Filtered Corpus Training (FiCT) Shows that Language Models Can Generalize from Indirect Evidence

Abhinav Patil[▲,*] and **Jaap Jumelet**[🚲,*]
Yu Ying Chiu[▲] and **Andy Lapastora**[☺,†] and **Peter Shen**[▲] and
Lexie Wang[▲] and **Clevis Willrich**[π,†]
Shane Steinert-Threlkeld[▲]

▲University of Washington, Seattle, WA, USA
🚲ILLC, University of Amsterdam, Amsterdam, Netherlands
☺AWS AI Labs, Amazon, Seattle, WA, USA
πMicrosoft, Redmond, WA, USA

Abstract

This paper introduces **Fi**ltered **C**orpus **T**raining, a method that trains language models (LMs) on corpora with certain linguistic constructions filtered out from the training data, and uses it to measure the ability of LMs to perform linguistic generalization on the basis of indirect evidence. We apply the method to both LSTM and Transformer LMs (of roughly comparable size), developing filtered corpora that target a wide range of linguistic phenomena. Our results show that while transformers are better qua LMs (as measured by perplexity), both models perform equally and surprisingly well on linguistic generalization measures, suggesting that they are capable of generalizing from indirect evidence.

1 Introduction

Language models (LMs) play an increasingly large role in natural language processing systems and have become capable of producing surprisingly fluent and grammatical text. However, the mechanisms underlying the acquisition and use of such linguistic proficiency remain largely unknown. In particular, the degree that language learning relies on memorization versus generalization remains a topic of investigation (Hupkes et al., 2023). The reliance of LMs on large amounts of training data raises the suspicion that they do not generalize in a ''human-like manner'' (McCoy et al., 2019; Hu et al., 2020; Oh and Schuler, 2023b), but it is hard to address such questions with traditional evaluation metrics such as perplexity.

This paper introduces *Filtered Corpus Training* (FiCT) as a method for measuring the linguistic generalization abilities of language models. As depicted in Figure 1, FiCT involves training models on corpora that have been filtered to remove specific linguistic constructions, thereby testing the models' ability to generalize beyond their training data. For example: We can train a model on a corpus that has never seen subjects modified by a prepositional phrase (e.g., ''A sketch *of lights* {doesn't / *don't}...''), and then ask whether it can judge the grammaticality of such sentences. If a model has learned that verbs must agree with the head noun of the subject noun phrase (NP), and that NPs can be modified by PPs (e.g., from seeing these in object but not subject position), it should be capable of generalizing to the unseen PP-modified subjects.

This method enables us to ask whether models can form relevant linguistic generalizations from *indirect evidence*, or whether they require direct evidence (e.g., examples of constructions during training; Warstadt and Bowman, 2022; Mueller and Linzen, 2023). In essence, by *intervening* on patterns in the training data we obtain a more causal account of the relation between training data and model behavior (Pearl, 2009). Furthermore, by carefully controlling for the number of parameters, we can investigate the inductive biases of two major LM architectures, Transformers and LSTMs, which allows us to give more detailed answers about the recent successes of Transformer models on a fine-grained linguistic level.

We apply the FiCT methodology by developing filters targeting a wide range of the linguistic

*Co-first authors. Full author contribution statement at the end of paper, after acknowledgements. Correspondence: abhinavp@uw.edu, jumeletjaap@gmail.com, shanest@uw.edu.

†Work done while the author was a student at University of Washington.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1597–1615, 2024. https://doi.org/10.1162/tacl_a_00720
Action Editor: Marco Baroni. Submission batch: 5/2024; Revision batch: 8/2024; Published 11/2024.

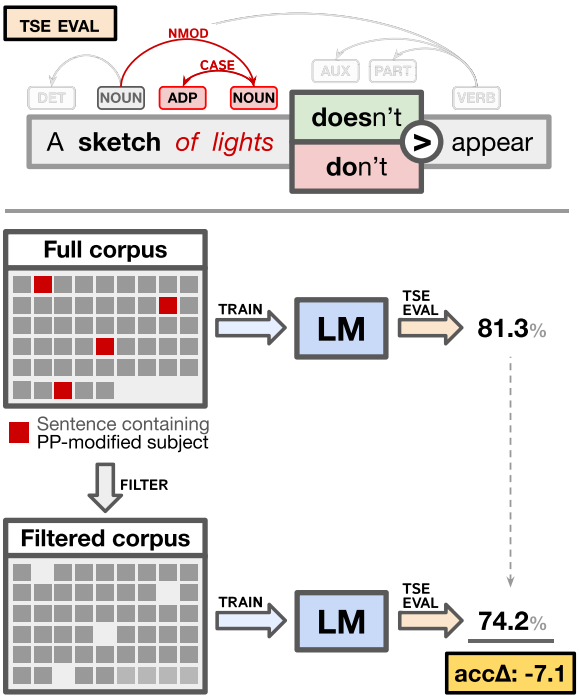

Figure 1: Overview of the **Fi**ltered **C**orpus **T**raining methodology (FiCT). For a linguistic construction of interest (e.g., prepositionally modified subjects), we filter out sentences containing that construction and train a new language model on the filtered corpus. We measure performance on *targeted syntactic evaluations* to assess the capacity of the LM to generalize from related constructions to this novel, unseen construction.

phenomena evaluated by BLiMP (§3; Warstadt et al., 2020) and training both LSTM and Transformer LMs on the resulting corpora (§4). Our results (§5) show that while Transformers are uniformly better qua language models (as measured by perplexity), their linguistic generalization abilities are not better than that of the LSTMs (as measured by a metric we introduce called *accuracy delta*), demonstrating a dissociation between perplexity and linguistic generalization. Furthermore, for both models, the impact of filtered corpus training on grammaticality judgments is quite low, suggesting that language models are able to form sophisticated linguistic generalizations on the basis of only indirect evidence (as discussed in §6).

These results shed light on the debate between memorization and generalization in language models: By causally intervening on the training data, we ensure that models have never seen instances of their evaluation targets. That they can still make correct grammaticality judgments shows they generalize in subtle and linguistically relevant ways that go beyond their training data.

2 Background

2.1 Surprisal Theory

Language modeling performance can be measured using *perplexity*, indicating a model's fit to a corpus distribution. Intuitively, one might expect that lower perplexity leads to more human-like linguistic behavior. This connection has been explored in detail in the context of *surprisal theory* (Hale, 2001; Levy, 2008): Encountering a highly surprising token results in a longer reading time. Initial findings indicate that lower perplexity, as measured by language models, leads to better reading time predictions (Fossum and Levy, 2012; Goodkind and Bicknell, 2018; Wilcox et al., 2020), although affected by model architecture (Hao et al., 2020), cross-lingual effects (Kuribayashi et al., 2021), and syntactic ambiguity (Arehalli et al., 2022). It has been shown, however, that lower perplexity only results in better predictive power up to around 2 billion training tokens (Oh and Schuler, 2023a): After this point LMs become too accurate at predicting low-frequency constructions and long-distance dependencies (Oh et al., 2024). The present paper also explores the connection between perplexity and human-like linguistic behavior and will find a dissociation with perplexity.

2.2 Targeted Syntactic Evaluations

Perplexity should be augmented with other evaluations that specifically target the models' ability to generalize in a human-like way. Such investigations often draw on psycholinguistic paradigms, treating language models as participants in order to learn what such models ''know'' about specific linguistic phenomena (Futrell et al., 2019; Warstadt et al., 2019b; Ettinger, 2020). A common paradigm in this body of literature, usually referred to as ''targeted syntactic evaluations'' (Linzen et al., 2016; Jumelet and Hupkes, 2018; Marvin and Linzen, 2018; Kann et al., 2019; Newman et al., 2021) involves comparing language models' preferences between minimal pairs of sentences: A model is deemed to understand a phenomenon if it assigns a higher probability to the grammatical alternation.

