Coloring the Blank Slate: Pre-training Imparts a Hierarchical Inductive Bias to Sequence-to-sequence Models

Anonymous ACL submission

Abstract

Relations between words are governed by hierarchical structure rather than linear ordering. Sequence-to-sequence (seq2seq) models, despite their success in downstream NLP applications, often fail to generalize in a hierarchy-sensitive manner when performing syntactic transformations—for example, transforming declarative sentences into questions—instead generalizing linearly using positional surface heuristics. However, syntactic evaluations of seq2seq models have only observed models that were *not* pre-trained on natural language data before being trained to perform syntactic transformations, in spite of the fact that pre-training has been found to induce hierarchical linguistic generalizations in language models; in other words, the syntactic capabilities of seq2seq models may have been greatly understated. Here, we make use of the pre-trained seq2seq model T5 (and its multilingual variant mT5) and evaluate whether they generalize hierarchically on two syntactic transformations in two languages: question formation and passivization in English and German. We find that T5 and mT5 generalize hierarchically when performing syntactic transformations, whereas non-pre-trained baseline models do not. This result presents additional evidence for the learnability of hierarchical syntactic information from non-annotated natural language text while also demonstrating that seq2seq models are capable of syntactic generalization.

1 Introduction

Human language is structured hierarchically. In NLP tasks like natural language inference, syntactic competence is a prerequisite for robust generalization (e.g., McCoy et al., 2019). Probing studies have found that masked language models (MLMs) contain hierarchical representations (Tenney et al., 2019; Hewitt and Manning, 2019; Clark et al., 2019), while behavioral studies of recurrent neural language models (Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018; van Schijndel et al., 2019) and MLMs (Goldberg, 2019; Hu et al., 2020) have found that models are largely able to capture long-range syntactic dependencies that require hierarchical representations of sentences.

Recent evidence suggests that MLMs like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) can learn to make hierarchical linguistic generalizations through exposure to text (Warstadt and Bowman, 2020), although the acquisition of many of these linguistic generalizations requires large amounts of data (Warstadt et al., 2020). However, this evidence comes from binary acceptability judgment tasks, where a classifier head is attached to an MLM and the model is tuned to classify which sentence in a given minimal pair is consistent with a hierarchical linguistic generalization, rather than a linear positional generalization. Consider the following two transformations of Example (1):

(1) The farmer that **has** seen the horse **hasn't** helped his friend.

a. **Hasn't** the farmer that **has** seen the horse helped his friend?

b. ***Has** the farmer that seen the horse **hasn't** helped his friend?

Example (1-a) correctly forms the question by moving the main auxiliary verb to the front of the sentence, while (1-b) relies on the incorrect positional heuristic that the first auxiliary in the declarative sentence is always inverted. When differentiating grammatical and ungrammatical auxiliary inversions, a model could rely on distributional information (Lewis and Elman, 2001) such as bigram heuristics (Reali and Christiansen, 2005; Kam et al., 2008) to make correct judgments in many cases, so high performance on binary classification tasks may overstate the syntactic competence of a model.

By contrast, *performing* a syntactic transformation—e.g., given a declarative sentence like

Example (1) as input, transforming it into a polar question like (1-a)—is more difficult, as it requires multiple complex but systematic operations (such as movement, case reinflection, and number agreement) that rely on hierarchical structure. Evaluations of syntactic transformational abilities can therefore act as more targeted behavioral indicators of syntactic structural representations in neural models. McCoy et al. (2018) evaluate non-pre-trained recurrent sequence-to-sequence (seq2seq; Sutskever et al., 2014) models on the question formation task, finding that they rely on linear/positional surface heuristics rather than hierarchical structure to perform this syntactic transformation. More recent studies have also exclusively observed non-pre-trained recurrent seq2seq models and non-pre-trained Transformer models (Petty and Frank, 2021) on other transformations like tense reinflection (McCoy et al., 2020) and passivization (Mulligan et al., 2021), finding similar results. These studies were designed to understand the inductive biases of various seq2seq architectures, hence why they do not pre-train the models on non-annotated natural language data before training them to perform syntactic transformations.

However, as Warstadt and Bowman (2020) find that non-annotated natural language text can induce preferences for hierarchical generalization in MLMs—and as positive results from syntactic evaluations have come from language models which *have* been trained on large amounts of data (Hu et al., 2020)—we hypothesize that a seq2seq model exposed to a large amount of language will also acquire preferences for hierarchical generalizations. That is, we expect pre-trained models to make use of structural rather than surface features when generalizing to held-out examples. In this study, we make use of the recent availability of a large pre-trained seq2seq model T5 (Raffel et al., 2020) and its multilingual variant mT5 (Xue et al., 2021) to investigate whether seq2seq models acquire preferences for hierarchical linguistic generalizations through pre-training. We test this by observing T5 and mT5 (henceforth, (m)T5)'s syntactic transformational abilities on English and German question formation and passivization tasks.

