Systematic Generalization by Finetuning? Analyzing Pretrained Language Models Using Constituency Tests

Aishik Chakraborty **Jackie CK Cheung** **Timothy J. O'Donnell**
School of Computer Science, McGill University
Québec AI Instuite (Mila)
aishik.chakraborty@mail.mcgill.ca, {timothy.odonell, jackie.cheung}@mcgill.ca

Abstract

Constituents are groups of words that behave as a syntactic unit. Many linguistic phenomena (e.g., question formation, diathesis alternations) require the manipulation and rearrangement of constituents in a sentence. In this paper, we investigate how different finetuning setups affect the ability of pretrained sequence-to-sequence language models such as BART and T5 to replicate *constituency tests* — transformations that involve manipulating constituents in a sentence. We design multiple evaluation settings by varying the combinations of constituency tests and sentence types that a model is exposed to during finetuning. We show that models can replicate a linguistic transformation on a specific type of sentence that they saw during finetuning, but performance degrades substantially in other settings, showing a lack of systematic generalization. These results suggest that models often learn to manipulate sentences at a surface level unrelated to the constituent-level syntactic structure, for example by copying the first word of a sentence. These results may partially explain the brittleness of pretrained language models in downstream tasks [1].

1 Introduction

The study of syntax revolves around understanding how words and phrases are combined to form sentences. Certain groups of words, known as *constituents*, behave as units in a sentence. In linguistics, groups of words that form constituents are often identified via *constituency tests* (see, e.g., Haegeman, 1994). The tests involve transforming an input sentence using operations that substitute, displace, or otherwise modify constituents. For example, one well-known constituency test is *pro-form substitution* whereby a constituent is replaced by a corresponding pronominal form: *John gave the book about syntax to the student*⟶*John gave it to the student*. The fact that the phrase *the book about syntax* can be replaced by *it* as a single unit, indicates that it is a constituent. Other examples of constituency tests include clefting and wh-question formation.

Previous work has shown that pretrained transformer-based (Vaswani et al., 2017; Liu et al., 2019b) language models (LMs), trained on large amounts of text achieve unparalleled performance on virtually every downstream task. Nevertheless, these models suffer from robustness issues which call into question their reliability and ability to recover human-like linguistic generalizations. For example, in natural language inference, it has been shown that models rely on superficial cues such as lexical overlap between the premise and the hypothesis in order to make the correct predictions (McCoy et al., 2019; Nie et al., 2019).

A line of work has thus attempted to probe pretrained LMs by training classifiers on top of frozen LM weights in order to extract some desired linguistic representation such as syntax trees (Hewitt and Manning, 2019; Coenen et al., 2019). Work by Prasad et al. (2019) also shows that LSTM based LMs contain information about relative clauses in an interpretable manner. However, such analyses have several limitations. First, it is difficult to separate the contributions of the pre-trained LM weights from that of the probing classifier. In practice, it is necessary to place some constraints on the classifier (e.g., they must be linear) to ensure that extractive performance can be attributed to the pre-trained LMs. More fundamentally, understanding what can be extracted from *pre-trained* representations is a different issue from the NLP system designer's ultimate concern, which is whether a *fine-tuned* system will perform well on new data which involves novel combinations of the units it has seen during pre-training and finetuning; that is, its ability to generalize systematically (Tamkin et al., 2020).

In this paper, we propose analyzing the syntactic

[1] Code is available at https://github.com/aishikchakraborty/constituency

Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 357–366
December 7, 2023. ©2023 Association for Computational Linguistics

competence of pre-trained *and fine-tuned* LMs. We focus on the phenomenon of constituency because of its core role in supporting semantic understanding and natural language inference, with applications to a wide variety of downstream tasks in NLU and NLG.

We ask whether LMs after fine-tuning are able to perform transformations on input sentences which correspond to well-known constituency tests from the literature. Performing these transformations correctly requires that models represent the constituency structure of sentences. In our experiments, we systematically vary two dimensions of generalization: i) whether input sentences represent novel syntactic constructions and ii) whether the main verb of the sentence is novel. We also pose as a control a version of the input sentence where constituents of varying lengths are replaced by their head word.

