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

Aaron Mueller[c] **Robert Frank**[y] **Tal Linzen**[3]
Luheng Wang[3] **Sebastian Schuster**[3]
[c]Johns Hopkins University [y]Yale University [3]New York University
amueller@jhu.edu, schuster@nyu.edu

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. 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. We address this gap using the pre-trained seq2seq models T5 and BART, as well as their multilingual variants mT5 and mBART. We evaluate whether they generalize hierarchically on two transformations in two languages: question formation and passivization in English and German. We find that pre-trained seq2seq models generalize hierarchically when performing syntactic transformations, whereas models trained from scratch on syntactic transformations do not. This result presents 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, though only after exposure to much more language data than human learners receive.

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.

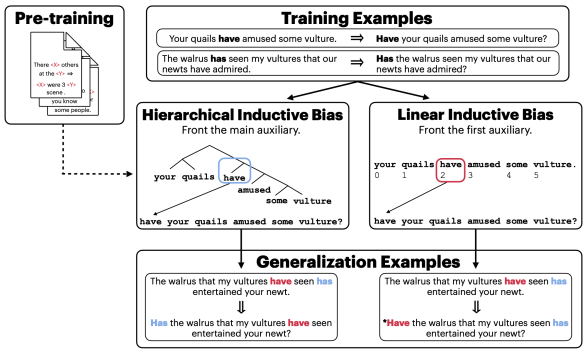

Figure 1: The poverty of the stimulus experimental design. We fine-tune pre-trained seq2seq models and train small seq2seq models from scratch to perform syntactic transformations. The training set contains ambiguous examples consistent with hierarchical and linear transformation rules. The generalization set contains examples where only the hierarchical rule results in the correct output. Pre-trained models generalize using the hierarchical rule, while models trained from scratch generalize using the linear rule.

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), though acquiring 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 fine-tuned to classify which sentence in a given minimal pair is consistent with a hierarchical linguistic generalization, rather than a positional surface heuristic. Consider the following

Findings of the Association for Computational Linguistics: ACL 2022, pages 1352 - 1368
May 22-27, 2022

two transformations of Example (1):

(1) The yak that your unicorns **have** amused **hasn't** entertained a newt.
   a. **Hasn't** the yak that your unicorns **have** amused entertained a newt?
   b. ***Have** the yak that your unicorns amused hasn't entertained a newt?

Example (1a) correctly forms the question by moving the main auxiliary verb to the front of the sentence, while (1b) relies on the incorrect positional heuristic that the first auxiliary in the declarative sentence should be moved to the front of the sentence. When differentiating grammatical and ungrammatical auxiliary movements, 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 (1a)—is more difficult. It requires multiple complex but systematic operations that rely on hierarchical structure, including movement, number agreement, and—in languages that have grammatical case, such as German—case reinflection. 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) models (Sutskever et al., 2014) 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 considered recurrent seq2seq models and Transformer models (Petty and Frank, 2021) trained from scratch 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, which is why they do not pre-train the models on non-annotated natural language data before training them to perform syntactic transformations.

In this study, we create German datasets and modify English datasets for evaluating the inductive biases of pre-trained models. We use these datasets to analyze performance in monolingual and zero-shot cross-lingual settings. Further, we analyze *how* pre-trained models perform syntactic transformations. Our findings indicate that pre-trained models generally perform syntactic transformations in a hierarchy-sensitive manner, while non-pre-trained models (including randomized-weight versions of pre-trained models) rely primarily on linear/positional heuristics to perform the transformations. This finding presents additional evidence to Warstadt et al. (2020) and Warstadt and Bowman (2020) for the learnability of hierarchical syntactic information from natural language text input. Our code and data are publicly available.[1]

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 (Figure 1).[2] In other words, we are interested in whether T5 and mT5 (henceforth, (m)T5), as well as BART and mBART (henceforth, (m)BART), demonstrate a **hierarchical inductive bias**,[3] unlike the linear inductive bias displayed in prior work by non-pre-trained models.

[1] https://github.com/sebschu/multilingual-transformations

[2] 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.

[3] 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. 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 A for example sentences in German.

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. See Table 2 for examples of hierarchical and linear generalizations for each transformation.

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. Examples of both rules are provided in Figure 1 and Example (1). We train the model on sentences with no relative clauses (RCs) or with RCs on the object, where the first auxiliary verb is always the matrix verb. Disambiguating examples are those which place RCs on the subject, where the matrix auxiliary verb is the linearly second auxiliary in the sentence.