We find that (m)T5 generally performs syntactic transformations in a hierarchy-sensitive manner, while non-pre-trained models (including randomized-weight versions of (m)T5) rely primarily on linear/positional heuristics to perform the transformations. This finding presents additional evidence for the learnability of hierarchical syntactic information from natural language text input.

2 Syntactic Transformations

2.1 Languages

We evaluate on syntactic transformations in English and German. We choose English to allow for comparisons to previous results (McCoy et al., 2018; Mulligan et al., 2021). We further extend our evaluations to German because it exhibits explicit case marking on determiners and nouns; this typological feature has been found to increase the sensitivity of language models to syntactic structure (Ravfogel et al., 2019). This allows us to compare transformational abilities for languages with different levels of surface cues for hierarchy.

2.2 Tasks

We employ a *poverty of the stimulus* experimental design (Wilson, 2006), where we train the model on examples of a linguistic transformation that are compatible with either a hierarchical rule or a linear/positional rule, and then evaluate the model on sentences where only the hierarchical rule leads to the generalization pattern that is consistent with the grammar of the language.[1] In other words, we are interested in whether (m)T5 demonstrates a **hierarchical inductive bias**,[2] unlike the linear inductive bias displayed in prior work by non-pre-trained models (McCoy et al., 2020).

We focus on two syntactic transformation tasks: **question formation** and **passivization**. See Table 1 for a breakdown of which structures we present to the model during training and which we hold out to evaluate hierarchical generalization.

Question formation. In this task, a declarative sentence is transformed into a polar question by moving the main (matrix) auxiliary verb to the start of the sentence; this hierarchical rule is called MOVE-MAIN. The linear rule, MOVE-FIRST, entails moving the linearly first auxiliary verb to the front of the sentence. We train the model only on sentences with no relative clauses (RCs) or with RCs on the object—both cases in which the first

[1] Note that there are other rules that could properly transform the stimuli we use, but we find that the models we test do learn one of these rules or the other.

[2] When multiple generalizations are consistent with the training data, "inductive bias" refers to a model's choice of one generalization over others.

	☐ Train, dev, test	▨ Generalization
Structure	Question Formation	Passivization
No RC/PP	quest: some xylophones have remembered my yak. → have some xylophones remembered my yak?	passiv: your quails amused some vulture. → some vulture was amused by your quails.
RC/PP on object	quest: my zebras have amused some walrus who has waited. → have my zebras amused some walrus who has waited?	passiv: some tyrannosaurus entertained your quail behind your newt. → your quail behind your newt was entertained by some tyrannosaurus.
RC/PP on subject	quest: my vultures that our peacock hasn't applauded haven't read. → haven't my vultures that our peacock hasn't applauded read?	passiv: the zebra upon the yak confused your orangutans. → your orangutans were confused by the zebra upon the yak.

Table 1: The distribution of syntactic structures in the train, test, and generalization sets. Note: to expose the model to all structures during training and fine-tuning, we also include identity transformations for all structures using the "decl:" prefix, where the input and output sequences are the same declarative or active sentence (see §3.1). We use the test set to evaluate whether models have learned the task on in-distribution examples, and the generalization set to evaluate whether models generalize hierarchically. See Appendix B for example sentences in German.

auxiliary verb is always the matrix verb. We withhold examples in which RCs modify the subject, thus making the matrix auxiliary verb the linearly second auxiliary in the sentence, as such examples disambiguate between the two rules.

In English, we use the auxiliaries 'has', 'hasn't', 'have', and 'haven't', with past participle main verbs. We use affirmative and negative forms of the auxiliary to distinguish between the multiple auxiliaries in test sentences: exactly one of the auxiliaries in such sentences is negative and the other is positive (though we vary which is which). As a result, we can determine whether the induced mapping follows a hierarchical or linear inductive bias. In German, negation is realized as a separate word that is not fronted with the auxiliary. To make the multiple auxiliaries in a test sentence distinct, we therefore use the modal 'können' (*can*) along with the auxiliary 'haben' (*have*), together with past participle or infinitival main verbs as appropriate. As before, this allows us to distinguish models with a hierarchical bias from those with a linear bias on the basis of the fronted auxiliary.

Passivization. In this task, an active sentence is transformed into a passive sentence by moving the object noun phrase (NP) to the front of the sentence (MOVE-OBJECT). The training examples we use are also compatible with a linear rule, MOVE-SECOND, in which the linearly second NP moves to the front of the sentence. We train on sentences with either no prepositional phrases (PPs) or with PPs modifying the object—i.e., where the second NP is always the object. Disambiguating examples are those which place prepositional phrases (PPs) on the subject, thus making the object the third NP in the sentence.

Passivization additionally requires other movements, insertions, tense reinflection, and (for German) case reinflection. In Examples (2) and (3) below, in addition to the displacement of the object (in blue), 'be'/'werden' (in red) is inserted in a form appropriate to the grammatical features of the fronted NP; the original subject NP (in brown) is moved to a 'by'/'von' phrase at the end of the sentence; and the main verb (in orange) is reinflected to be a past participle or infinitive. In German, there are even more required operations: the case of the NPs (reflected largely in the determiners) must be reinflected and the main verb needs to be moved to the end of the sentence.