Our results show that models are able to correctly transform input sentences only when tested on verbs and syntactic constructions that they were trained on. These results indicate that it is unlikely that these models have acquired a human-like representation of constituent structure, instead suggesting that they instead leverage surface-level cues.

2 Background

Constituency Tests. Constituency tests on sentences are a well-known tool from linguistics (see, e.g., Haegeman, 1994) for identifying groups of words that behave as units in a sentence. The idea of building grammar in terms of constituent structure is old (e.g., Wells, 1947) and has been at the heart of formal models of generative grammar since the 1950s (e.g., Chomsky, 1979). Constituents remain of interest as they are a fundamental building block of most modern approaches to grammars and are an important part of most theories of form-meaning mapping.

Extractive Probing. A thread of research focuses on the construction of *probes* to study the representations of pre-trained language models (Conneau et al., 2018). Extractive probes do so by attempting to extract linguistically interpretable structures. Probing word representations like GloVe (Pennington et al., 2014) for linguistic properties have been proposed by Köhn (2015). These early probing methods, that use linear functions as probing functions, have been further used to understand intermediate representations of deep neural networks (Shi et al., 2016; Ettinger et al., 2016; Veldhoen et al., 2016). Since then, probing methods have been used to study questions such as whether neural representations capture information about linguistic structure such as verb tense, part-of-speech, or syntactic dependency type (e.g., Liu et al., 2019a; Hewitt and Manning, 2019).

As pointed out by Hewitt and Liang (2019), an important limitation with existing probing tasks is that they fail to distinguish between information present in probed representations and information that comes from the probe supervision signal. Thus, distinguishing between decoding and learning the probing task is essential.

Understanding Language Model Behavior through Syntactic Tests. Mueller et al. (2022) show that pretraining of language models can induce some specific forms of hierarchical generalization. The authors create linguistic tasks using different sentence transformations, such as, question formation and passivisation. They show that language models can exhibit syntactic generalization on pretraining. McCoy et al. (2020) show that inducing syntactic structures in model architectures is essential for exhibiting syntactic generalization capabilities similar to human beings. Lake and Baroni (2018) show the zero-shot compositional generalization capabilities of several sequence-to-sequence models on a specialized dataset called SCAN. The authors demonstrate that these sequence-to-sequence models posses very limited capabilities to generalize compositionally in the absence of surface-level cues that can be exploited.

In our work, we use sequence-to-sequence models to probe for constituency in pretrained language models.

3 Syntactic Constructions

Our interest in this work is to probe the behaviour of a pre-trained language model M after it is fine-tuned to transform input sentences according to a number of *constituency-sensitive transformations* (CST) which are inspired by constituent tests from the literature. In this section, we describe the set of syntactic constructions which we use to construct our training and test datasets, as well as introduce a notation which allows us to compactly specify CSTs with respect to these constructions.

We start with the set of simple declarative *base sentences* presented in Kann et al. (2018) (D_{base}).

Each of these sentences is constructed using a main verb which is either an English dative or locative alternator (Levin and Rappaport Hovav, 2005; Levin, 1993), which are well-understood syntactic alterations in the literature. Each sentence S can appear as one of four types, which we denote using the feature A with the following notation $S[A] \in \{\text{DO}, \text{PO}, \textsc{Loc}, \textsc{In}\}$.

Dative Sentences using dative main verbs may either be in the double-object ($S[A] = \text{DO}$) or the prepositional object construction ($S[A] = \text{PO}$):

(1) Michael passed the people across the table the salt. ($S_1[A] = \text{DO}$)

(2) Michael passed the salt to the people across the table. ($S_2[A] = \text{PO}$)

Locative Sentences using locative main verbs may be in the locative ($S[A] = \textsc{Loc}$) or instrumental form ($S[A] = \textsc{In}$) constructions:

(3) John sprayed the paint onto the wall. ($S_3[A] = \textsc{Loc}$)

(4) John sprayed the wall with paint. ($S_4[A] = \textsc{In}$)

Our corpus also includes more complex sentences that result from modifying one of the arguments of each verb using a transformation inspired by a constituency test from the literature.