In English, we use the auxiliaries "has", "hasn't", "have", and "haven't", with past participle main verbs (e.g., "have *entertained*", "has *amused*"). We use affirmative and negative forms to distinguish between the multiple auxiliaries: exactly one of the auxiliaries in such sentences is negative and the other is positive (counterbalanced across examples). As a result, we can determine whether the induced mapping is linear or hierarchical. In German, negation is realized as a separate word that is not fronted with the auxiliary. To distinguish the multiple auxiliaries, we therefore use the modal "können" (*can*) along with the auxiliary "haben" (*have*), together with infinitival or past participle main verbs as appropriate. 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). Our training examples 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 no prepositional phrases (PPs) or with PPs modifying the object, where the second NP is always the object. Disambiguating examples are those which place prepositional phrases (PPs) on the subject, where the object is the linearly third NP in the sentence.

Passivization additionally requires other movements, insertions, tense reinflection, and (for German) case reinflection. In Examples (2) and (3) below, the object (in blue) is fronted; 'be'/'werden' (in red) is inserted and inflected to agree with the fronted NP; the original subject NP (in brown) is moved to a 'by'/'von' phrase after the inserted verb; and the main verb (in orange) is reinflected to be a past participle or infinitive. In German, 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
Your.NOM donkey entertained my.ACC
Salamander.
salamander.

b. Mein Salamander wurde von ihrem
My.NOM salamander was from your.DAT
Esel unterhalten.
donkey entertained.

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. Glossed German examples are provided in Appendix A.

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.[4] 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) and BART (Liu et al., 2020), two English pre-trained sequence-to-sequence models. We also experiment with their multilingual variants mT5 (Xue et al., 2021) and mBART (Liu et al., 2020).[5] These are 12-layer Transformer-based (Vaswani et al., 2017) architectures with bidirectional encoders and autoregressive decoders. While we use the base sizes of (m)T5, we use the large sizes of (m)BART to keep the sizes of the models similar.

When fine-tuning (m)T5 and (m)BART, we use task prefixes in the source sequence. 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 from scratch, we follow McCoy et al. (2020) and append the task markers to the end of the input sequence.

For fine-tuning on syntactic transformations, we use batch size 128 and initial learning rate 5×10^{-5}. We fine-tune for 10 epochs and evaluate every 500 iterations. We find that the validation loss generally converges within 1–2 epochs.

To confirm the finding of McCoy et al. (2020) and Petty and Frank (2021) that non-pre-trained models fail to generalize hierarchically, we also train baseline seq2seq models similar to the models used in those studies. 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.[6] We find that the 1-layer models consistently achieve higher sequence accuracies on the dev sets, so we focus on the 1-layer baselines. We re-use all hyperparameters from McCoy et al. (2020). 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 tar-

[4] We 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.

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

[6] 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
BART	0.96	–	0.95	–
mBART	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
BART	0.96	–	1.00	–
mBART	0.59	0.82	0.80	0.98

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 pre-trained models generalize hierarchically.

get sequence present in the proper order in the predicted sequence? However, it is possible that models could generalize hierarchically while making some other mistake, so we also use two more relaxed metrics. For question formation, we use *main auxiliary accuracy*, which evaluates whether the correct auxiliary was moved to the front of the sentence. 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. For passivization, we use *object noun accuracy*, which measures whether the correct object noun was moved to the subject position. The second word in the target sequence is always 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 whose structures were 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. However, (m)T5 converges to higher sequence accuracies than the non-pre-trained models. Additionally, while the non-pre-trained models require about 15–20 epochs of training to converge to a high score, (m)T5 and (m)BART converge to near-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 (where the linear rule leads to incorrect generalization) reveal that none of the trained-from-scratch 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 non-pre-trained models have not acquired the hierarchical rules.

Low accuracies do not necessarily indicate reliance on the linear MOVE-FIRST or MOVE-SECOND rules. To test whether the non-pre-trained models 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 model and language, the sum of the main auxiliary accuracy and move-first frequency for question formation is ≈ 1.0; the sum of the object noun accuracy and move-second frequency for passivization is also ≈ 1.0. Thus, where the model did not move the main auxiliary or object noun, it generally used the linear rule. In other words, **the non-pre-trained models demonstrate *linear* inductive biases.** 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).[7]

By contrast, (m)T5 and (m)BART achieve very high main auxiliary/object noun accuracies on the generalization sets. mBART struggles with En-

[7]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).

