Language model acceptability judgements are not always robust to context

Koustuv Sinha [*,∞] **Jon Gauthier** [*,1]
Aaron Mueller [†,3] **Kanishka Misra** [†,2] **Keren Fuentes** [∞]
Roger Levy [1] **Adina Williams** [∞]
[∞]Meta AI; [1]MIT [2]Purdue University [3]Johns Hopkins
*, † Equal contributions
koustuvs@meta.com, jon@gauthiers.net

Abstract

Targeted syntactic evaluations of language models ask whether models show stable preferences for syntactically acceptable content over minimal-pair unacceptable inputs. Our best syntactic evaluation datasets, however, provide substantially less linguistic context than models receive during pretraining. This mismatch raises an important question: how robust are models' syntactic judgements across different contexts? In this paper, we vary the input contexts based on: length, the types of syntactic phenomena it contains, and whether or not there are grammatical violations. We find that model judgements are generally robust when placed in randomly sampled linguistic contexts, but are unstable when contexts match the test stimuli in syntactic structure. Among all tested models (GPT-2 and five variants of OPT), we find that model performance is affected when we provided contexts with matching syntactic structure: performance significantly improves when contexts are acceptable, and it significantly declines when they are unacceptable. This effect is amplified by the length of the context, except for unrelated inputs. We show that these changes in model performance are not explainable by acceptability-preserving syntactic perturbations. This sensitivity to highly specific syntactic features of the context can only be explained by the models' implicit in-context learning abilities.

1 Introduction

The unprecedented progress in the development of neural large language models (LLMs; Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022) has been accompanied by a comparable proliferation of methods that aim to better understand and characterize models' linguistic capacities (Linzen et al., 2016; Ettinger et al., 2016; Alishahi et al., 2019; Hu et al., 2020; Jeretic et al., 2020; Mueller et al., 2020, *i.a.*). Of the many methods for this, the minimal-pair paradigm (MPP), which is methodologically standard in linguistics, has emerged as a popular approach to evaluate models' knowledge of linguistic phenomena in an unsupervised manner (Marvin and Linzen, 2018; Kann et al., 2019; Warstadt et al., 2019, 2020a; Misra et al., 2023). Under the MPP, models are presented with datasets containing pairs of minimally differing text sequences—usually differing in word order or in a few tokens—one of which is deemed by humans to be acceptable and the other unacceptable. Drawing on the LLMs' trained ability to produce probabilities over token sequences, we can evaluate them according to the MPP by testing whether models assign relatively greater probability to the acceptable sequence.

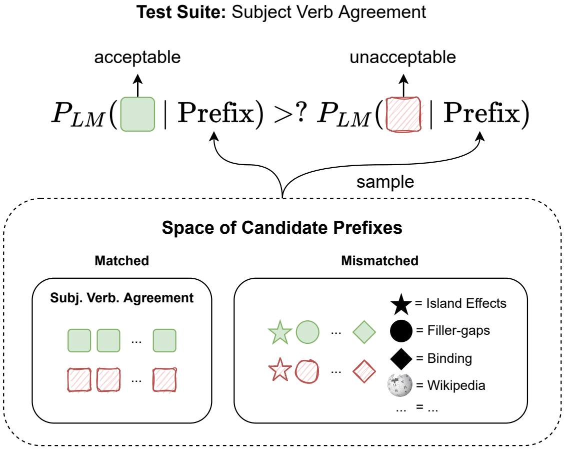


Figure 1: We measure the impact of different contexts on the performance of an LM on linguistic acceptability tasks by prefixing sentences (here, sourced from subject-verb agreement challenge sets) from a diverse sources. Each block represents a sentence: **Red** striped blocks are unacceptable sentences within a given task, while **green** solid ones are acceptable. We also vary the diversity of prefixes by sampling them from tasks/datasets different from the test suite (indicated by shape).

Studies that employ MPP datasets generally compare the probability of two stand-alone text sequences without any explicit linguistic context. However, this is not a naturalistic or realistic ap-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6043–6063
July 9-14, 2023 ©2023 Association for Computational Linguistics

proach: utterances usually occur *in some linguistic context*, where the context itself could affect linguistic preferences. The syntactic priming literature investigates the effect of linguistic contexts to some extent, but mostly in a constrained setting with only one or a small number of context sentences (van Schijndel and Linzen, 2018; Prasad et al., 2019). The interaction of context with minimal pair accuracies remains underexplored for multi-sentence contexts, despite the fact that multi-sentence inputs are more likely for many NLP tasks—especially with the rise of prompting and in-context learning (Brown et al., 2020; Schick and Schütze, 2021b). Furthermore, Transformer-based language models are typically trained on large sequences, where masked tokens are predicted given a completely full context window, consisting of many sentences. It is unclear how to evaluate MPP by utilizing this context window, given recent research that has raised questions about the sentence representations acquired in long-form input (Sinha et al., 2022; Haviv et al., 2022).