The benchmark suites with the widest coverage over linguistic phenomena are SyntaxGym (Gauthier et al., 2020) and the Benchmark of Linguistic Minimal Pairs (BLiMP, Warstadt et al., 2020), the latter of which we will use in our experiments. BLiMP consists of 67 different benchmarks, each consisting of 1,000 minimal pairs, which target twelve different linguistic areas, broadly construed, across morphology, syntax, semantics, and the syntax-semantics interface. This is the benchmark we use as a primary means of evaluation in the present investigation, discussed in greater detail in §4.

2.3 Linguistic Generalization

While targeted syntactic evaluations give an insight into a model's linguistic competence, it does not show *how* a model acquires this notion of grammaticality. In this paper we focus on two kinds of linguistic generalization. *Structural generalization* (Hupkes et al., 2023) asks: Can language models make grammaticality judgments in syntactically more complex constructions than seen during training? One line of work approaches this question from a fine-tuning perspective: By fine-tuning a model on a particular set of constructions we can measure the impact that this has on other linguistic constructions (Prasad et al., 2019; Weber et al., 2024). *Lexical generalization* asks whether models can generalize a seen construction to new lexical items that it has not seen in that construction (Kim and Linzen, 2020).

In order to gain a *causal* perspective on how the training data influences model performance, we retrain models from scratch on *filtered* corpora. This methodology has been deployed in earlier work to investigate how LMs learn the licensing conditions of negative polarity items from different contexts (Jumelet et al., 2021; Weber et al., 2021). Warstadt (2022) investigates the *poverty of the stimulus* debate through the lens of filtered corpora, focusing on the phenomenon of subject auxiliary inversion. Finally, Misra and Mahowald (2024) investigate rare adjective-noun constructions and manipulate training corpora to investigate how models acquire an understanding of rare constructions. Whereas most of these focus on a particular linguistic construction, our work applies the approach to a wide range of phenomena.

3 Filtered Corpus Training (FiCT)

This section first introduces the logic of the FiCT method before detailing the specific filters that we use in our experiments. The final experimental setup is described in §4. Code and data, as well as a link to all models on the HuggingFace Hub, can be found at `https://github.com/CLMBRs/corpus-filtering`.

3.1 Logic of the Method

The core methodological basis of this paper is what we call *Filtered Corpus Training*, or FiCT. This involves comparing the performance of otherwise identical learners that are trained on data which differs in some interesting way.

In this paper, the FiCT methodology is primarily used to test whether LMs are capable of extrapolating linguistic rules learned from environments in training data to unseen environments. In order to ensure that the specified environments are not seen in the training data, we use *filters* to remove sentences with the specified environments from a naturalistic corpus. By comparing models trained on the ablated data and models trained on the full, naturalistic corpus, we can potentially determine whether, how, and when language models are able to make such generalizations.

Figure 1 illustrates the logic of our method. The sentence pair ''A sketch of lights {doesn't / *don't} appear'' contains a subject with a prepositional phrase (PP) modifying a noun, itself with a noun that differs in number from the main subject. We filter from the training corpus all sentences with subjects containing PP modifiers, and then compare the ability to make the correct grammaticality judgments on this pair between a model trained on the full corpus and this filtered corpus. This difference in performance we call $\text{acc}\Delta$ (formally defined in §4). A model that has not seen PP-modified subjects could still make the correct judgments by forming the following generalizations: Verbs agree with the head noun of the subject, and noun phrases with PP modifiers (which can be seen in object, but not subject position) are headed by the main noun. Low $\text{acc}\Delta$ would then provide evidence that the model has developed such generalizations.

The filters used in the present investigation are listed in Table 1, along with the BLiMP benchmark(s) each targets, and some descriptive summary statistics for each. These filters

Corpus name	BLiMP benchmark	Example	%BLiMP items targeted	%sentences filtered out	#Tokens as % of full
FULL	–	–	–	0.00	100.0
AGR-PP-MOD	distractor_agr_relational_noun	*A sketch of lights doesn't/*don't appear*	99.5	18.50	95.80
AGR-REL-CL	distractor_agr_relative_clause	*Boys that aren't disturbing Natalie suffer/*suffers.*	94.4	2.76	98.99
AGR-RE-IRR-SV	irregular_plural_subject_verb_agr_1	*This goose isn't/*weren't bothering Edward.*	99.4	11.29	98.59
	irregular_plural_subject_verb_agr_2	*The woman/*women cleans every public park.*	97.2		
	regular_plural_subject_verb_agr_1	*Jeffrey hasn't/*haven't criticized Donald.*	99.3		
	regular_plural_subject_verb_agr_2	*The dress/*dresses crumples.*	99.1		
NPI-ONLY	only_npi_licensor_present	*Only/*Even Bill would ever complain.*	100	0.09	99.93
	only_npi_scope	*Only those doctors who Karla respects ever . . . / *Those doctors who only Karla respects ever ...*	100		
NPI-SENT-NEG	sentential_negation_npi_licensor_present	*Those banks had not/*really ever lied.*	100	0.45	99.82
	sentential_negation_npi_scope	*The turtles that are boring me could not ever . . . / *The turtles that are not boring me could ever ...*	100		
NPI-SIM-QUES	matrix_question_npi_licensor_present	*Should I ever join? / *I should ever join.*	100	0.01	99.98
QUANTIFIER-SUPERLATIVE	superlative_quantifiers_1	*No man has revealed more than/*at least 5 forks.*	98.5	7.29	97.72
	superlative_quantifiers_2	*An/*No actor arrived at at most 6 lakes.*	99.3		
QUANTIFIER-EXISTENTIAL-THERE	existential_there_quantifiers_1	*There aren't many/*all lights darkening.*	99.1	1.15	99.82
BINDING-C-COMMAND	principle_A_c_command	*A lot of actresses that thought about Alice healed themselves/*herself.*	96.6	0.01	100.0
BINDING-CASE	principle_A_case_1	*Tara thinks that she/*herself sounded like Wayne.*	100	1.54	99.54
	principle_A_case_2	*Anna imagines herself praising/*praises this boy.*	92.5		
BINDING-DOMAIN	principle_A_domain_1	*Carlos said that Lori helped him/*himself.*	100	0.44	99.84
	principle_A_domain_2	*Mark imagines Erin might admire herself/*himself.*	99.3		
	principle_A_domain_3	*Nancy could say every guy hides himself. / *Every guy could say Nancy hides himself.*	99.5		
BINDING-RECONSTRUCTION	principle_A_reconstruction	*It's herself who Karen criticized / *criticized Karen.*	99.1	0.01	99.99
PASSIVE	passive_1	*Jeffrey's sons are insulted/*smiled by Tina.*	96.9	2.67	99.57
	passive_2	*Most cashiers are disliked/*flirted.*	98.9		
DET-ADJ-NOUN	det_noun_agr_with_adj_1	*Tracy praises those lucky guys/*guy.*	95.6	1.14	99.78
	det_noun_agr_with_adj_2	*Some actors buy these/*this gray books.*	93.0		
	det_noun_agr_with_adj_irregular_1	*He shouldn't criticize this upset child/*children.*	92.0		
	det_noun_agr_with_adj_irregular_2	*That adult has brought that/*those purple octopus.*	93.9		
DET-NOUN	det_noun_agr_1	*Craig explored that grocery store/*stores.*	99.7	0.47	99.95
	det_noun_agr_2	*Carl cures those/*that horses.*	99.8		
	det_noun_agr_irregular_1	*Phillip was lifting this mouse/*mice.*	100		
	det_noun_agr_irregular_2	*Those ladies walk through those/*that oases.*	100		

Table 1: An overview of all the filters, the BLiMP benchmark they target, an example for each benchmark, and number of items targeted by the filter. The rightmost column represents the relative number of tokens in each filtered corpus after they have been downsampled to the same number of lines.

utilized part-of-speech, morphological features, and syntactic dependency annotations generated via the use of Stanza (Qi et al., 2020), an off-the-shelf package that uses pretrained neural models to generate grammatical annotations within the framework of Universal Dependencies (UD) (Nivre et al., 2017, 2020). We now describe the filters in more detail.