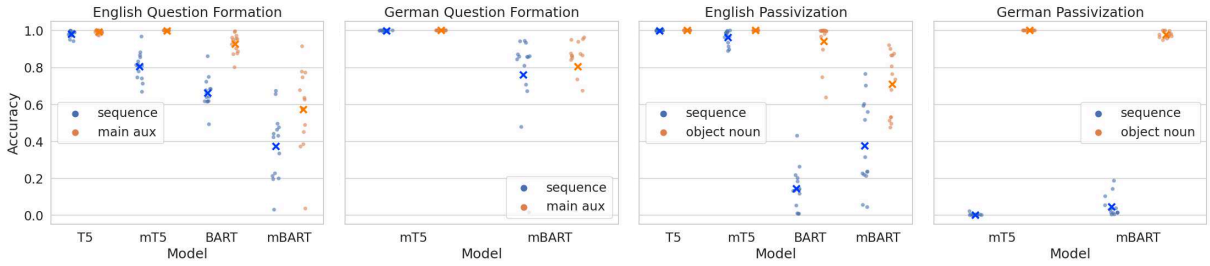

Figure 2: Accuracies at every 500 fine-tuning iterations across 10 epochs of fine-tuning on each syntactic transformation. Xs indicate mean accuracies across epochs. T5 models are generally better at performing syntactic transformations than BART models. Monolingual models tend to achieve higher accuracies than multilingual models. We present full learning curves in Appendix B.

glish question formation, achieving an average 59% main auxiliary accuracy throughout fine-tuning. However, it does achieve a maximum accuracy >90%, indicating that it is capable of hierarchical generalization after observing certain training examples. These accuracies are still well above the ≈0% accuracies of the non-pre-trained models.

Because sequence accuracy on the generalization set is often unstable for all pre-trained models, we present plots showing the distribution of accuracies sampled at every 500 fine-tuning iterations throughout 10 epochs of fine-tuning (Figure 2). Each pre-trained model learns the in-distribution transformation before the first 500 iterations of fine-tuning, so each plotted accuracy can be taken as indicative of model preferences after they have learned the transformations. (m)T5's sequence accuracies are generally close to 100% for all transformations except German passivization; this is far better than the non-pre-trained models' 0% sequence accuracies. (m)BART struggles more with syntactic transformations as indicated by its lower average accuracies, though it is still capable of detecting the correct auxiliaries and objects to move as indicated by the high *maximum* main auxiliary and object noun accuracies in Figure 2. This indicates that **pre-trained seq2seq models demonstrate a hierarchical inductive bias**, and that they can quickly learn syntactic transformations.

There are two main differences between the two classes of models we test: (m)T5 and (m)BART are not only pre-trained, but are also much deeper and much more parameterized than our non-pre-trained models. Are hierarchical inductive biases a feature of deep architectures, then, or are they acquired during pre-training? To control for pre-training while keeping the model size consistent, we randomize the weights of mT5 (the better-performing model)

	Question Formation		Passivization	
Model	English	German	English	German
T5	0.48	–	0.25	–
mT5	0.50	0.44	0.25	0.50
BART	0.40	–	0.30	–
mBART	0.48	0.38	0.29	0.44

Table 5: *Maximum* main auxiliary and object noun accuracies through 500 epochs of fine-tuning *after randomizing the weights* of each pre-trained model. Sequence accuracies remain near 0 throughout fine-tuning.

and fine-tune for up to 500 epochs using an initial LR[8] of 5×10^{-4}. For all of the transformations, the maximum accuracies of the randomized models are much lower than the average accuracies of the pre-trained models (Table 5), which suggests that the deeper architecture on its own does not lead to structure-sensitive generalizations. This in return indicates that **pre-trained models do not start with a hierarchical inductive bias; they *acquire* it through pre-training**, extending the findings of Warstadt and Bowman (2020) to generative sequence-to-sequence models. However, as indicated by the non-zero main auxiliary/object noun accuracies, the randomly initialized mT5 models do not exhibit a consistent linear generalization either—unlike the 1-layer non-pre-trained models. 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 stable generalizations.

Each pre-trained model almost always chooses the correct auxiliary/object to move; what errors

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

account for their sub-perfect sequence accuracies, then? We perform a detailed error analysis, finding that pre-trained models drop PPs from the second noun phrase but otherwise perform many complex hierarchy-sensitive transformations properly. See Appendix C for details.

5 Transformation Strategies

Our results indicate that pre-trained seq2seq models can consistently perform hierarchy-sensitive transformations. What strategy do they follow to do this? Because pre-training corpora include actives, passives, declaratives, and questions, 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, models could learn to correctly identify the relevant syntactic units in the input, and then learn a "recipe" of steps leading to the correct transformations (RECIPE strategy).