Pronominalization (P) Proform substitution involves replacing one of the verbal arguments with an appropriate proform (he, it, them, etc.). We will use the feature P to indicate whether a sentence contains a noun phrase that is pronominalized, and if so, which syntactic position is pronominalized.[2] For example, (2) has $S_2[P] = \textsc{None}$, in contrast to:

(5) Michael passed *it* to the people across the table. ($S_5[P] = \textsc{dobj}$)

Clefting (C) Clefting is a syntactic construction that involves displacing a verb argument X... into a copular structure "It was X that ...". We use the feature C to indicate whether a constituent has been clefted, and its original syntactic position. So, (2) has $S_2[C] = \textsc{None}$, whereas:

(6) It was *Michael* that passed the salt to the people across the table. ($S_6[C] = \textsc{subj}$)

Wh-question (W) Wh-question formation involves replacing a verb argument with a corresponding *wh*-phrase, displacing it to the beginning of the sentence, and adding appropriate *do*-support. We use the feature W to indicate the formation of a wh-question from a declarative sentence. For example, (2) has $S_2[W] = \textsc{none}$, whereas:

(7) What did Michael pass to the people across the table? ($S_7[W] = \text{DO}$)

Putting this all together, the syntactic form of any sentence in our training or test sets can be described by specifying values for our four features A, C, P, and W. For example, the complete featural description of (2) would be:

$$S_2 = \{A: \text{PO},\ P: \textsc{None},\\ C: \textsc{None},\ W: \textsc{None}\}$$

Informally, a *constituency-sensitive transformation* T can be can be thought of as function $S_{out} = T(S_{in})$ such that T substitutes, displaces, or otherwise modifies one or more constituents in a grammatical input sentence S_{in} in such a way that grammaticality is maintained. In terms of the featural representation we have just introduced, a CST T can be described by (re)assigning a value VAL for feature F in the description of some sentence. We will denote this (re)assignment as $t_{F\leftarrow\text{VAL}}$. For example, a CST which pronominalizes the direct object of a sentence would be written $t_{P\leftarrow\text{DO}}(S) : S[P] \leftarrow \text{DO}$. The sentence that results from applying this transformation to input S_2 that is, $t_{P\leftarrow\text{DO}}(S_2)$, would then be (5).

4 Evaluation Framework

In this section, we describe the framework we use to evaluate our fine-tuned models. Our goal is to test the degree to which our fine-tuned models have captured a notion of constituency. If a model has represented CSTs in terms of the abstract constituent structure of input and output sentences, it should generalize easily to novel words, novel combinations of words, and novel combinations of CSTs. On the other hand, if it is instead relying on low-level cues, it may successfully learn the mappings involved in particular cases of a CST, but not be able to generalize across these dimensions. We test models' constituent structure by systematically varying the amount of generalization we demand in different test conditions. In this section, we describe this evaluation framework.

[2] In this work, we ignore the case where multiple arguments are pronominalized. This case can be easily handled by an extension of our notation to allow sets as features.

In our setting, a training or test *item* is a triple (T, S_{in}, S_{out}) where T is a CST as described in the preceding section and S_{in} and S_{out} are sentences such that $T(S_{in}) = S_{out}$. Note that in our framework, all three parts of the triple are observed during training—the models observe which CST characterizes the relationship between S_{in} and S_{out}.

Test and training datasets are disjoint sets of such items. A model M will be finetuned on $\mathcal{D}^{\text{train}}$ and tested according to its performance on $\mathcal{D}^{\text{test}}$.