3.2 Corpus Filters

In general, we favor ''stronger'' filters, i.e., those that include false positives (and so filter out more training data), since our goal is to ensure that the LM has not seen a given construction during training. In what follows, $x >_z y$ means that there is a dependency from x to y with label z.

3.2.1 Structural Generalization

In the following filters, a particular structural configuration has been completely removed from the corpus, and a model must generalize to it from similar/related configurations.

AGR-PP-MOD The benchmark targeted by this filter tests subject-verb number agreement in the presence of an intervening *distractor* in a prepositional phrase, as illustrated in Figure 1. AGR-PP-MOD filters all sentences containing the dependency structure VERB $>_{\text{nsubj}}$ NOUN $>_{\text{nmod}}$ NOUN $>_{\text{case}}$ ADP. The resulting filtered corpus will still contain PPs modifying nouns in other contexts (e.g., object position). If a learner has formed a general 'rule' for subject-verb agreement, and seen PP-modified objects, it should be able to generalize to agreement

with PP-modified subjects, even when it hasn't seen them during training.

AGR-REL-CL This filter is similar to the previous one, but targets sentences where the distractor occurs in a relative clause in subject position, removing all sentences containing the structure VERB $>_{\text{nsubj}}$ NOUN $>_{\text{acl:relcl}}$ ADJ, e.g., ''The boys that aren't disturbing Natalie dream''. A model might generalize again from its general 'rule' for subject-verb agreement, and learn about relative clause structure from relative clauses in object position.

NPI-Filters We use the list of negative polarity items (NPIs) provided by Jumelet et al. (2021) and filter as follows: NPI-ONLY removes all sentences with an NPI occurring after 'only' (e.g., ''Only students have ever complained about morning classes''), NPI-SENT-NEG removes sentences with a negation and an NPI, and NPI-SIM-QUES removes questions with NPIs in them. In each of these cases the model can generalize NPI licensing conditions for a particular environment from other environments that are still present.

QUANTIFIER-SUPERLATIVE Superlative quantifiers (e.g., *at least*, *at most*) cannot be embedded under negation: *An actor arrived at at most six lakes* vs. **No actor arrived at at most six lakes*. BLiMP targets this phenomenon in two ways: either by replacing the superlative quantifier under negation with a relative quantifier (e.g., *more than 5*), or by removing the negation. We cannot detect superlative quantifiers based on dependency information alone, so we use morphological feature annotations. Next, we filter all such constructions that appear in object position: VERB $>_{\text{obl/obj/iobj}}$ NOUN $> \cdots >$ QUANTIFIER. It is less clear for this filter how a model can still infer the grammaticality from other constructions that are not covered by the filter.

QUANTIFIER-EXISTENTIAL-THERE *Weak* quantifiers can occur in the scope of existential *there* constructions, whereas *strong* quantifiers cannot: *There are many people here* vs. **There are all people here* (Milsark, 1974). BLiMP targets this phenomenon in two ways: either replacing a weak quantifier with a strong one, or increasing the scope of a locative *there* such that it becomes existential. We filter all weak quantifiers occurring in subject position under an existential *there*: THERE $<_{\text{expl}}$ ARE $>_{\text{nsubj}}$ NOUN $>$ WEAK-Q. However, we only filter the 5 weak quantifiers occurring in the BLiMP benchmark (*a(n), no, some, few, many*), which still allows a model to generalize from other weak quantifiers to infer the grammaticality conditions. Furthermore, weak vs. strong quantification plays a role in other linguistic phenomena as well, a fact which a learner could leverage.

BINDING-Filters Four filters, BINDING-C-COMMAND, BINDING-CASE, BINDING-DOMAIN, and BINDING-RECONSTRUCTION target the seven binding-related benchmarks of BLiMP. All seven benchmarks typify various facets of Chomsky's (1993) Principle A. The implementations of all four filters is generally similar: They target sentences where a reflexive or non-reflexive pronoun occurs in the specific context(s) illustrated by the corresponding benchmarks, narrowly construed, while leaving in sentences where the same or similar principle is applied in a different environment. For example, the BINDING-C-COMMAND filter removes evidence of the use of the c-command relationship in anaphora licensing *in relative clauses*, but not elsewhere, as in sentences like *Mary's brother hurt himself* (but not **Mary's brother hurt herself*).[1] The other three benchmarks operate in similar ways.

DET-ADJ-NOUN One of the filters targeting determiner-noun agreement focuses on cases where an adjective occurs between a demonstrative determiner and a noun, e.g., *These/*This red cars*. We create a filter that removes *all* occurrences of a demonstrative determiner followed by an adjective and a noun. A model can then still infer the number agreement from determiner/noun pairs without an intervening adjective.

3.2.2 Lexical Generalization

In the following filters we do not filter out an entire configuration, but only do so for a subset of lexical items. This way a model can indirectly generalize to a specific occurrence of the configuration from other occurrences, but no longer rely on direct co-occurrences. These filters focus on lexical generalization because the BLiMP benchmarks

[1] BLiMP assumes a straightforward one-to-one relationship between certain names and their grammatical gender. While such a relationship may not actually be borne out in practice today, the corpora used in this investigation likely do adhere to such a formulation.

that they target are centered around particular lexical items and not particular syntactic constructions.

AGR-RE-IRR-SV The four BLiMP benchmarks targeted by AGR-RE-IRR-SV all test language model performance on subject-verb agreement, targeting regular plurals, like *dress*/*dresses* and irregular plurals, like *goose*/*geese*. The filter removes all sentences with nominal subjects where the noun occurs in any of the four benchmarks. A learner on this filtered corpus can still beat the benchmark if it develops a notion of grammatical number, a representation of the grammatical number of the nouns in the benchmark based on their usage in other contexts, and then generalizes the subject-verb agreement it sees for other nouns to these nouns.

DET-NOUN The other filter besides DET-ADJ-NOUN that targets determiner-noun agreement for demonstrative determiners (e.g., *These*/**This books*) does so with the determiner directly adjacent to the noun. We create a filter based on all nouns occurring in the BLiMP benchmark that are preceded by a demonstrative determiner. A model can still infer the number agreement between determiner and noun from other nouns, and learn the number information of the filtered nouns from other agreement tasks like subject-verb agreement.

PASSIVE In English, passive constructions can only be formed from transitive verbs. BLiMP targets this phenomenon by replacing transitive verbs in passive constructions by intransitive verbs: *John is insulted by Mary* vs. **John is smiled by Mary*. Much like AGR-RE-IRR-SV and DET-NOUN, the PASSIVE filter operates by removing sentences that contain words on a word list in a specific linguistic environment. Concretely, this word list consists of the verbs that are actually used in these two benchmarks in passive form, and the filter removes sentences where such words appear in passive voice.

4 Experimental Setup

Data The base train, validation, and test corpora are the English Wikipedia corpora released by Gulordava et al. (2018), with the train corpus consisting of 3.05M sentences (83M tokens, with a vocabulary size of 50,000 plus an unknown and EOS token). The 15 filtered corpora are derived from this base corpus by discarding all sentences that are targeted by the filter. The number of sentences and tokens discarded by each filter varied from as little as $\sim 0.1\%$ to as much as $\sim 18.5\%$; for specifics, refer to Table 1. Then, as an additional control, the 15 filtered corpora plus the original, FULL training corpus were uniformly downsampled to 2.4M lines, corresponding to $\sim 80\%$ the size of the original training corpus. It is worth noting that the number of *tokens* did vary by as much as $\sim 4.2\%$, as reflected in the rightmost column of Table 1: This is explained by the fact that certain filters target longer sentences more often.