To distinguish which strategy models use to perform syntactic transformations, we observe cross-lingual zero-shot transfer on 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. If structural representations are shared across English and German,[10] we do not expect divergent behaviors for question formation and passivization: if a model employs the REPRESENTATION strategy, then after fine-tuning on only English passivization, it should also correctly perform German passivization, including the additional steps of case reinflection and moving the main verb. Conversely, if it 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 incorrect case marking and no main verb movement in German.

We first verify that mT5 and mBART are 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

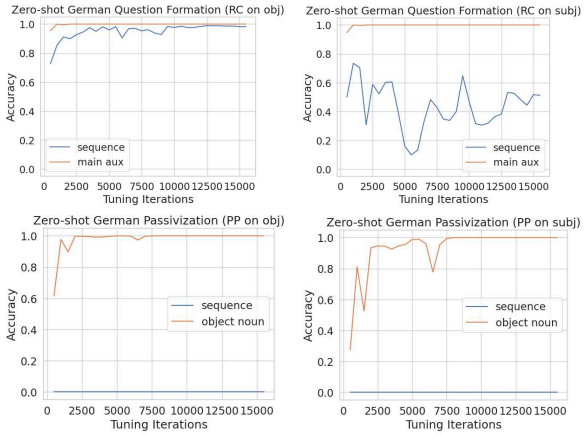

Figure 3: Learning curves for mT5 on German transformations after fine-tuning on English/German identity examples and English transformations. We show accuracies for German question formation with RCs on objects (top left) and RCs on subjects (top right), as well as accuracies for German passivization with PPs on objects (bottom left) and PPs on subjects (bottom right).

therefore also include German identity transformations in the training data to train the decoder to also output German sentences.

As the top two panels of Figure 3 show, mT5 can correctly perform German question formation on in-domain structures (RCs on objects) after being exposed only to English transformations. For out-of-domain structures (RCs on subjects), mT5 almost always moves the main auxiliary but almost never deletes it from its original position, resulting in lower sequence accuracies. Apart from this error, the model is capable of cross-lingual transfer on the question formation task. By contrast, mBART achieves poor results on zero-shot German question formation, so we cannot make conclusive arguments using this approach; see Appendix D.

Given that cross-lingual transfer is possible for mT5, how does the model behave in the passivization task, which differs between English and German? We fine-tune mT5 on English passivization (as well as German identity transformations on active sentences). The results of this experiment (the lower two panels in Figure 3) 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, resulting in outputs with English syntax:

[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.

(4) Meinen Kater bei ihrem Molch was
My.ACC cat by your.DAT newt was
verwirrten by die Esel.
confused.PAST by the.NOM donkeys.

These behavioral patterns suggest that mT5 employs 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. Given the similarities between mT5's and mBART's architectures and training setups, one could reasonably presume that mBART may follow a similar strategy to perform syntactic transformations; nonetheless, mBART is less consistent in performing syntactic transformations, so this method cannot present strong evidence for use of the RECIPE strategy for that model.

6 Corpus Analysis

Pre-trained models learn to use hierarchical features for performing syntactic transformations. Is this because there is explicit supervision for the hierarchical rules in the pre-training corpora? In other words, are there *disambiguating examples* in these models' training corpora that helps them memorize hierarchical transformation patterns? Here, we focus on English question formation examples in mT5's training corpus.[11] Disambiguating examples would be rare, as a single pre-training context window must contain a declarative sentence as well as the same sentence transformed into a question; humans would tend to replace at least some of the constituents with pronouns or delete them (e.g., Ariel, 2001). It would also require the MOVE-FIRST rule to not correctly transform the sentence—*and* for at least one of the auxiliaries to be noised in one sentence but not the other, such that the auxiliary has to be recovered from the other sentence. For example:

(5) . . . Has this company which hasn't had any legal violations been reported to the Better Business Bureau? This company which hasn't had any legal violations <X> been reported to the Better Business Bureau. . .

We search for English disambiguating question formation examples. To this end, we sample 5M English documents from mT5's training corpus mC4, segmenting each document into sentences using spaCy.[12] This yields 118.3M sentences. We examine each pair of adjacent sentences in each document, manually inspecting any sentence pair meeting the following criteria: (1) the token Jaccard similarity of the sentences is > 0.7; (2) one sentence begins with an auxiliary verb and the other does not; (3) there are at least two distinct auxiliaries in both sentences. There are 277 sentence pairs in our sample that met all criteria, of which 13 are adjacent declarative/question pairs that are equivalent except for the fronted auxiliary. Thus, the probability of an equivalent declarative/question pair with two auxiliaries in mC4 is $\approx 1.1 \times 10^{-7}$. As T5's and mBART's training corpora consist of data from similar webtext distributions, it is likely that these structures exist in those corpora as well.