(2) *English Passivization*:
  a. Your quails amused some vulture.
  b. Some vulture was amused by your quails.

(3) *German Passivization*:
  a. Ihr Esel unterhielt meinen Salamander.
     Your.NOM donkey entertained my.ACC salamander.
  b. Mein Salamander wurde von ihrem Esel unterhalten.
     My.NOM salamander became from your.DAT donkey entertained.

We provide examples of both transformations in both languages in Table 2. When tuning (m)T5, we use task prefixes in the source sequence before the input. We use "quest:" for question formation and "passiv:" for passivization. As in previous work, we also include identity transformation examples (prefixed with "decl:"), i.e., examples for which the model has to output the unchanged declarative or active sentence. When training seq2seq baselines, we follow McCoy et al. (2020) and append those task markers to the end of the input sequence.

Input	Output (hierarchical)	Output (linear)
quest: My unicorn that **hasn't** amused the yaks **has** eaten.	**Has** my unicorn that hasn't amused the yaks eaten?	**Hasn't** my unicorn that amused the yaks has eaten?
quest: Die Hunde, die deine Löwen bewundern **können**, **haben** gewartet.	**Haben** die Hunde, die deine Löwen bewundern können, gewartet?	**Können** die Hunde, die deine Löwen bewundern, haben gewartet?
passiv: Her walruses above **my unicorns** annoyed **her quail**.	**Her quail** was annoyed by her walruses above my unicorns.	**My unicorns** were annoyed by her walruses.
passiv: Unsere Papageie bei **meinen Dinosauriern** bedauerten **unsere Esel**.	**Unsere Esel** wurden von unseren Papageien bei meinen Dinosauriern bedauert.	**Meine Dinosaurier** wurden von unseren Papageien bedauert.

Table 2: Examples from the generalization set with hierarchical- and linear-rule transformations.

3 Experimental Setup

3.1 Data

We modify and supplement the context-free grammar of McCoy et al. (2020) to generate our training and evaluation data.[3] For each transformation, our training data consists of 100,000 examples with an approximately 50/50 split between identity examples (where the input and output sequences are the same) and transformed examples. The identity examples include the full range of declarative or active structures (including sentences with RCs/PPs on subjects), thereby exposing the network to the full range of input structures we test. For the transformed examples, however, training data includes only examples with no RCs/PPs or RCs/PPs on the object NP—i.e., cases that are compatible with both the hierarchical and linear rules. We also generate development and test sets consisting of 1,000 and 10,000 examples, respectively, containing sentences with structures like those used in training; these are for evaluating in-distribution transformations on unseen sentences.

For each transformation, we also generate a generalization set consisting of 10,000 transformed examples with RCs/PPs on the subject NP. For such examples, models relying on the linear rules will not generalize correctly.

3.2 Models

We experiment with T5 (Raffel et al., 2020), an English pre-trained sequence-to-sequence model, as well as its multilingual extension mT5 (Xue et al., 2021).[4] This is a 12-layer Transformer-based (Vaswani et al., 2017) architecture. For fine-tuning on syntactic transformations, we use batch size 4 and initial LR 5×10^{-5}. (m)T5 converges and overfits quickly to the training set, so we only fine-tune for 1 epoch and evaluate every 500 iterations.

To confirm the finding of McCoy et al. (2020) that non-pre-trained models fail to generalize hierarchically, we also implement baseline seq2seq models similar to those used in that study. We implement 1- and 2-layer LSTM-based seq2seq models, as well as 1- and 2-layer Transformer-based seq2seq models where the Transformers have 4 attention heads.[5] We find that the 1-layer models consistently achieve higher sequence accuracies on the dev sets than the 2-layer models, so we focus on the 1-layer baselines. We re-use all hyperparameters from McCoy et al. (2020), additionally limiting the number of training epochs to 100. All baseline scores are averaged over 10 runs.

3.3 Metrics

For all transformations, we are primarily interested in *sequence accuracy*: is each token in the target sequence present in the proper order in the predicted sequence? However, it is possible that the model could generalize hierarchically while making some other mistake, so we also use two more relaxed metrics: *main auxiliary accuracy* for question formation, which evaluates whether the correct auxiliary was moved to the front of the sentence; and *object noun accuracy* for passivization, which measures whether the correct object noun was moved to the subject position. In the question formation task, the first word in the target sequence is always the main auxiliary verb, so we calculate main auxiliary accuracy by checking if the first word is the same in the predicted and target sequences. In the passiviza-

[3] We artificially generate our evaluation set such that it consists of grammatical but semantically improbable sentences which are unlikely to occur in a natural language corpus. This is to alleviate the confound of token collocations in the pre-training corpus.

[4] We use the HuggingFace implementations (Wolf et al., 2020).