Models Two architectures are used for the models trained in this investigation: LSTMs (Hochreiter and Schmidhuber, 1997) and decoder-only Transformers (Vaswani et al., 2017). For each architecture, we train separate models on the 16 training corpora for five random seeds each, resulting in a total of 160 models. Model hyperparameters were selected to control for number of parameters as closely as possible. The LSTMs have two layers with embedding and hidden dimension of 1024. Output and embedding layer weights were tied, and we used dropout of 0.1 during training. The Transformers were constructed with feed-forward and hidden layer dimensions of 768, eight attention heads, and eight hidden layers. The LSTMs and the Transformers had 68.0M and 67.1M trainable parameters, respectively.

Training Each model was trained on a single A40 GPU for 40 epochs with mixed-precision training, using the AdamW optimization algorithm (Loshchilov and Hutter, 2017), a linear scheduler with an initial learning rate of 5×10^{-5}, and a batch size of 32. We evaluated each model at the end of every epoch, and report results for the model with the best validation perplexity. The full hyperparameter set may be found in Appendix A.

Evaluation We use four metrics—three standard and one novel—as the primary means of evaluation for all models. The first is perplexity over the (unfiltered) test corpus of Gulordava et al. (2018). The second is accuracy on each of the 67 benchmarks in the BLiMP challenge set (Warstadt et al., 2020). Accuracy on the BLiMP benchmarks was assessed via the ‘‘full-sentence’’ method (Marvin and Linzen, 2018), where a ‘‘success’’, for any minimal pair, is defined by the

model assigning a higher probability to the grammatical sentence in the minimal pair (s^+) than to the ungrammatical sentence (s^-).

However, the FiCT methodology's main advantage lies not in looking at the performance of each model in isolation, but on the *difference* in performance between two models that are otherwise identical but for their training data. Thus, for each model and each BLiMP benchmark, a change score (or delta) was calculated with respect to the average performance of all models of the same architecture trained on the FULL corpus (i.e., average over the five seeds).

To be more precise, with M a model type (i.e., $M \in \{\mathsf{LSTM}, \mathsf{Transformer}\}$), F a filter, and B a benchmark, $F(B)$ will refer to the filtered corpus targeting B, and M_F will refer to a model trained on F. We can then define the accuracy delta by:

$$\mathsf{acc}\Delta(M, F, B) := \mathsf{acc}_B^{M_F} - \overline{\mathsf{acc}_B^{M_{\text{FULL}}}} \quad (1)$$

where acc_B^M refers to the accuracy of model M on benchmark B. We will often be interested in the case where $F = F(B)$, i.e., the benchmark(s) corresponding to the corpus filter, but report others as well.

Our final evaluation metric looks at the *probability deltas* between grammatical and ungrammatical sentences:

$$P\Delta(M, F)(s) = \log P_{M_F}(s^+) - \log P_{M_F}(s^-) \quad (2)$$

$P\Delta$ expresses the magnitude of a model's grammaticality judgment: Whereas $\mathsf{acc}\Delta$ only expresses the ratio of items for which a model assigned a higher probability to the grammatical case, $P\Delta$ can be interpreted as the confidence of a model's judgment.

5 Results

We present our results along the four metrics of §4: perplexity (§5.1), TSE accuracy (§5.2), accuracy delta (§5.3), and probability delta (§5.4).

5.1 Perplexity

We found that Transformers uniformly achieve lower perplexities on the test corpus than the LSTMs for all training corpora, as expected. The mean test perplexity across all corpora and random seeds was 47.13 for the Transformers and 53.56 for the LSTMs; a paired t-test of mean perplexities per corpus found the difference between the model types to be significant ($t = 270.94$, $p \ll 0.01$). As noted in §4, while we downsampled all corpora to the same number of lines, the number of tokens varies between different training corpora. Previous research has shown a clear negative relationship between the number of tokens seen in training and test corpus perplexity (Kaplan et al., 2020). This effect is also present in our data, for both architectures (LSTMs: Pearson's $r = -0.970$; Transformers: $r = -0.976$).

We also investigate the perplexity on the BLiMP sentences for the FULL and Filtered models. This provides us insight into the likelihood of these sentences: If the model assigns a relatively low likelihood to them, then grammaticality judgments will be less reliable as well (Newman et al., 2021). In Figure 3 we show the scores for this. Surprisingly, the LSTM models yield *lower* perplexity on the BLiMP sentences than the Transformers. This shows that Transformers have shifted their probability mass to other sentence types than found in BLiMP, but where to exactly remains an open question. Nonetheless, the perplexity scores on BLiMP are similar to the average perplexity on the test corpus, which demonstrates that these items are of similar likelihood.

5.2 TSE Accuracy on BLiMP

Mean overall accuracy on all of BLiMP across different training corpora (i.e., $\overline{\mathsf{acc}_{\text{ALL}}^{M_F}}$) was 70.4 for the LSTMs and 71.9 for the Transformers. This result was statistically significant (paired $t = -17.38$, $p \ll 0.01$). Figure 6 in Appendix B shows all of the accuracies.

We next look only at benchmark accuracy data where the filtered corpus targeted a given benchmark, i.e., where $F = F_B$. Here, the mean is 68.8 for the Transformers and 66.7 for the LSTMs *and this difference is not statistically significant* (paired $t = -1.18$, $p = 0.258$). In other words, we find no difference in the two models' ability to make grammaticality judgments when trained on filtered data that forces them to perform subtle generalizations, despite differences in perplexity.

5.3 Accuracy Delta

A table of the accuracy deltas, averaged across all random seeds, can be found in Figure 2. Mean overall accuracy delta over all benchmarks and across all training corpora (i.e., $\bar{\Delta}(M, F, B)$) was