Crucially, however, none of the declarative/question pairs were disambiguating examples: each pair was consistent with the linear MOVE-FIRST rule. What is the probability of a disambiguating example, then? If we assume that the probability of a sentence containing an RC on the subject is independent from the probability of a declarative/question sentence pair, we can take the product of both probabilities to obtain an estimate. From the same sample of 118.3M sentences, we use spaCy's dependency parser to extract sentences containing an RC on the subject and where at least one auxiliary verb appears in the sentence. We obtain $526,944$ such sentences, meaning that the probability of an RC on a subject in an auxiliary-containing sentence in mC4's English corpus is $\approx 4.5 \times 10^{-3}$. Thus, the probability of declarative/question pair with an RC on the subject and auxiliary in the RC is $\approx (4.5 \times 10^{-3}) \cdot (1.1 \times 10^{-7}) = 4.95 \times 10^{-10}$. mT5 is trained on up to 1T tokens of data, and 5.67% of its documents are English; it therefore observes ≈ 56.7B English tokens. If we optimistically assume that English sentences contain an average of 15 tokens, it observes 3.78B English sentences. Then we would expect $3.78\text{B} \times (4.95 \times 10^{-10}) \approx 2$ disambiguating examples. This is not including the auxiliary masking criterion, which would make such examples even less likely.

Thus, while we cannot definitively rule out the

[11] mT5 outperforms mBART. If disambiguating contexts in the pre-training data lead to syntactic generalizations, then we expect these examples to be more likely in mT5's training corpus.

[12] https://spacy.io

possibility of disambiguating examples in mC4, they are rare if they exist in the corpus at all. Nonetheless, we have found evidence for supervision on question formation in the form of adjacent declarative/question sentence pairs, even if they do not explicitly support the hierarchical rule.

7 Discussion

Our experiments provide evidence that pre-trained seq2seq models 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). This also provides additional evidence that masking and reconstructing subsets of input sequences is a powerful training objective for inducing linguistic generalizations, whether in masked language models like RoBERTa (Warstadt and Bowman, 2020) or sequence-to-sequence models. Span denoising ((m)T5's objective) appears more effective for learning syntactic transformations than full sequence reconstruction ((m)BART's objective) given that (m)T5 is more consistently able to perform transformations, though there are too many other differences in training data and hyperparameters between (m)T5 and (m)BART for us to be able to directly implicate the training objective. This hypothesis can be tested explicitly in future work by training identical models that differ only in their pre-training objective.

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—possibly far more language than humans would be exposed to. 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. We found declarative/question pairs in mT5's training corpus, but we did not find any examples that explicitly demonstrated the hierarchical rule for question formation. While we cannot fully rule out the possibility of disambiguating examples, this strategy is still unlikely given that pre-trained models produce 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. Additionally, because we use greedy decoding, models are not able to take future words into account when predicting the fronted auxiliary: they must select the appropriate auxiliary to move solely based on the encoder's representations.

More broadly, our findings counter the assumption that a hierarchical constraint is necessary in language learners to acquire hierarchical generalization (Chomsky, 1965). While the pre-trained models that we considered observe far more input than a child would receive (Linzen, 2020), Huebner et al. (2021) recently demonstrated high performance on grammaticality judgments for models trained on much smaller child-directed speech corpora, suggesting that our findings may also hold when training models on more human-like input.

8 Conclusions

We have performed an analysis of the syntactic transformational ability of large pre-trained sequence-to-sequence models. We find that pre-trained models 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 such deep and highly parameterized models or such large pre-training datasets are necessary for hierarchical generalization. Future work could ablate over model depth and pre-training corpus size to observe the relative contribution of architecture and the training set to inducing hierarchical inductive biases in seq2seq models.

Acknowledgments

We thank the members of NYU's Computation and Psycholinguistics Lab and the reviewers for their thoughtful feedback. We also thank R. Thomas McCoy for providing sentence generation scripts.

This material is based upon work supported by the National Science Foundation (NSF) under Grant #BCS-2114505, #BCS-1919321, as well as Grant #2030859 to the Computing Research Association for the CIFellows Project. Aaron Mueller was supported by an NSF Graduate Research Fellowship (Grant #1746891).

References

Mira Ariel. 2001. Accessibility theory: An overview. In *Text Representation: Linguistic and Psycholinguistic aspects*. John Benjamins, Amsterdam.

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, Boston, Massachusetts. Citeseer.

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

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, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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, Madison, Wisconsin. Cognitive Science Society.

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.

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, Montreal, Canada. Curran Associates, Inc.