[5] Our implementations are based on the syntactic-transformation-focused transductions repository: `https://github.com/clay-lab/transductions`

	Question Formation		Passivization	
Model	English	German	English	German
LSTM	0.95	0.94	0.97	0.97
Transformer	0.95	0.93	0.98	0.98
T5	1.00	–	1.00	–
mT5	1.00	1.00	1.00	1.00

Table 3: Sequence accuracies on the (in-distribution) test sets for English and German syntactic transformations. All models learn the in-distribution transformations.

	Question Formation		Passivization	
Model	English	German	English	German
LSTM	0.11	0.33	0.05	0.44
Transformer	0.07	0.05	0.04	0.07
T5	0.87	–	1.00	–
mT5	0.99	1.00	1.00	1.00

Table 4: Main auxiliary accuracies (for question formation) or object noun accuracies (for passivization) on the generalization sets for English and German syntactic transformations. Only T5 and mT5 generalize hierarchically.

tion task, the second word in the target sequence is the original object noun, so we calculate object noun accuracy by checking if the second word is the same in the predicted and target sequences.

4 Results

All models learn the in-distribution transformations. We first present results on unseen sentences with the structure seen in training, where both the hierarchical and the linear rules result in correct generalization (Table 3). All models perform well in this setting, including the LSTM- and Transformer-based models trained from scratch on this task. However, English and multilingual T5 converge to higher sequence accuracies on both languages and tasks than the non-pre-trained models. Additionally, while the baselines require about 15–20 epochs of training to converge to a high score, (m)T5 converges to perfect sequence accuracy after only a fraction of an epoch of fine-tuning.

Only pre-trained models generalize hierarchically. Evaluations on the generalization-set examples with RCs/PPs on subjects (i.e., examples where the linear rule leads to incorrect generalization) reveal that that none of the baseline models have learned the hierarchical rule. These models consistently stay at or near 0% sequence accuracy on the generalization set throughout training, so we present main auxiliary/object noun accuracies (Table 4). Accuracy remains low even on these more forgiving metrics, indicating that the baselines have not acquired the hierarchical rules.

Low accuracies do not necessarily indicate reliance on the linear MOVE-FIRST or MOVE-SECOND rules, since the baseline models could be using other heuristics to perform the transformations. To test whether the baselines have learned the linear rules, we implement metrics which calculate the proportion of generalization-set examples for which the MOVE-FIRST rule (for question formation) or MOVE-SECOND rule (for passivization) were used; we refer to these as the move-first frequency and move-second frequency, respectively. For each baseline and language, the sum of the main auxiliary accuracy and move-first frequency for question formation is ≈ 1.00; the sum of the object noun accuracy and move-second frequency for passivization is also ≈ 1.00. Thus, in most cases where the model did not move the main auxiliary or object noun, it used the linear rule to move the incorrect word. In other words, **the baseline models generalize using the linear rules.** This finding is in line with prior evaluations of non-pre-trained seq2seq models (McCoy et al., 2020; Mulligan et al., 2021; Petty and Frank, 2021).[6]

In contrast, (m)T5 achieves very high main auxiliary/object noun accuracies on the generalization set. Even more strikingly, (m)T5 also consistently achieves high sequence accuracies.[7] Because sequence accuracy on the generalization set is unstable, we present learning curves for mT5 (Figure 1) for the first epoch of fine-tuning. While the sequence accuracy is not consistently at 100%, it is generally very high for mT5; this is far better than the baselines' 0% sequence accuracies. This indicates that **T5 and mT5 demonstrate a hierarchical inductive bias**, and that they can quickly learn syntactic transformations.

Is (m)T5's hierarchical inductive bias a feature of the deep architecture, or is this bias acquired during pre-training? To test this, we randomize the

[6] Nonetheless, higher accuracies on German transformations support the hypothesis that more explicit cues to syntactic structure (here, case-marked articles and nouns) allow models to learn hierarchical syntactic generalizations more easily. This agrees with the findings of Ravfogel et al. (2019) and Mueller et al. (2020).

[7] T5 performs very similarly to mT5, so we present results for T5 in Appendix A.

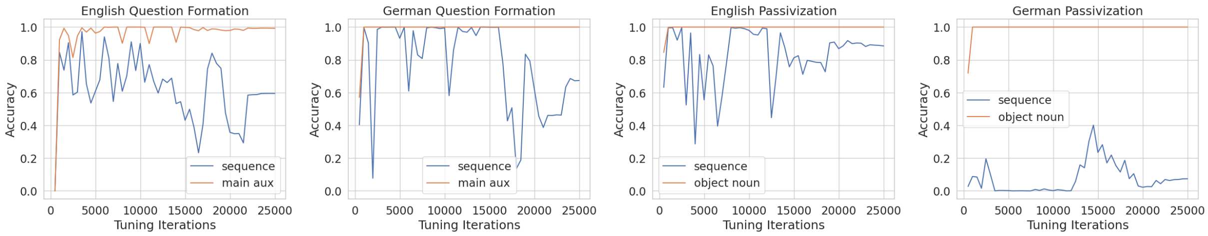

Figure 1: Learning curves during the first epoch of fine-tuning for mT5 on both transformation tasks in English and German.