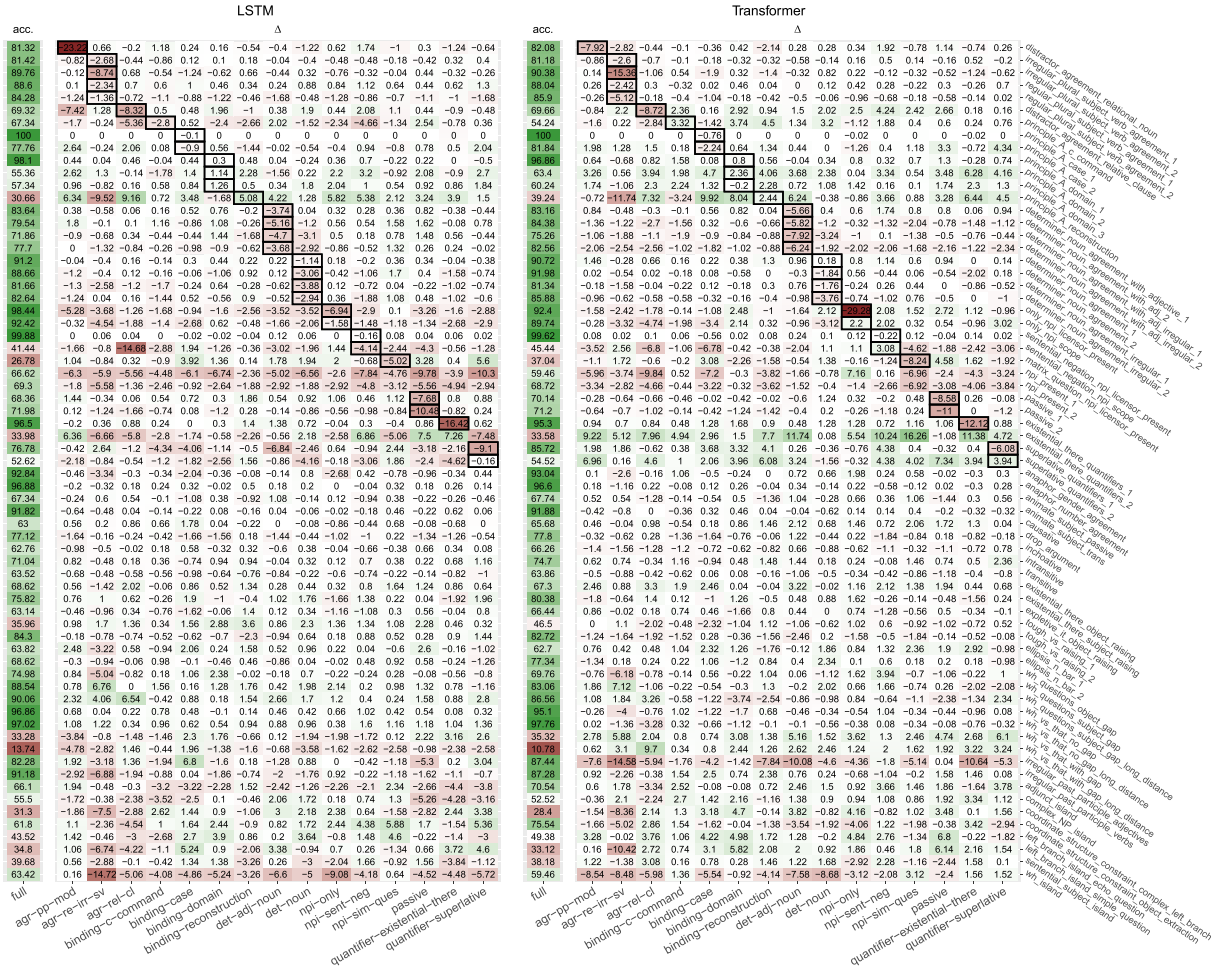

Figure 2: BLiMP benchmark *accuracy* for the models trained on the full corpus, and *accuracy delta* ($\Delta(M, F, B)$) for the filtered corpora, averaged across seeds. Boxes with bold outlines correspond to benchmarks targeted by the model's corpus filter (i.e., where $F = F(B)$). The accuracy scored by a given model on a given benchmark trained on a filtered corpus can be recovered by adding its delta to the accuracy score in the ''full'' column of the same row.

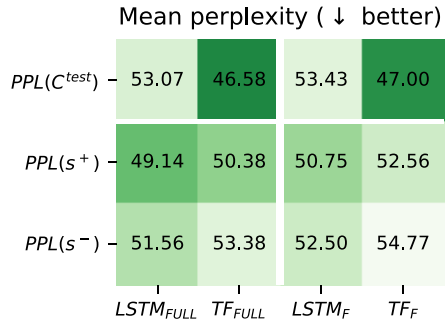

Figure 3: Perplexity scores on the test corpus (C^{test}) and the grammatical and ungrammatical BLiMP sentences (s^+ & s^-). BLiMP scores for the FULL models are averaged over all benchmarks, and for the Filtered models for their corresponding benchmark.

-0.393 for the LSTMs and 0.0313 for the Transformers. This result was statistically significant (paired $t = -5.10$, $p \ll 0.01$).

Focusing on the $F = F(B)$ cases (i.e., black-outlined cells in the chart), we note that most deltas are generally negative but fairly close to zero, with a few outliers, such as the models trained on the EXISTENTIAL-THERE, AGR-PP-MOD, and NPI-ONLY corpora. These results suggest that, overall, learners *are* usually able to use various sorts of indirect evidence to acquire correct grammatical generalizations when direct evidence has been made unavailable, as otherwise we could expect much larger deltas across the board.

We may also observe that, for the cases where the absolute value of the deltas was appreciably larger than zero, it is not the case that one architecture is uniformly better than the other. For example, LSTMs perform better than Transformers (that is, their deltas are smaller in magnitude) on the benchmarks associated

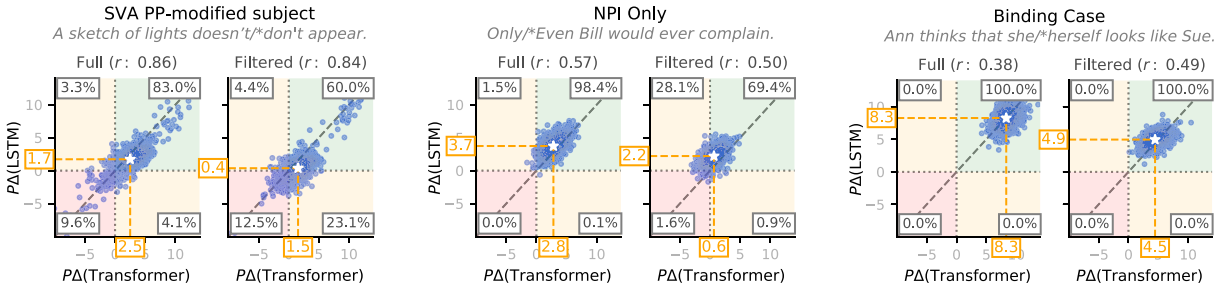

Figure 4: Log probability differences between grammatical and ungrammatical minimal pairs ($P\Delta(M, F)(s)$), with Transformer performance plotted against LSTM performance. Individual points are the averaged scores across the five model seeds. The four quadrants indicate the cases where i) both architectures got a correct prediction (green), ii) only one architecture got a correct prediction (orange), and iii) neither architecture was right (red). It can be seen that corpus filtering results in probability differences moving closer to the origin, and that the magnitude of the difference of the full models can create a sufficient margin for the model to generalize in the filtered cases as well.

with the AGR-RE-IRR-SV and the NPI-ONLY corpora, while the converse is true for AGR-PP-MOD and quantifier-existential-there. This is true *even* for phenomena that are seemingly relatively similar; for example, the AGR-PP-MOD and AGR-RE-IRR-SV-AGR filters are extremely similar, in that they both test long distance agreement in the present of a clausal distractor intervening between the subject and the verb; they differ only in the nature of that distractor. Yet, as noted, LSTMs trained on the AGR-RE-IRR-SV corpus have, on average, a less negative delta on the associated benchmarks than the analogous Transformer models ($\overline{\text{acc}\Delta}$(LSTM, AGR-RE-IRR-SV, $F(B)) = -3.78$; for the Transformer, -6.38); conversely, on the models trained on the AGR-PP-MOD corpus, it is Transformers which have the smaller magnitude delta ($\overline{\text{acc}\Delta}$(LSTM, AGR-PP-MOD, $F(B)) = -23.22$; Transformer, -7.92).

As in the previous section, we can make this precise by analyzing all of the accuracy deltas where $F = F_B$. The mean here is -5.41 for the LSTMs and -4.62 for the Transformers; this difference is not statistically significant (paired $t = -0.562, p = 0.583$). That means that we again find no difference between the two architectures in the extent to which filtering affects their accuracy, despite significant differences in perplexity. This suggests that perplexity *does not* predict the ability of a model to perform linguistic generalizations from indirect evidence.

5.4 Probability Delta

In order to gain a more fine-grained insight into the impact of corpus filtering, we examine the results at an item-level. For this we make use of the $P\Delta$ metric, which expresses a model's magnitude of a grammaticality judgment. In Figure 5A we plot the average $P\Delta$ scores for the FULL models for each BLiMP benchmark, averaged across seeds. It can be seen here that the Transformers and LSTMs result in highly similar $P\Delta$'s ($r = 0.98; p \approx 0$), although the Transformer scores are slightly higher on average than those of the LSTMs (2.99 vs. 2.41, respectively), which is in line with the significant difference in TSE accuracy of §5.2.

