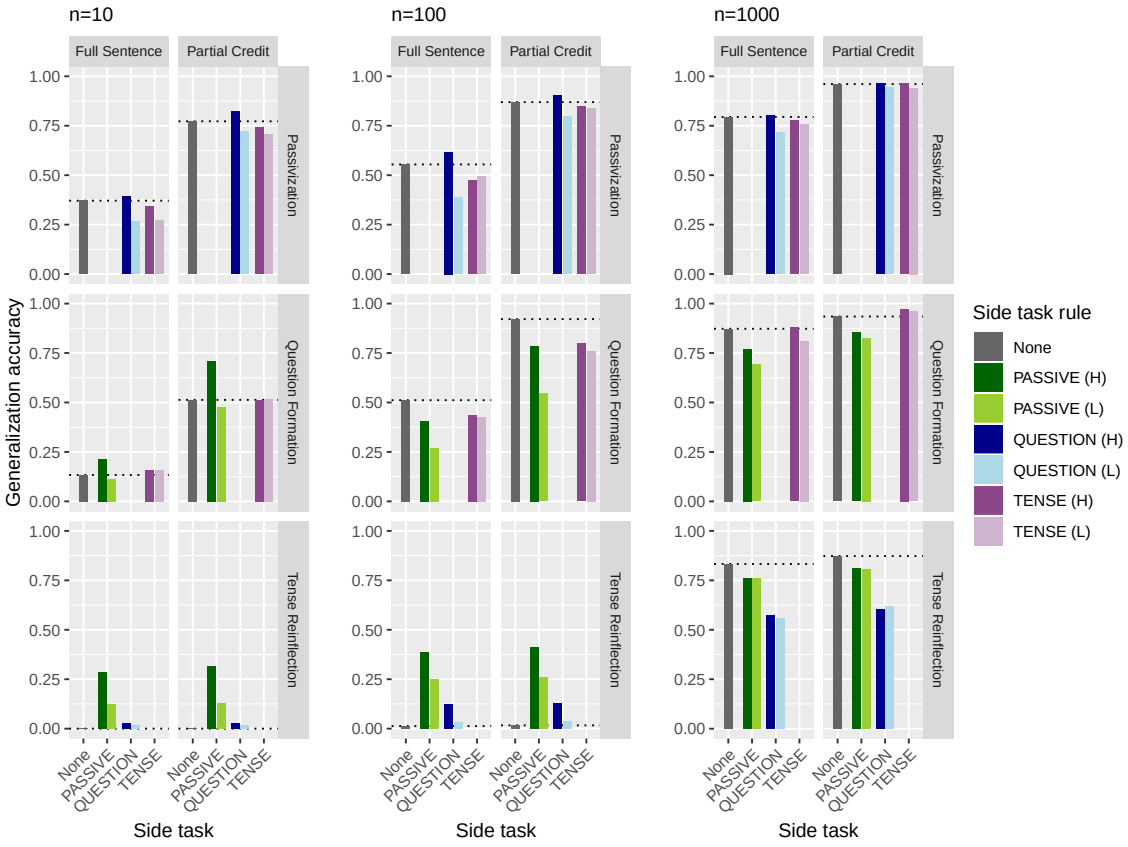

Figure 3: Mean generalization accuracy for few-shot models ($n = 10, n = 100, n = 1000$). For greater number of disambiguating examples n, the single-task models "raise the bar" at a more competitive rate, thereby weakening or eliminating instances of multitask advantage observed in the zero-shot setting.