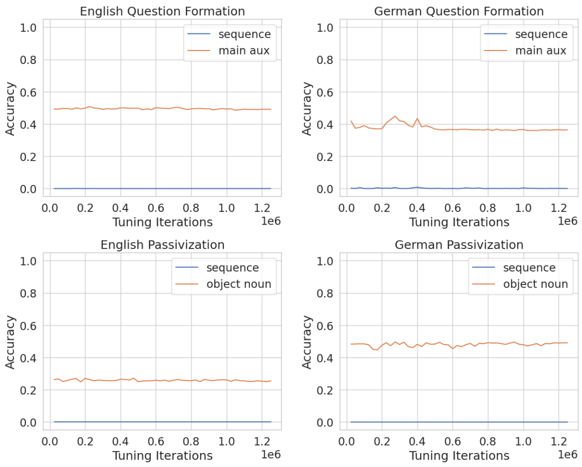

Figure 2: Learning curves for mT5 with randomized weights on the generalization set. Note: the x-axis is scaled by 1,000,000.

weights of mT5 and fine-tune for up to 50 epochs using an initial LR of 5×10^{-5}.[8] For all of the transformations, accuracies are much lower than for the pre-trained models (Figure 2), which suggests that the deeper architecture on its own does not lead to structure-sensitive generalizations. This in return indicates that **mT5 does not start with a hierarchical inductive bias; the model *acquires* it through pre-training**, extending the findings of Warstadt and Bowman (2020) to the generative sequence-to-sequence setting. However, as indicated by the non-zero main auxiliary/object noun accuracies, the randomly initialized mT5 models—unlike the baseline models—do not exhibit a consistent linear generalization either. This may be due to the large number of parameters compared to the size of the transformations training corpus. A randomly initialized model of this size would likely need orders of magnitude more training data to learn any stable generalizations.

[8]We tune over learning rates $\in 5 \times 10^{\{-2,-3,-4,-5\}}$ for the randomized models, finding that this setting yields the best main auxiliary and object noun accuracies on in-domain evaluations.

Error Analysis. T5 and mT5 almost always choose the correct auxiliary/object to move; what errors account for their sub-perfect sequence accuracies? We implement more specific metrics to observe more closely what mistakes (m)T5 makes.

Figure 3 depicts results for German passivization, the transformation with the lowest sequence accuracy. mT5 is almost always successful at the hierarchical transformation of moving the object NP to subject position (including its attached PP when present), and it correctly moves the original subject noun to a "by" phrase following the auxiliary. However, the model fails to preserve the PP on the second NP (in the by-phrase). We find the same results on English passivization for both T5 and mT5: discrepancies between sequence accuracy and object noun accuracy are almost always due to the model dropping the PP on the second NP in the target sequence. For example, "My yaks below the unicorns comforted the orangutans." is often transformed to "The orangutans were comforted by my yaks.", where the PP "below the unicorns" has not been moved with "my yaks". As mT5 has not been fine-tuned on output sequences where PPs appear at the end of the sentence, perhaps the decoder assigns very low probability to end-of-sentence PPs while otherwise encoding a hierarchical analysis of sentence structure.

Errors for question formation are more varied. T5 and mT5's sub-perfect main auxiliary accuracy on question formation is mainly due to improper negations on the main auxiliary: when the noun in the relative clause and the main noun agree in number, (m)T5 will sometimes delete the main auxiliary (as expected) while copying the incorrect auxiliary to the beginning of the sentence. Additionally, the discrepancy between sequence and main auxiliary accuracies is almost always attributable to (m)T5 not deleting the main auxiliary after moving it to the start of the sentence. These results (as with the passivization results) suggest that (m)T5 is actually

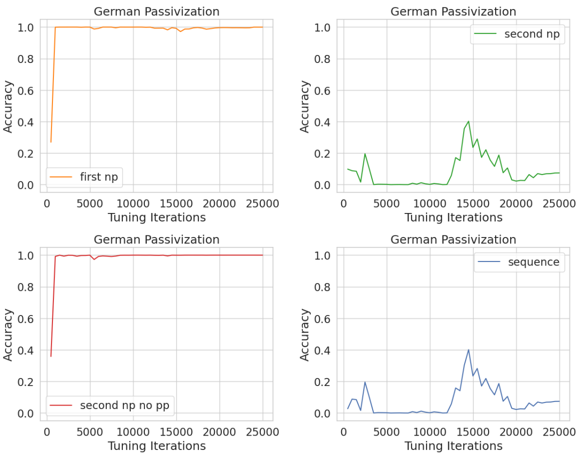

Figure 3: Learning curves displaying alternative accuracy metrics for mT5 on German passivization. We present the accuracy of the model in properly moving the object NP to the start of the sentence (top left), moving the subject NP after the auxiliary verb (top right), moving the subject NP after the auxiliary verb with or without its attached PP (bottom left), and the full sequence accuracy (bottom right).