For the sake of brevity, we focus on three salient filters that each yielded distinct results: i) Subject-Verb Agreement for PP-modified subjects, in which LSTMs are more impacted than Transformers (accΔ: -23.2 vs. -7.9); ii) NPI Only, in which LSTMs are *less* impacted than Transformers (accΔ: -6.9 vs. -29.3); and iii) Binding Case, in which neither architecture is impacted by filtering. In Figure 4 we plot the item-level scores of the LSTMs against the Transformers (averaged across seeds). For each benchmark B we plot the results on the FULL model and the $F(B)$ filtered model. This demonstrates that corpus filtering has the effect of moving $P\Delta$ closer to the origin: The model becomes *less certain* in its grammaticality judgment. The resulting accΔ score for a benchmark is then dependent on the $P\Delta$ scores of the FULL model: A sufficient margin here makes it robust to the decrease in $P\Delta$ and allows it to correctly assign higher probability to the grammatical item.

To investigate this observation across all benchmarks we plot the difference in $P\Delta$ going from FULL to Filtered in Figure 5B. This difference

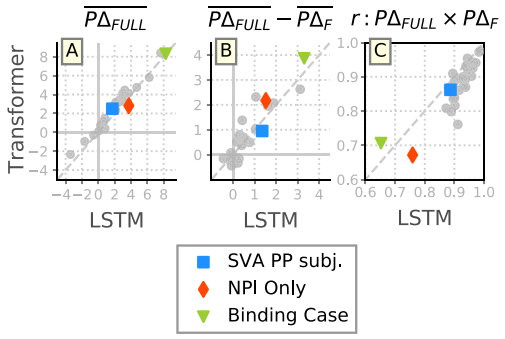

Figure 5: A: $P\Delta$ scores for the FULL Transformers and LSTMs for each BLiMP paradigm. The more positive this score, the more certain a model is in its grammaticality judgment. B: Paradigm-level differences in $P\Delta$ scores going from the FULL to the Filtered model. The closer to the origin, the less impact the filtering procedure had on model behavior. C: Pearson correlation of $P\Delta$ scores between the FULL and Filtered models. A detailed table with these results per paradigm is provided in Figure 7 in Appendix B.

represents the *absolute impact* of filtering on the TSE task. By plotting the Transformer results against the LSTM we gain an insight whether filtering has a similar impact on both architectures. We observe a strong correlation between these $P\Delta$ differences (r : 0.91, $p \approx 0$). Subtle difference are present, however, for a number of filters the $P\Delta$ score *increases* after filtering which is especially prevalent for the Transformer models.

Finally, we examine the *robustness* of a model's grammaticality judgments: Does filtering have a significant impact on the distribution of judgments? For this we compute the Pearson correlation of $P\Delta$ before and after filtering for each filter benchmark. A model is robust to filtering if this correlation remains high. In Figure 5C we plot the LSTM correlations against the Transformer. A striking difference between the two architectures arises here: the LSTM correlations are systematically larger than those of the Transformer. This shows that LSTMs are less impacted by filtering on an item-level than Transformers.

6 Discussion

Perplexity Versus Linguistic Generalization Our findings contribute to a growing body of research that suggest a dissociation between perplexity and more targeted evaluations of linguistic competence in artificial learners (Hu et al., 2020). In a carefully controlled setting and for a wide range of phenomena, we demonstrate that the training objective of minimizing perplexity does not predict linguistic generalization. This raises interesting questions on the relation between perplexity and grammaticality judgments (Lau et al., 2017): While Transformers are better at *memorizing* the structure of its training data, we show they are less capable than LSTMs of forming robust linguistic generalizations. An interesting step for future work would be to uncover what language modeling aspects Transformers *do* excel at, which allows them to obtain a superior test perplexity (e.g., word frequency, as studied in Wei et al., 2021). Future work should also compare our measure(s) of generalization with others in the literature, given evidence that these are not always well-correlated with each other (Sun et al., 2023).

We also note that while likelihood judgments do not necessarily directly measure grammaticality, since likelihood is the outcome of many other factors (e.g., semantic plausibility, pragmatic felicity), the use of minimal pairs for BLiMP does help control for this since it reports judgments on sentences which differ on (usually) one word, thus keeping these other components constant between the two sentences. That being said, it would be a worthwhile follow-up to conduct probing experiments to more directly model grammaticality judgments, in the style of, e.g., Jumelet et al. (2021) (see the next subsection as well).[2]

Our results also have consequences for how we think about language model evaluation more broadly: To the extent that we believe that models should be able to generalize from indirect evidence, we cannot rely on perplexity as the sole measure of LM quality but must measure and test for this ability directly.

Generalizing from Indirect Evidence Our study also builds on the insights of numerous other works that use artificial learners as models for understanding human language acquisition, and gaining better insights in the inductive biases of such learners (Warstadt and Bowman, 2020; Mueller and Linzen, 2023; Weber et al., 2024). The present study conducts for a wide range of phenomena what Warstadt (2022) calls a ''proof-of-concept [of a] large-scale controlled ablation study on the input to model learners,''

[2]We thank an anonymous reviewer for encouraging us to think about this distinction.

and finds that direct attestation of linguistic evidence is not strictly necessary for the development of sophisticated linguistic generalizations. Rather, learners can leverage much more indirect sources of evidence to arrive at the correct generalizations.

Where earlier work has focused on specific linguistic constructions, such as subject auxiliary inversion (Warstadt, 2022), relative clauses (Prasad et al., 2019), and negative polarity items (Warstadt et al., 2019a; Jumelet et al., 2021; Weber et al., 2021), the results of this paper essentially confirm a similar result for a much wider array of syntactic and semantic phenomena. While in many cases the ablations we performed did clearly negatively affect the performance of our artificial learners on the relevant linguistic evaluations, the magnitude of this effect was generally quite small for all but a small handful of the linguistic phenomena we analyzed. In general, even when tested on the specific benchmarks corresponding to the environments that were ablated from their input, models still perform considerably better than chance. Thus, our research provides evidence in favor of the indirect evidence hypothesis.

Notably, we find that this is true not only for filters where there are fairly obvious sources of indirect evidence (as enumerated in §3), but also for filters where potential sources of indirect evidence for a correct generalization are much less clear (such as the SUPERLATIVE-QUANTIFIER filter). This suggests that there may be complex mechanisms by which certain linguistic generalizations can be derived via highly indirect means. Thus, our results open a door to future research that can provide a more thorough account of the source of these generalizations, with potentially significant ramifications for linguistics.

Explaining Linguistic Generalization As just discussed, the primary contribution of this paper has been the development of the FiCT method and the use of it to demonstrate LMs' successful generalization from indirect evidence across a *wide range* of linguistic phenomena. This success raises a very natural follow-up question: What explains this successful generalization behavior?

While a complete answer to this question must await future work, a detailed look at the NPI cases can provide insight into what an answer may look like. Jumelet et al. (2021) used a filtered corpus method to test LSTM LMs' understanding of negative polarity items, but then also did a further analysis to examine the basis upon which the models made their grammaticality judgments. In particular, they found (via probing classifiers) that LMs' were successfully recognizing the *monotonicity* of a linguistic environment and (via a novel correlation method) that these judgments of monotonicity were highly correlated with the LMs' judgment of NPI acceptability, reflecting human acceptability judgments (Denić et al., 2021; Chemla et al., 2011).

This example suggests two paths forward for explaining the generalization observations in the present paper. On the one hand, in the same way that the monotonicity explanation was inspired by human generalization, detailed explanations of individual cases of generalization can be developed with human behavior as an initial inspiration. On the other hand, in the same way that this paper extends the filtered corpus training method to a much wider range of phenomena, one can attempt to generalize these forms of explanation on the breadth axis as well. We leave these exciting pursuits to future work.