Recall that all of the sentences in our dataset are built using main verbs which are either dative or locative alternators. Thus, for all sentences, the value of the base argument structure feature A is one of DO, PO, LOC, or IN.

The high-level idea of our experiments is to see if the models can correct apply a *target CST* to sentences which vary in terms of their similarity to training sentences. In each evaluation we choose a single CST called the *target CST*. In practice, we only evaluate CSTs which correspond to dative or locative alternations (i.e., we only test the following CSTs $A \leftarrow$ PO, $A \leftarrow$ DO, $A \leftarrow$ INS, $A \leftarrow$ LOC). For the sake of concreteness, our description of training data construction below will use the target $A \leftarrow$ PO, but the other targets are handled analogously.

For this target transformation, define the set of base sentences

$$\begin{aligned}\mathcal{S}^{\text{base}} = &\{S|S[A] = \text{DO},\\ &S[P] = S[C] = S[W] = \text{NONE}\};\end{aligned}$$

These are simple declarative sentences in the double object construction without any further pronominalization, clefting, or wh-question formation applied.

$\mathcal{D}^{\text{train}}$ will consist of the union of following sets of triples (T, s_{in}, s_{out}):

I. Non-base transformations applied to each $S \in \mathcal{S}^{\text{base}}$:

$$\begin{gathered}(T_{C\leftarrow *}, S, T_{C\leftarrow *}(S)),\\ (T_{P\leftarrow *}, S, T_{P\leftarrow *}(S)),\\ (T_{W\leftarrow *}, S, T_{W\leftarrow *}(S)),\end{gathered}$$

where * denotes all possible non-null values for that feature. That is, training includes all non-base transformation applied to the double object base sentences.

II. The target transformation applied to the base sentences:

$$(T_{A\leftarrow PO}, S, T_{A\leftarrow PO}(S)).$$

Thus, training also includes the prepositional object alternation applied to double object base sentences.

III. The target verb alternation, applied in the opposite direction (i.e., A ← DO), to a sentence which is in one of the more complex construction forms (pronominalized, clefted, wh-item). Let S' be the base sentence S transformed to be in the PO construction $S' = T_{A\leftarrow PO}(S)$.

$$\begin{gathered}(T_{A\leftarrow DO}, T_{C\leftarrow *}(S'), T_{A\leftarrow DO}(T_{C\leftarrow *}(S'))),\\ (T_{A\leftarrow DO}, T_{P\leftarrow *}(S'), T_{A\leftarrow DO}(T_{P\leftarrow *}(S'))),\\ (T_{A\leftarrow DO}, T_{W\leftarrow *}(S'), T_{A\leftarrow DO}(T_{W\leftarrow *}(S')))\end{gathered}$$

Test At test time, the model will be evaluated on its performance on the set $\mathcal{D}^{\text{test}}$, which consists of the target transformation applied to the output of I.:

$$\begin{gathered}(T_{A\leftarrow PO}, T_{C\leftarrow *}(S), T_{A\leftarrow PO}(T_{C\leftarrow *}(S))),\\ (T_{A\leftarrow PO}, T_{P\leftarrow *}(S), T_{A\leftarrow PO}(T_{P\leftarrow *}(S))),\\ (T_{A\leftarrow PO}, T_{W\leftarrow *}(S), T_{A\leftarrow PO}(T_{W\leftarrow *}(S)))\end{gathered}$$

Notably, the model is trained on each of the transformations involved in generating the test set ($T_{A\leftarrow PO}$, $T_{C\leftarrow *}$, $T_{P\leftarrow *}$, $T_{W\leftarrow *}$), as well as on all construction types. Our test sets vary (i) whether particular combinations of CST and input sentence type are held out (ii) whether particular verbs and arguments are held out and (iii) whether constituents vary in length between test and train.