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, Long Beach, California. 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*, Online. 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.

A 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 6). 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.

(6) *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?"

(7) *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?"

(8) *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?"

(9) *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 was from your.DAT
Kater bedauert.
cat pitied.
"The dinosaur was pitied by your cat."

(10) *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
was from our.DAT ground-squirrels
amüsiert.
amused.
"A cat behind the dinosaur was amused by our ground squirrels."

(11) *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 were 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."

B Learning Curves

Here, we present learning curves for 10 epochs of fine-tuning for each transformation in each language (Figures 4,5,6,7). The accuracies shown

	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 6: 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.

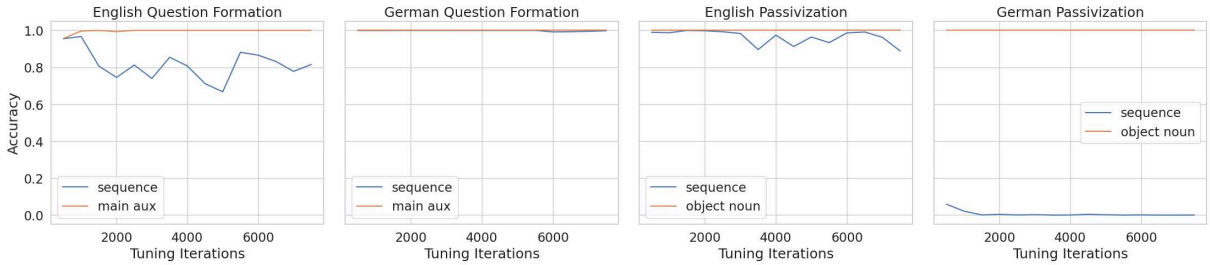

Figure 4: Learning curves over 10 epochs of fine-tuning for mT5 on both syntactic transformation tasks.

.

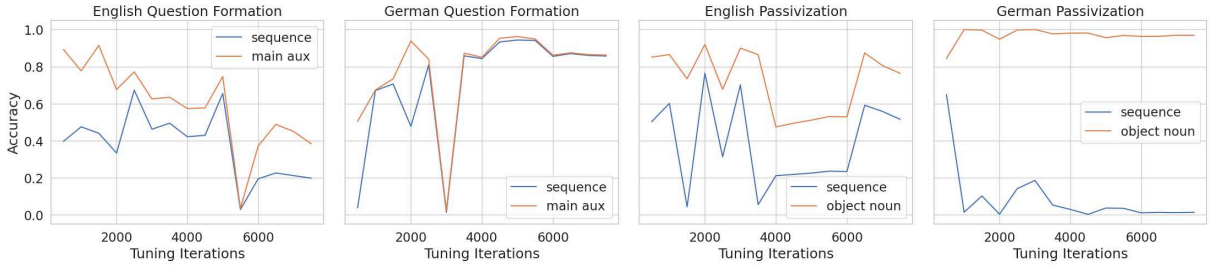

Figure 5: Learning curves over 10 epochs of fine-tuning for mBART on both syntactic transformation tasks.

.

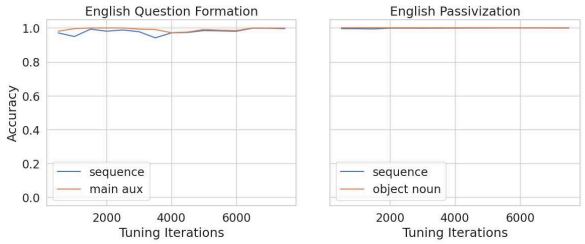

Figure 6: Learning curves over 10 epochs of fine-tuning for T5 on both syntactic transformation tasks.

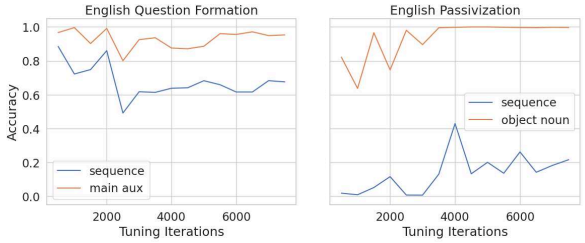

Figure 7: Learning curves over 10 epochs of fine-tuning for BART on both syntactic transformation tasks.

in these curves are the same as those shown in Figure 2, but now associated with their respective fine-tuning iteration.

All models except mBART immediately achieve near-perfect main auxiliary and object noun accuracies. Their loss on the validation sets converges almost immediately, so it's possible that reductions in generalization accuracies throughout fine-tuning are due to overfitting to the training distribution. For mBART, however, main auxiliary and object noun accuracies start high and then begin to vary dramatically throughout fine-tuning. This is perhaps due to quick overfitting on the training distribution. We analyze what errors cause mBART's deficiencies in §C.2.