7 Conclusion

We introduced the **Fi**ltered **C**orpus **T**raining methodology and applied it to a wide range of linguistic constructions from the BLiMP benchmark. Our results show that while Transformers are better language models (via perplexity) than comparable LSTMs, the latter generalize equally well (via $\mathsf{acc}\Delta$ and $P\Delta$). The relatively low $\mathsf{acc}\Delta$ scores in general show that all of our LMs exhibit a strong ability to generalize from indirect evidence, even for models of relatively low parameter count trained on relatively small data. In summary, this shows that language model success cannot be attributed solely to memorization from its training data, since the data has been systematically purged of the evaluation targets. They are, instead, able to form subtle and linguistically relevant generalizations from indirect evidence.

Future work will i) extend this approach to models of different sizes and pretraining corpora, ii) perform deeper analyses of the bases on which the models do make their generalizations (including with probing experiments), and iii) analyze other forms of lexical and structural generalization through the lens of the filtered corpus training methodology.

Acknowledgments

For helpful discussion, we thank Milica Denić, Dieuwke Hupkes, Jakub Szymanik, and the audience at the UW Computational Linguistics Treehouse. We thank the action editor and the anonymous reviewers for their valuable feedback.

Author Contribution Statement

Following a practice in several other fields, we here list author contributions according to the Contributor Role Taxonomy (CRediT; Allen et al., 2019). **Abhinav Patil**: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing—original draft, Writing—review and editing, Visualization, Supervision. **Jaap Jumelet**: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing—original draft, Writing—review and editing, Visualization, Supervision. **Yu Ying Chiu**: Software, Data curation, Writing—review and editing. **Andy Lapastora**: Methodology, Software, Investigation, Data curation, Writing—review and editing. **Peter Shen**: Software, Data curation, Writing—review and editing. **Lexie Wang**: Software, Data curation, Writing—review and editing. **Clevis Willrich**: Software, Data curation, Writing—review and editing. **Shane Steinert-Threlkeld**: Conceptualization, Methodology, Software, Formal analysis, Resources, Writing—original draft, Writing—review and editing, Supervision, Project administration.

References

Liz Allen, Alison O'Connell, and Veronique Kiermer. 2019. How can we ensure visibility and diversity in research contributions? How the Contributor Role Taxonomy (CRediT) is helping the shift from authorship to contributorship. *Learned Publishing*, 32(1):71–74. https://doi.org/10.1002/leap.1210

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 301–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.conll-1.20

Emmanuel Chemla, Vincent Homer, and Daniel Rothschild. 2011. Modularity and intuitions in formal semantics: The case of polarity items. *Linguistics and Philosophy*, 34(6):537–570. https://doi.org/10.1007/s10988-012-9106-0

Noam Chomsky. 1993. *Lectures on Government and Binding*. De Gruyter Mouton, Berlin, New York. https://doi.org/10.1515/9783110884166

Milica Denić, Vincent Homer, Daniel Rothschild, and Emmanuel Chemla. 2021. The influence of polarity items on inferential judgments. *Cognition*, 215:104791. https://doi.org/10.1016/j.cognition.2021.104791

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48. https://doi.org/10.1162/tacl_a_00298

Victoria Fossum and Roger Levy. 2012. Sequential vs. hierarchical syntactic models of human incremental sentence processing. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, pages 61–69, Montréal, Canada. Association for Computational Linguistics.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1004

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*,

pages 70–76, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-demos.10

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-0102

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1108

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. https://doi.org/10.3115/1073336.1073357

Yiding Hao, Simon Mendelsohn, Rachel Sterneck, Randi Martinez, and Robert Frank. 2020. Probabilistic predictions of people perusing: Evaluating metrics of language model performance for psycholinguistic modeling. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 75–86, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.cmcl-1.10

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

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. https://doi.org/10.18653/v1/2020.acl-main.158

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, 5:1161–1174. https://doi.org/10.1038/s42256-023-00729-y

Jaap Jumelet, Milica Denic, Jakub Szymanik, Dieuwke Hupkes, and Shane Steinert-Threlkeld. 2021. Language models use monotonicity to assess NPI licensing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4958–4969. Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.439

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? On the ability of LSTMS to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5424

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R. Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

pages 9087–9105, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.731

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower perplexity is not always human-like. 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 5203–5217, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.405

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241. https://doi.org/10.1111/cogs.12414

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177. https://doi.org/10.1016/j.cognition.2007.05.006

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. https://doi.org/10.1162/tacl_a_00115

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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. https://doi.org/10.18653/v1/D18-1151

Tom 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. https://doi.org/10.18653/v1/P19-1334

Gary Milsark. 1974. *Existential Sentences in English*. Ph.D. thesis, MIT, Cambridge, MA.

Kanishka Misra and Kyle Mahowald. 2024. Language Models Learn Rare Phenomena from Less Rare Phenomena: The Case of the Missing AANNs.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11237–11252, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.629

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.290

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. Valencia, Spain. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023a. Transformer-based language model surprisal predicts human reading times best with about two billion training tokens. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1915–1921, Singapore. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023b. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350. https://doi.org/10.1162/tacl_a_00548

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency explains the inverse correlation of large language models' size, training data amount, and surprisal's fit to reading times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2644–2663, St. Julian's, Malta. Association for Computational Linguistics.

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA. https://doi.org/10.1017/CBO9780511803161

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. https://doi.org/10.18653/v1/K19-1007

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Kaiser Sun, Adina Williams, and Dieuwke Hupkes. 2023. The validity of evaluation results: Assessing concurrence across compositionality benchmarks. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 274–293, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.conll-1.19

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*, volume 30. Curran Associates, Inc.

Alex Warstadt. 2022. *Artificial Neural Networks as Models of Human Language Acquisition*. Ph.D. thesis, New York University. https://doi.org/10.1201/9781003205388-2

Alex Warstadt and Samuel R. Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *42nd Annual Meeting of the Cognitive Science Society: Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020*, pages 1737–1743.

Alex Warstadt and Samuel R. Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic Structures in Natural Language*, pages 17–60. CRC Press. https://doi.org/10.1201/9781003205388-2

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019a. Investigating BERT's knowledge of language: Five analysis methods with NPIs. 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 2877–2887, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1286

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392. https://doi.org/10.1162/tacl_a_00321

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641. https://doi.org/10.1162/tacl_a_00290

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2021. Language modelling as a multi-task problem. In *Proceedings of the*

16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 2049–2060, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.176

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2024. Interpretability of language models via task spaces. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.248

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.72

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Philip Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 42.

A Training Hyperparameters

adam_beta1	0.9
adam_beta2	0.999
adam_epsilon	1e-08
dataloader_num_workers	8
evaluation_strategy	epoch
fp16	True
gradient_accumulation_steps	1
ignore_data_skip	True
learning_rate	5e-05
lr_scheduler_type	linear
num_train_epochs	40
per_device_train_batch_size	32
per_device_eval_batch_size	32
optim	adamw_torch
seed	0,1,2,3,4
save_strategy	epoch

Table 2: Selected training hyperparameters, as provided to the `transformers` package's `TrainingArguments` class. Any omitted values were set to the defaults associated with version 4.30.2 of the `transformers` package.

B Full Result Tables

Figure 6 contains the mean accuracies (across random seeds) on all BLiMP benchmarks for both models and every filtered corpus.

Figure 7 contains the paradigm-level $P\Delta$ scores for the FULL and Filtered models, and various Pearson correlations.