4.1 Evaluation Dimension 1: Novel Combinations of CST and Input

As our first dimension of evaluation, we vary whether the test items described in Section 4 are included in the training set or not. We call these conditions NOVEL-CST-INPUT and OBSERVED-CST-INPUT. Note, that in NOVEL-CST-INPUT, the model will have never seen the particular combination of target CST and input item construction type in the training data. A model which is able to generalize in this condition must be able to correctly identify the arguments of the main verb despite the fact that these argument appear pronominalized, clefted, or as what-items, and then apply the corresponding target transformation. For instance, the model might be presented with the sentence *what did the painter spray the wall with* and have to correctly identify *what* as the instrumental argument of this instrumental construction, and transform the sentence to the locative construction *what did the painter spray onto the wall*. This is a very challenging task.

4.2 Evaluation Dimension 2: Novel Verbs and Arguments

As our second dimension of evaluation, we vary whether the test items use completely novel verbs and arguments (NOVEL-WORDS) or whether they reuse verbs and arguments observed during test (OBSERVED-WORDS). Note that the verbs and argument constituents are novel with respect to the fine-tuning task; they appear however in pre-training.

As mentioned before, our base transformations come from the dataset proposed by Kann et al. (2018). In this dataset, the test set verbs are disjoint from both the training and development set verbs. This helps us test for novel verbs and arguments.

The tests for the case where the test set verbs are non-novel are done by creating the *non-novel verb test set*. We describe this in Section 6.2.

4.3 Evaluation Dimension 3: Generalization across Constituent Lengths

A critical property of constituents is that they are sets of words of varying size that behave as single units. Thus, as our third dimension of evaluation, we introduce a baseline condition where all constituents are replaced by their head word HEAD-WORD-ORACLE (HWO) resulting in a corpus where constituents can always be identified with single words.

5 Model

5.1 Model Architecture

We evaluate two pretrained sequence-to-sequence transformer models BART (Lewis et al., 2019) and T5 (Raffel et al., 2020). We utilize the BART-base checkpoint to initialize the BART model. For T5, we utilize the t5-base checkpoint for initialization.

Input Embedding Details The inputs to the encoder are transformed into the embedding space by using the input embeddings of pretrained BART (or T5). Similar to the original implementation, we use *0* as the decoder start token for BART and the *PAD* token as the decoder start token for the T5 model. To inform the model of what kind of transformation we want to get, we append a special token TRANSFORMATION: *uid* at the start of every input. The *uid* is a unique identifier corresponding to a transformation $t \in T$. We always use greedy search for decoding purposes.

The model is trained using a standard next-word cross-entropy loss function.

5.2 Building the Head Word Oracle

We introduce a *head word oracle (HWO)* model that controls for the effect of the varying constituent lengths. This HWO identifies the head of each noun phrase constituent in our dataset using a dependency parse of the sentences. We then replaced each noun phrase in the dataset with its head word piece as a pre-processing step. This procedure standardizes the lengths of constituents, and simplifies the problem that the model must solve. The head words are identified using a dependency parser on the original sentence. After the model does the necessary transformation task, we transform the head words into their original constituents and evaluate the model using the original constituents. In case such a transformation is not possible due to incorrect outputs, we keep the head words as is.

6 Experimental Setup

6.1 Evaluation Metrics

We adopt the following three evaluation measures of the similarity between the predicted output and the reference sentence.

Edit Distance We make use of a standard (i.e., Levenshtein) edit distance between predicted and gold standard output sentences.

BLEU BLEU is a widely used automatic evaluation metric from machine translation that considers N-gram overlap with a brevity penalty (Papineni et al., 2002).

METEOR Meteor (Banerjee and Lavie, 2005) is an automatic evaluation metric that measures how well a system adds, deletes or preserves words. This metric is a standard measure for evaluating several language generation systems.

6.2 Datasets

Base Sentences We adopt the dataset of Kann et al. (2018) as the base upon which we build more complex sentences. This subset of sentences consists of simple declarative sentences using dative and locative (spray/load) alternator verbs. The original dataset provided by Kann et al. (2018) (D_{base}) contains both grammatical and ungrammatical sentences, we remove the latter for the purposes of our study.