C Error Analysis

Each pre-trained model almost always chooses the correct auxiliary/object to move; what other errors account for their sub-perfect sequence accuracies? We implement more specific metrics to observe more closely what mistakes (m)T5 and (m)BART are making. We show results for mT5 in §C.1 and mBART in §C.2, but the errors we discuss are generally consistent across models. We also present more detailed metrics for the most complex transformation, German passivization, in §C.3.

C.1 mT5

Figure 8 depicts results for mT5 for German passivization, the transformation on which all models

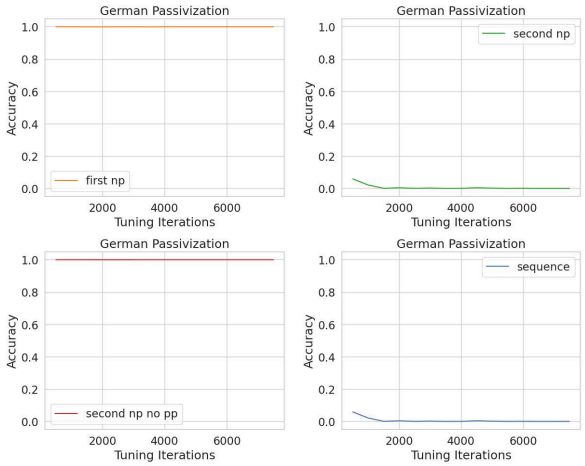

Figure 8: Learning curves displaying alternative accuracy metrics for mT5 on German passivization. We present the accuracy of the model in properly moving the object NP to the start of the sentence (top left), moving the subject NP after the auxiliary verb (top right), moving the subject NP after the auxiliary verb with or without its attached PP (bottom left), and the full sequence accuracy (bottom right).

achieve 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, for both English and German passivization, the main error accounting for sub-perfect sequence accuracies is that the model fails to preserve the PP on the second NP (in the by-phrase):

(12) My yaks below the unicorns comforted the orangutans.
→ The orangutans were comforted by my yaks.

As mT5 has not been fine-tuned on output sequences where PPs appear at the end of the sentence, the decoder could be assigning low probabilities to end-of-sentence PPs while otherwise encoding a hierarchical analysis of sentence structure.

Errors for question formation are more varied. Pre-trained models' sub-perfect main auxiliary accuracies on question formation are mainly due to improper negations on the main auxiliary: when the noun in the relative clause and the main noun agree in number, models 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 models not deleting the main auxiliary after moving it to the start of the sentence. These results (as with the passivization results) suggest that **pre-trained seq2seq models are better at performing hierarchy-sensitive transformations than the sequence accuracies initially suggest—but also that they can fail to perform theoretically simpler operations,** such as deletions and moving all parts of a constituent.

We present more detailed error analyses in Appendix C.3. We find that pre-trained models also consistently succeed in case reinflection, tense reinflection, and passive auxiliary insertion.

C.2 mBART

We have shown in §C.1 that mT5 achieves subperfect sequence accuracies on passivization due to its dropping the prepositional phrase on the second NP. Here, we present results for mBART (Figure 9). The takeaways for mBART are similar to mT5's: the model succeeds in moving the proper nouns, but it often drops the prepositional phrase from the second NP during movement.

As the model also fails to perform English question formation consistently, we also observe what errors it makes in that task. We find that deficiencies in main auxiliary accuracy are due to the model copying the incorrect auxiliary to the beginning of the sentence, while gaps between main auxiliary and sequence accuracy are due to the model dropping the relative clause on the second NP.

C.3 More Detailed Metrics

We also present more detailed analyses of other required operations in passivization: namely, are mT5 and mBART capable of tense reinflection, case reinflection, and auxiliary insertions? And are they capable of this in zero-shot settings? Results for mT5 (Figure 10) and mBART (Figure 11) suggest that both models are generally capable of tense reinflection, case reinflection, and auxiliary insertion in supervised contexts.

D Zero-shot mBART Accuracies

Here, we present learning curves for mBART on zero-shot cross-lingual syntactic transformations (Figure 12). While mBART is typically able to select the correct auxiliary verb or object noun to move, it never transforms the sequence fully correctly.