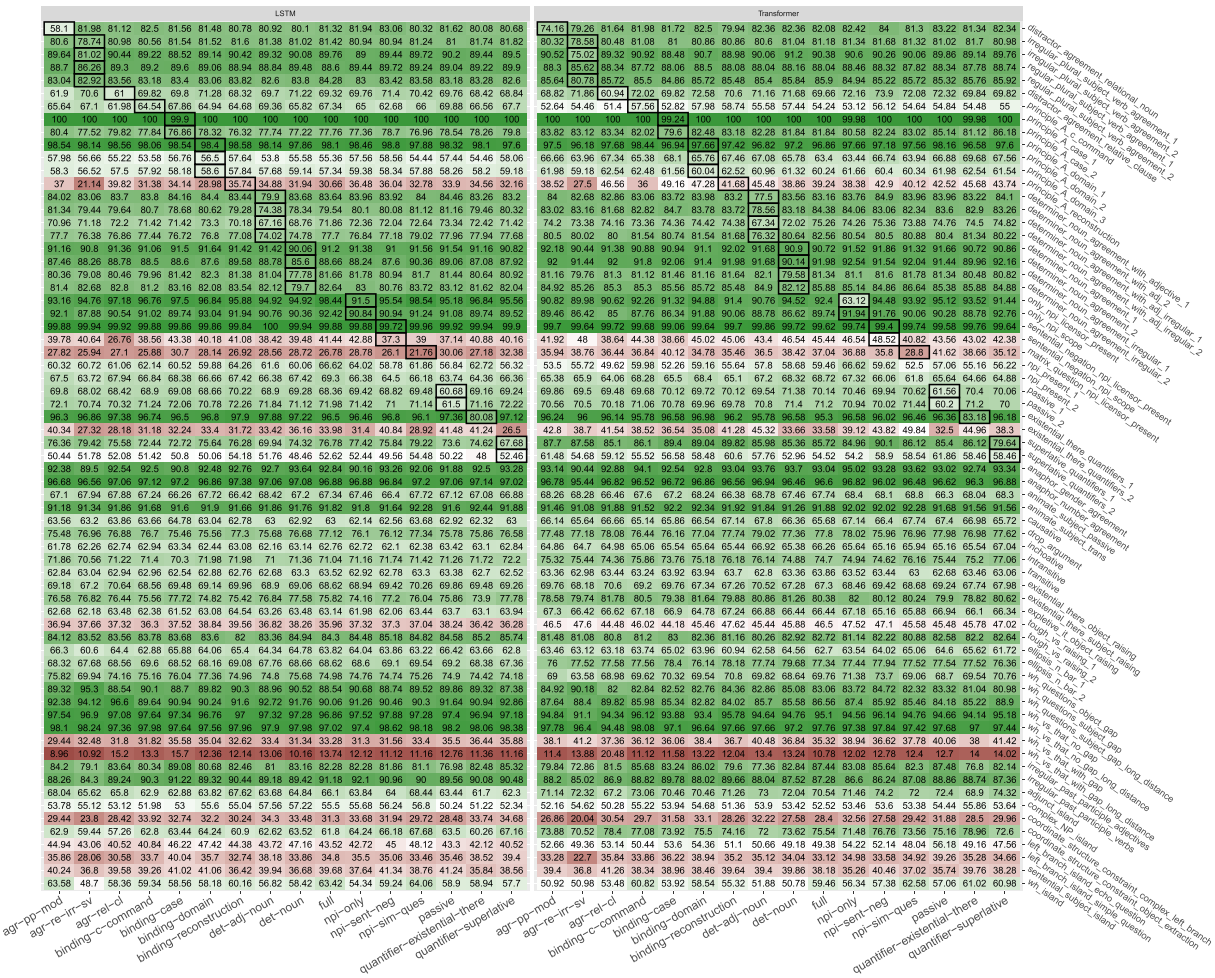

Figure 6: Complete BLiMP benchmark accuracy results for all models, averaged across the five starting seeds for a given training corpus and benchmark. Boxes with bold outlines correspond to benchmarks targeted by the model's corpus filter (i.e., where $F = F(B)$).

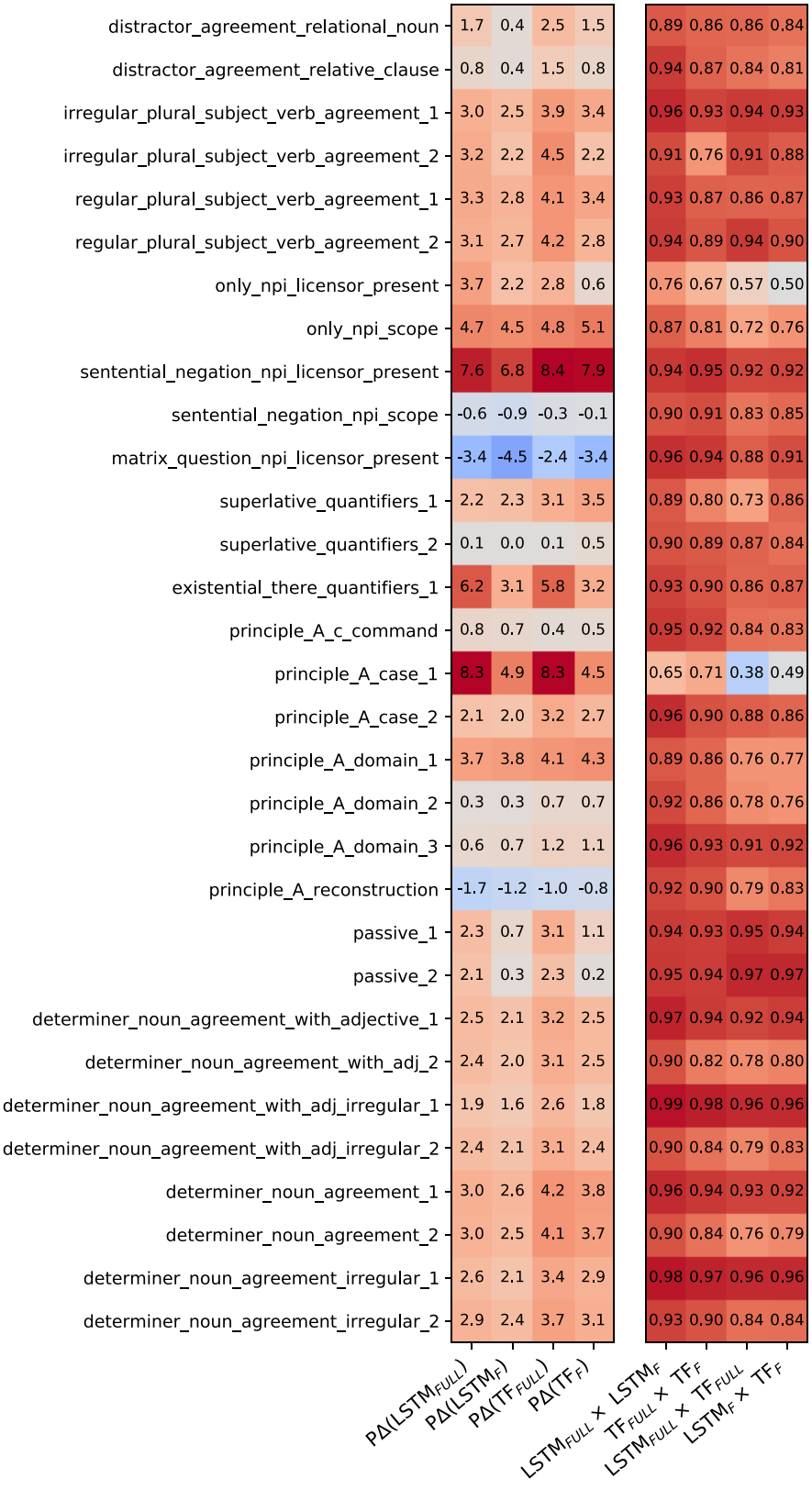

Figure 7: $P\Delta$ scores for the LSTMs and Transformers (first four columns), and the Pearson correlations between these $P\Delta$ scores (last four columns).