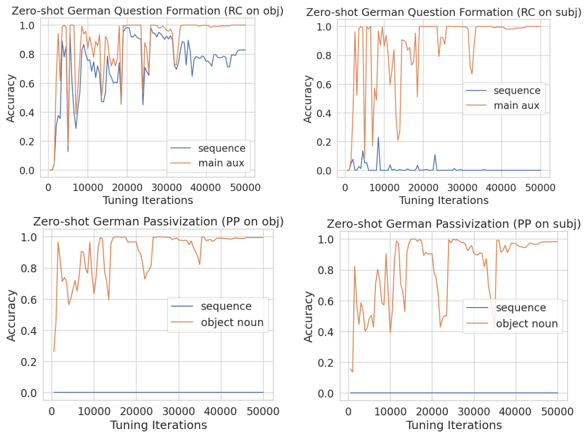

Figure 4: Learning curves for German transformations after tuning only on English/German identity examples and English transformations. We show accuracies for German question formation with RCs on objects (top left) and RCs on subjects (top right), as well as accuracies for German passivization with PPs on objects (bottom left) and PPs on subjects (bottom right).

better at performing hierarchy-sensitive transformations than the learning curves initially suggest—but also that (m)T5 can fail to perform theoretically simpler operations, such as deletions and moving all parts of a constituent.

5 Transformation Strategies

Our results indicate that (m)T5 can consistently perform hierarchy-sensitive transformations. What strategy does the model follow to do this? Because (m)T5's pre-training data includes active, passive, declarative, and question sentences, the model representations could encode these high-level sentence features.[9] Thus, one strategy could be to learn a mapping between abstract representations of different sentence structures (REPRESENTATION strategy). Alternatively, the model could learn to correctly identify the relevant syntactic units in the input (e.g., the main auxiliary for question formation, and the subject and object NPs for passivization), and then learn a "recipe" of steps leading to the correct transformations, such as those outlined in Section 2 (RECIPE strategy).

To distinguish which strategy (m)T5 uses to perform syntactic transformations, we exploit that English and German use the same operations for question formation, whereas passivization in German involves the additional steps of case reinflection and moving the main verb to the sentence-final position. Thus, if structural representations are shared across English and German in mT5,[10] we expect divergent behaviors for question formation and passivization: if the model employs the REPRESENTATION strategy, we expect it to also correctly turn German active sentences into passive sentences, including the additional steps of case reinflection and moving the main verb. Conversely, if the model employs the RECIPE strategy, we expect a model trained on English passivization to only perform the steps that are required for English passivization, resulting in reordered noun phrases with incorrect case marking and no main verb movement in German.

We first verify that mT5 is capable of cross-lingual transfer by training a model on the English question formation task and evaluating on German. In early experiments, we noticed the issue of "spontaneous translation" (Xue et al., 2021); we therefore also include German declarative identity transformations in the training data to train the decoder to also output German sentences.

As the top two panels of Figure 4 show, an mT5 model that has been fine-tuned for English question formation can correctly perform German question formation, especially on in-domain structures (where RCs are attached to the object). For out-of-

[9] For example, (sets of) neuron activations have been found to encode syntactic features in MLMs (Ravfogel et al., 2021; Finlayson et al., 2021; Hernandez and Andreas, 2021).

[10] Shared cross-lingual structural representations have been found for multilingual MLMs (Chi et al., 2020), and we provide further evidence for shared representations in this section.

domain structures (where RCs are attached to the subject), mT5 almost always moves the main auxiliary but almost never deletes it from its original position (which we found to a lesser extent in §4), resulting in lower sequence accuracies. Apart from this error, the model is capable of cross-lingual transfer on the question formation task.

Given that cross-lingual transfer seems possible, how does the model behave in the passivization task, which differs between English and German? We fine-tune mT5 on the English passivization task (as well as German identity transformations on active sentences). The results of this experiment (the lower two panels in Figure 4) show that the model is still able to move the main object to the subject position, but also that it never correctly performs German passivization in its entirety. This is because the model performs exactly the same steps for German sentences as for English sentences: it moves the object NP to the subject position, moves the subject NP to a prepositional phrase headed by 'by' instead of the German 'von', inserts the English auxiliaries 'was' or 'were' instead of the correct German 'werden', and performs neither case reinflection nor movement of the main verb to sentence-final position. This results in mixed German-English outputs such as *"meinen Kater bei ihrem Molch* ***was*** *verwirrten* ***by*** *ihre Esel."*

These patterns of behavior suggest that (m)T5 is learning the RECIPE strategy: it succeeds if a transformation's required operations are the same across languages (as for question formation) but fails if the steps differ (as for passivization). Even in passivization, however, the model still learns to move the correct NPs, which provides additional evidence that mT5 makes use of structural features when performing transformations.

6 Discussion

Our experiments provide evidence that pre-trained seq2seq models such as (m)T5 acquire a hierarchical inductive bias through exposure to non-annotated natural language text. This extends the findings of Warstadt and Bowman (2020) and Warstadt et al. (2020) to a more challenging generative task, where models cannot rely on n-gram distributional heuristics (Kam et al., 2008). In general, noising and denoising subsets of input sequences appears to be a powerful training objective for inducing linguistic generalizations in different neural architectures—including sequence-to-sequence architectures—especially when data is abundant.