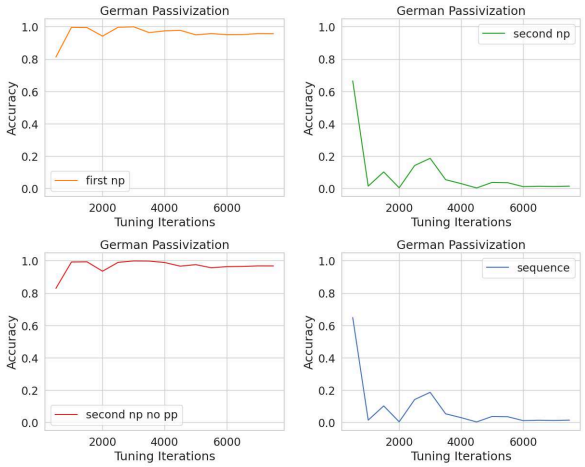

Figure 9: Learning curves displaying alternative accuracy metrics for mBART on German passivization. We present the accuracy of the model in properly moving the object NP to the start of the sentence (top left), moving the subject NP after the auxiliary verb (top right), moving the subject NP after the auxiliary verb with or without its attached PP (bottom left), and the full sequence accuracy (bottom right).

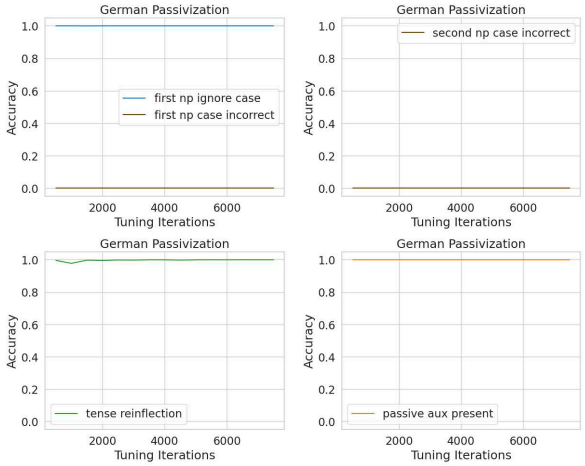

Figure 10: Learning curves displaying alternative accuracy metrics for mT5 on German passivization. We present the proportion of examples for which the model moves the first NP without reinflecting its case (top left), moves the second NP without reinflecting its case (top right), reinflects the tense of the main verb (bottom left), and inserts the passive auxiliary *werden* with the proper inflection.

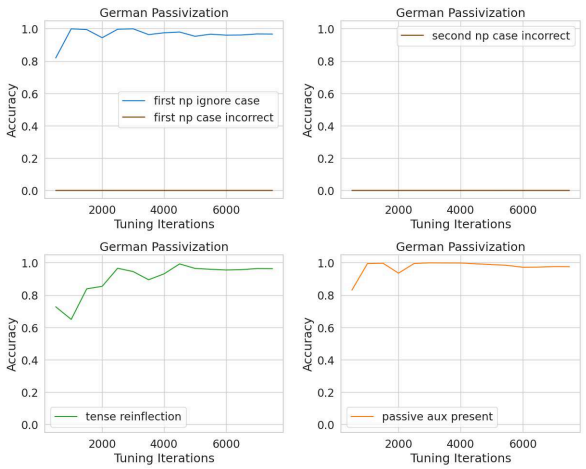

Figure 11: Learning curves displaying alternative accuracy metrics for mBART on German passivization. We present the proportion of examples for which the model moves the first NP without reinflecting its case (top left), moves the second NP without reinflecting its case (top right), reinflects the tense of the main verb (bottom left), and inserts the passive auxiliary *werden* with the proper inflection.

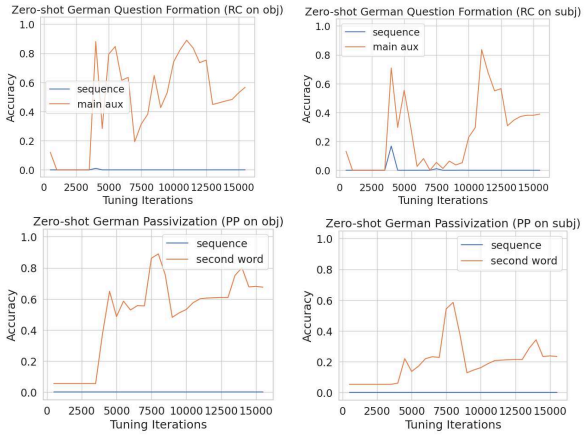

Figure 12: Learning curves for mBART on German transformations after fine-tuning only on English/German identity examples and English transformations. We show accuracies for German question formation with RCs on objects (top left) and RCs on subjects (top right), as well as accuracies for German passivization with PPs on objects (bottom left) and PPs on subjects (bottom right).