Proform Substitution: All sentences in our base dataset make use of dative or locative verbs and thus have three verbal arguments: in the case of dative verbs, a subject, object, and indirect object or oblique; in the case of locative verbs a subject, and two oblique arguments. Sentences with a proform substitution set replace one of these three arguments with an appropriate pronominal form such as he, she, they, or it. Thus S[P] can take on values DOBJ, INOBJ and SUBJ.

Clefting: We generate clefted sentences in a similar way to the pro-form substitutions, targeting one of the three arguments for extraction. Thus the S[C] can take on values DOBJ, INOBJ and SUBJ.

***wh*-questions:** We generate wh-questions by targeting one of the three arguments in each base sentence for extraction. Thus, S[W] can take on values DOBJ, INOBJ and SUBJ.

Creating the NOVEL-WORDS test set and every train, val split: We use the training, validation and test sets from the base corpus D_{base}. We apply the relevant syntactic transformations to create the experiments described in Section 4 by using the clefting, proform and wh sentence generation strategy discussed above. The final corpus statistics are shown in Table 1.

Creating the OBSERVED-WORDS test set: This test set is made by using the base transformation dataset already available to us. The main property of this new test set is that the main verb in the test set is seen during training. We randomly chose 30 sentences from the training corpus. We create the new test set the by replacing the direct object and the prepositional objects of the randomly chosen training sentences with new objects. These objects can appear during training and must make the final sentence a grammatical sentence.

Dataset Split	Dative	Locative
Train NOVEL-CST-INPUT	4,268	5,668
Train OBSERVED-CST-INPUT	8,208	8,208
Validation	153	612
Test NOVEL-WORDS	225	585
Test OBSERVED-WORDS	90	-

Table 1: Final train, test and validation corpus statistics

6.3 Model Initializations and Hyperparameters

Encoder and Decoder Initializations For all BART models, the encoders and deocders are initialized with the *bart-base* checkpoint. Similarly, for the T5 model, the encoder is initialized with *t5-base*. Note that the BART and T5 models have different number of trainable parameters. During evaluation, we do not make any comparisons between the pretrained models. The outputs of the encoder-decoder model are subwords. We use the BART(T5) tokenizer to combine these subwords into words. Finally, during generation, we always use greedy decoding in all our experiments.

Optimization We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $3e-5$ to train all non-head word oracle models and $3e-6$ for the training of HWO models. We use a batch size of 32 and the model is trained for a total of 5 epochs.

For all models, we save the model with the best validation BLEU score and load the model corresponding to the best validation performance during test time.

7 Results

We evaluate the models proposed in Section 5 using the tasks proposed in Section 3. We apply the four base transformations ($T_{A\leftarrow PO}, T_{A\leftarrow DO}, T_{A\leftarrow LOC}, T_{A\leftarrow IN}$) to the three separate non-base transformations (clefting, pronormalization and wh-question generation). We take an average of the results corresponding to the four base transformations.

7.1 Performance of PLMs on Constituency Tasks

Generalization across sentence types and verbs: We look at the effect of holding out the target linguistic composition (NOVEL-CST-INPUT vs OBSERVED-CST-INPUT) on each of the constituency task . In this setup, the verbs in the test set are novel. Tables 2 and 3 shows that the Head Word

Input transformation type	Cleft			Proform			wh		
Models	**ED**	**BLEU**	**METEOR**	**ED**	**BLEU**	**METEOR**	**ED**	**BLEU**	**METEOR**
NOVEL-CST-INPUT and NOVEL-WORDS									
BART HWO	6.27	82.33	80.13	6.00	84.72	87.64	6.00	86.89	86.65
BART	14.91	70.17	70.74	13.30	76.66	75.03	13.00	80.86	80.59
NOVEL-CST-INPUT and OBSERVED-WORDS									
BART HWO	5.93	88.33	86.49	5.22	90.05	89.27	5.02	88.02	87.56
BART	14.82	72.82	72.91	15.72	72.62	72.87	15.98	72.02	72.22
OBSERVED-CST-INPUT and NOVEL-WORDS									
BART HWO	3.54	89.36	88.73	3.11	85.78	84.49	3.62	92.25	92.03
BART	8.19	84.65	84.53	4.88	81.44	80.31	2.32	90.74	91.34
OBSERVED-CST-INPUT and OBSERVED-WORDS									
BART HWO	0.31	98.32	96.53	0.38	98.54	96.09	0.32	98.10	96.73
BART	2.71	97.32	97.29	2.36	97.68	97.91	2.98	97.37	97.27