Counter to McCoy et al. (2020), our findings suggest that hierarchical architectural constraints (e.g., tree-structured networks) are not necessary for robust hierarchical generalization as long as the model has been exposed to large amounts of natural language text. However, one difference between the randomly initialized models employed by McCoy et al. (2020) and pre-trained models is that pre-trained models have likely seen the structures (but not sentences) present in the generalization set; thus, rather than relying on syntactic features, the model could choose the correct transformation because it is more similar to the grammatical examples it has already seen. While we cannot fully rule out this possibility, it seems unlikely given that mT5 produces ungrammatical transformations, both in monolingual transformations (e.g., not deleting the main auxiliary after copying it to the start of the sentence) and in cross-lingual German passivization.

More broadly, our findings seem to counter the assumption that a hierarchical constraint is necessary in language learners to acquire hierarchical generalization (Chomsky, 1965). However, we note that T5's pre-training corpus contains far more input than a child would receive, and this corpus is also likely to contain the "disambiguating examples" that Chomsky (1965) argues are not present in children's input. More work is needed on models pre-trained on input comparable to what a child receives; for example, Huebner et al. (2021) evaluate grammaticality judgments of models trained on much smaller child-directed speech corpora.

7 Conclusions

We have performed an analysis of the syntactic transformational ability of large pre-trained sequence-to-sequence models. Our findings indicate that both monolingual and multilingual T5 acquire a hierarchical inductive bias during pre-training, and that the architecture does not yield this hierarchical bias by itself.

It remains an open question whether a model this deep and highly parameterized and a pre-training dataset so vast is necessary for hierarchical generalization. Future work could perform ablations over model depth and pre-training corpus size to observe the relative contribution of architecture and the training set to inducing a hierarchical inductive bias in seq2seq models.

References

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *Computing Research Repository*, arXiv:1901.05287.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93, Online. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

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 the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

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.

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. Citeseer.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

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*, pages 2096–2101.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020.

Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Karl Mulligan, Robert Frank, and Tal Linzen. 2021. Structure here, bias there: Hierarchical generalization by jointly learning syntactic transformations. *Proceedings of the Society for Computation in Linguistics*, 4(1):125–135.

Jackson Petty and Robert Frank. 2021. Transformers generalize linearly. *Computing Research Repository*, arXiv:2109.12036. ArXiv preprint arXiv:2109.12036.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. 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, 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.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive Science*, 30(5):945–982.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

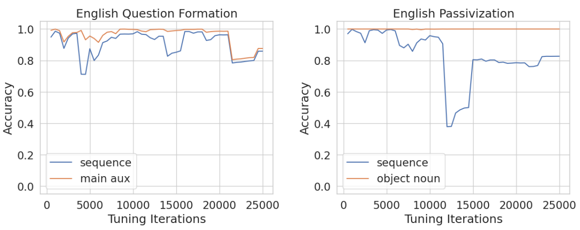

Figure 5: Learning curves during the first epoch of fine-tuning for monolingual (English) T5 on both syntactic transformation tasks.

A Monolingual T5 Results

Here, we present learning curves for the first epoch of fine-tuning on the English question formation and English passivization tasks for T5 (Figure 5). While T5 generally demonstrates the same hierarchical inductive bias that mT5 does, there are some discrepancies between the English and multilingual models. First, T5's sequence accuracies are generally more stable than mT5's, though main auxiliary and object noun accuracies are still unstable throughout fine-tuning. This is perhaps to be expected, as mT5 must acquire hierarchical inductive biases for many languages simultaneously, whereas T5 can devote its entire set of parameters to generalizing solely on English grammatical constructions.

Main auxiliary accuracy accuracy, however, is more unstable for T5 than mT5. This is unexpected, as T5 and mT5 generally achieve perfect main auxiliary and object noun accuracies after 1000 iterations of fine-tuning. This sub-perfect accuracy is due to improper negation on the inverted auxiliary, as was found for mT5: when the noun in the relative clause and the main noun agree in number, T5 sometimes delete the main auxiliary (as expected) while copying the incorrect auxiliary to the beginning of the sentence.

B German Structures

Here, we present examples of the sentences in the training, development, test, and generalization sets for the German question formation and passivization tasks (Table 5). As in English, we train the model on declarative or active sentences, as well as question-formation or passivization examples with no RCs/PPs or with RCs/PPs on subjects (i.e., sentences that are consistent with the hierarchical and linear rules described in §3.1). Then we evaluate its generalization on sentences where the linear rule does not properly transform the sentence.

For further clarity, we present glossed examples of each German structure below for both tasks.

(4) *German Question Formation (no RC)*:

a. Unsere Salamander haben die
Our.NOM salamanders have the.ACC
Pfaue bewundert.
peacocks admired.
"Our salamanders have admired the peacocks."

b. Haben unsere Salamander die
Have our.NOM salamanders the.ACC
Pfaue bewundert?
peacocks admired?
"Have our salamanders admired the peacocks?"