Table 2: Evaluating the BART models in the four different experimental settings. The results are an average of the results obtained by composing four base transformations with our non-base input transformations (cleft, proform and wh).

Input transformation type	Cleft			Proform			wh		
Models	**ED**	**BLEU**	**METEOR**	**ED**	**BLEU**	**METEOR**	**ED**	**BLEU**	**METEOR**
NOVEL-CST-INPUT and NOVEL-WORDS									
T5 HWO	6.32	86.23	86.97	6.04	86.98	86.39	6.98	86.09	86.80
T5	8.30	80.71	80.52	8.32	80.82	81.71	8.91	80.11	81.02
NOVEL-CST-INPUT and OBSERVED-WORDS									
T5 HWO	5.32	89.11	89.72	5.27	89.88	89.80	5.08	89.25	89.07
T5	8.83	83.61	83.70	7.91	86.82	86.42	7.63	86.39	86.99
OBSERVED-CST-INPUT and NOVEL-WORDS									
T5 HWO	3.01	97.82	97.37	3.02	97.79	97.34	3.91	96.92	97.05
T5	3.31	95.32	96.31	3.04	95.24	96.32	3.10	95.21	96.15
OBSERVED-CST-INPUT and OBSERVED-WORDS									
T5 HWO	2.71	98.66	98.59	2.83	98.30	98.54	2.85	98.75	98.02
T5	2.99	96.98	97.16	2.90	96.04	96.81	2.95	96.69	96.91

Table 3: Evaluating the T5 models in the four different experimental settings. The results are an average of the results obtained by composing four base transformations with our non-base input transformations (cleft, proform and wh).

Source Sentence	**Target Sentence**	**BART output**	**T5 output**
It was a plate of food that john gave to the little boy.	It was a plate of food that John gave the little boy.	It was a plate that John gave to the little boy.	It was a plate that John gave to the little boy to.
Michael gave a plate of food to them	Michael gave them a plate of food	Michael gave them them a plate to	Michael gave them to plate

Table 4: Common BART and T5 error cases while dealing with quantitative constructions.

Oracle models have superior performances over the *BART* and the *T5* model in the NOVEL-CST-INPUT setting. This shows that identifying constituency boundaries is a difficult task for the non-oracle models. The Head Word Oracle models need to learn which tokens need to be rearranged, substituted, or deleted, but they do not have to learn to group words into constituents. This is unlike the *BART* and the *T5* models that need to identify constituent boundaries and do rearragements, substitutions and deletions on those extracted constituents.

Similar trends are seen when the sentence types corresponding to the target linguistic composition are not withheld. Here, BART and T5 Oracle models, as well as the *BART* and the *T5* models in the non-held out setting outperform their counterparts in the held-out setting. This does indicate that the models do not excel at extracting and utilizing the constituency-level information even when the constituents are reduced to single tokens.

Generalization across sentence types: In Tables 2 and 3, we observe that compositions involving the non-base transformations clefting generation results in worse performance than pronormalization and wh question generation. We investigated this issue and found this to be a side effect of overfitting. During training, the model always learns to copy the first token. Thus, it fails to learn the fact that during clefting,it needs to generate new tokens.

Generalization across verbs: The performance of the *BART* and the *T5* model is significantly better when the target verbs and arguments are seen during training in the OBSERVED-WORDS experimental setting. This suggests that the PLMs do not learn verbal subcategorization frames which are important for deriving and manipulating sentential argument structure. Instead, they seem to rely on surface-level cues to make predictions, which is why the models when tested on non-novel verbs outperform the model tested on novel verbs.

7.2 Quantitative Constructions

We looked at the outputs of *BART* and *T5* models when the test set verbs are unseen and the target transformation is held-out. A common error among these models is that a lot of the time they end up copying the inputs without the necessary transformations. We also noticed that both models make errors consistently when the input sentence has a quantitative construction (a.k.a. pseudopartitives, e.g. *a plate of food*), as can be seen in Table 4. The models correctly rearrange the order of the nominal arguments. However, they have difficulty identifying the precise constituent boundaries, resulting in errors. This further illustrates why the Head Word Oracle model ends up having superior performance. In fact, on average, the BART Head Word Oracle correctly transforms 90.7% of the sentences containing quantitative constructions, as opposed to the BART model, which never transforms any such sentences correctly.

8 Conclusion

In this paper, we systematically vary the task setup and the training signals to do a behavioral analysis of pretrained sequence-to-sequence models. We design several linguistic tests including verb argument structure alternations, proform substitution, clefting and wh-question generation. We show that the models fail to generalize well when the target transformation is held-out. We attribute this to the failure of the pretrained language models in utilizing constituency information and relying on surface-level cues. We further show that simplifying the constituent boundaries improves the generalization capabilities of these models. Furthermore, increasing the number of out-of-vocabulary tokens in the test corpus decreases the generalization performance of these models.

9 Limitations

Throughout this paper, we use specialized datasets for analyzing the behavior of various pretrained language models. The datasets we use for creating the constituency tests are in English which has relatively fixed word order. One feature of the sentences in the base constructions like dative is that the first token is always a subject named entity in the base sentences. This makes it easy for the model we use to learn certain biases. For example, the first token can be copied when we apply a transformation to change the verb alteration. In languages with relatively free word order, this might create an issue for these models to do some of the basic transformations correctly as the syntactic patterns might be too complex to learn. This could make our current models including our oracle models not very effective for doing similar analyses. In addition, the current suite of constituency tests we use may not work on languages with different word orders.

10 Acknowledgements

We would like to thank the reviewers for their valuable comments. This work is supported by funding from Samsung Electronics. This research was enabled in part by support provided by Calcul Québec [3], and Compute Canada [4].

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Pro-*

[3] https://www.calculquebec.ca

[4] https://www.computecanada.ca

ceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Noam Chomsky. 1979. The logical structure of linguistic theory. *Synthese*, 40(2):317–352.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139.

L. Haegeman. 1994. *Introduction to Government and Binding Theory*. Blackwell Textbooks in Linguistics. Wiley.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743.

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.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R Bowman. 2018. Verb argument structure alternations in word and sentence embeddings. *arXiv preprint arXiv:1811.10773*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Arne Köhn. 2015. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *35th International Conference on Machine Learning, ICML 2018*, 35th International Conference on Machine Learning, ICML 2018, pages 4487–4499. International Machine Learning Society (IMLS). Publisher Copyright: © Copyright 2018 by the author(s).; 35th International Conference on Machine Learning, ICML 2018 ; Conference date: 10-07-2018 Through 15-07-2018.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press.

Beth Levin and Malka Rappaport Hovav. 2005. *Argument realization: Research surveys in linguistics*. Cambridge University Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019a. Linguistic knowledge and transferability of contextual 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 1073–1094.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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. *arXiv preprint arXiv:1902.01007*.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. Investigating transferability in pretrained language models. *arXiv preprint arXiv:2004.14975*.

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

Sara Veldhoen, Dieuwke Hupkes, and Willem Zuidema. 2016. Diagnostic classifiers: Revealing how neural networks process hierarchical structure. In *Proceedings of the Workshop on Cognitive Computation (CoCo@NIPS2016)*.

Rulon S. Wells. 1947. Immediate constituents. *Language*, 23(2):81–117.