(5) *German Question Formation (RC on object)*:

a. Einige Molche können meinen Papagei,
Some.NOM newts can my.ACC parrot,
der deinen Raben trösten kann,
that.NOM your.ACC ravens comfort can,
nerven.
annoy.
"Some newts can annoy my parrot that can comfort your ravens."

b. Können einige Molche meinen Papagei,
Can some.NOM newts my.ACC parrot,
der deinen Raben trösten kann,
that.NOM your.ACC ravens comfort can,
nerven?
annoy?
"Can some newts annoy my parrot that can comfort your ravens?"

(6) *German Question Formation (RC on subject)*:

a. Ihr Hund, den ihr Geier
Your.NOM dog, that.ACC your.NOM vulture
nerven kann, hat einige Pfauen
annoy can, has some.ACC peacocks
amüsiert.
amused.
"Your dog that can annoy your vulture has amused some peacocks."

b. Hat ihr Hund, den ihr
Has your.NOM dog, that.ACC your.NOM
Geier nerven kann, hat einige
vulture annoy can, some.ACC peacocks
Pfauen amüsiert.
amused?
"Has your dog that can annoy your vulture amused some peacocks?"

(7) *German Passivization (no PP)*:

a. Ihr Kater bedauerte den
Your.NOM cat pities the.ACC
Dinosaurier.
dinosaur.
"Your cat pities the dinosaur."

b. Der Dinosaurier wurde von ihrem
The.NOM dinosaur became from your.DAT
Kater bedauert.
cat pitied.
"The dinosaur was pitied by your cat."

(8) *German Passivization (PP on object)*:

a. Unsere Ziesel amüsierten einen
Our.NOM ground-squirrels amuse a.ACC
Kater hinter dem Dinosaurier.
cat behind the.DAT dinosaur.
"Our ground squirrels amuse a cat behind the dinosaur."

b. Ein Kater hinter dem Dinosaurier
A.NOM cat behind the.DAT dinosaur
wurde von unseren Zieseln
became from our.DAT ground-squirrels
amüsiert.
amused.
"A cat behind the dinosaur was amused by our ground squirrels."

(9) *German Passivization (PP on subject)*:

a. Die Geier hinter meinem
The.NOM vultures behind my.DAT
Ziesel akzeptieren die Molche.
ground-squirrel accept the.ACC newts.
"The vultures behind my ground squirrel accept the newts."

b. Die Molche wurden von den
The.NOM newts became from the.DAT
Geiern hinter meinem Ziesel
vultures behind my.DAT ground-squirrel
akzeptiert.
accepted.
"The newts were accepted by the vultures behind my ground squirrel."

	Train, dev, test	Generalization
Question Formation	Declarative	Question
No RC	decl: unsere Salamander haben die Pfaue bewundert. → unsere Salamander haben die Pfaue bewundert.	quest: ihre Hunde haben unseren Orang-Utan genervt. → haben ihre Hunde unseren Orang-Utan genervt?
RC on object	decl: unser Ziesel kann den Salamander, der meinen Pfau verwirrt hat, akzeptieren. → unser Ziesel kann den Salamander, der meinen Pfau verwirrt hat, akzeptieren.	quest: einige Molche können meinen Papagei, der deinen Raben trösten kann, nerven. → können einige Molche meinen Papagei, der deinen Raben trösten kann, nerven?
RC on subject	decl: dein Molch, den mein Wellensittich bewundert hat, kann meine Dinosaurier trösten. → dein Molch, den mein Wellensittich bewundert hat, kann meine Dinosaurier trösten.	quest: ihr Hund, den ihr Geier nerven kann, hat einige Pfaue amüsiert. → hat ihr Hund, den ihr Geier nerven kann, einige Pfaue amüsiert?
Passivization	Active	Passive
No PP	decl: die Löwen unterhielten einen Wellensittich. → die Löwen unterhielten einen Wellensittich.	passiv: ihr Kater bedauerte den Dinosaurier. → der Dinosaurier wurde von ihrem Kater bedauert.
PP on object	decl: ihre Geier verwirrten ihren Raben über unserem Ziesel. → ihre Geier verwirrten ihren Raben über unserem Ziesel.	passiv: unsere Ziesel amüsierten einen Kater hinter dem Dinosaurier. → ein Kater hinter dem Dinosaurier wurde von unseren Zieseln amüsiert.
PP on subject	decl: ein Löwe unter unserem Hund nervte einigie Ziesel. → ein Löwe unter unserem Hund nervte einigie Ziesel.	passiv: die Geier hinter meinem Ziesel akzeptieren die Molche. → die Molche wurden von den Geiern hinter meinem Ziesel akzeptiert.

Table 5: The distribution of syntactic structures in the German train, test, and generalization sets. We use the test set to evaluate whether models have learned the task on in-distribution examples, and the generalization set to evaluate hierarchical generalization.