We evaluate the sensitivity of LLMs' acceptability preferences in a more realistic evaluation setting, with one or more additional sentences in the input context. We focus on LLM sensitivity to three particular features of the context: (1) the length of the input sequence, (2) the similarity of the context to the minimal pair being judged, and (3) whether the context itself contains acceptability violations. Figure 1 illustrates our method at a high level: For a given MPP dataset (BLiMP, Warstadt et al. 2020a and SyntaxGym, Hu et al. 2020), we generate new minimal pair test examples for a given syntactic phenomenon by artificially simulating a long context window. Specifically, we prepend the given test example pair with sentences drawn by the axis of similarity, from *unrelated* (Wikipedia), minimal-pair sentences from different (*mismatched*) or the same (*matched*) syntactic phenomena in the MPP dataset. We also introduce violations in the context by drawing unacceptable counterparts of the above similarity scale from the MPP dataset.

We find that the model's judgements are highly robust to the presence of unrelated Wikipedia sentences in the context, regardless of the size of the prefix. However, we observe strong sensitivity to matched context manipulations. As the context length increases, acceptable matched contexts improve the models' judgements significantly. Conversely, we observe a strong opposite effect of exposing the model to longer and longer prefixes containing acceptability violations: models' judgements degrade drastically, performing far below chance. This sensitivity is specific to the particular type of syntactic structural similarity of the context: we do not see the same degree of improvement/degradation in prediction behavior for contexts consisting of mismatched sentences of valid or violated syntactic structures.

To better understand our results, we performed several exploratory analyses. To determine whether the results are an effect of the acceptability judgement task, we replicated our experiments for another task, that of stereotypicality judgements (Nangia et al., 2020), and found largely concurring results. We also investigated the syntactic overlap between the context and the test pair, and observe only minor effects on the judgements with phenomena-preserving syntactic perturbations. Our results, therefore, can only be explained by the model displaying some kind of implicit, instruction-free, in-context learning ability, and they invite further scrutiny of and investigation into long-form sentence understanding capabilities of LLMs.

2 Background

Sequence Length and Out-of-domain Generalization. When evaluating language models' linguistic abilities in particular, one ought to additionally consider the *domain* of the test data fed into the model, as it can have large consequences for model performance if it mismatches from the model training data. Length mismatches are quite common in NLP datasets. For example, MPP test sequences are considerably shorter than that of the inputs LLMs typically receive during pre-training (≈ 512–1024 tokens)—the test pairs in standard MPP datasets for the linguistic acceptability task, for example, are ≈ 4–30 tokens in the case of BLiMP. It is also relatively well established that mismatching sequence lengths between (pre-)training and testing scenarios can affect performance (Hupkes et al., 2020; Newman et al., 2020; Varis and Bojar, 2021; Hupkes et al., 2022), raising the question: how much does test sequence length impact our measurements of model performance on MPP datasets? We contextualize LLMs' performance on acceptability judgements against work in length extrapolation, and analyze generalization during test time to both shorter and longer sequences.

Priming Language Models. Recent work has explored the effects of providing additional linguistic context to LLMs by "priming" or prepending their inputs with words/sentences.[1] For instance, Misra et al. (2020) and Kassner and Schütze (2020) show LLMs' behave in ways that are reminiscent of semantic priming, assigning greater probabilities to words that were semantically related to their words/sentence prefixes. More recently, Sinclair et al. (2022) used a priming paradigm to measure the probability assigned by LLMs to sentences prefixed with well-formed but structurally different sentences. They found that several autoregressive LLMs assign greater probability to sentences that are similar in structure to their prefixes across a number of diverse constructions, thereby demonstrating a pattern analogous to what is known in psycholinguistics as structural priming (Bock, 1986; Pickering and Ferreira, 2008). Together with the findings of van Schijndel and Linzen (2018); Prasad et al. (2019), these works suggests that LLMs may represent at least some of the relevant structural similarities between sentences, and that their word predictions could reflect an expectation of repeating structures. While these methods do not focus on length *per se*, their manipulation of the input context is necessarily accompanied by an increase in length. This leaves open the question as to how structural properties of the context may interact with varying levels of input lengths.

In-context Learning. A practical application of the priming paradigm is that it can be used to elicit learning behavior in LLMs. That is, LLMs can be primed using labelled task demonstrations (Brown et al., 2020), instructions/explanations (Lampinen et al., 2022, though see Webson and Pavlick., 2022), or a combination of the two (Wei et al., 2022; Kojima et al., 2022) as supervision for tasks such as sentiment analysis or reasoning. This suggests that LLMs seem to be able to extract higher-level information from their context when processing a new test example from a supervised task. Our approach contributes to this body of work by testing whether LLMs can also extract more abstract features, such as grammaticality or stereotypicality, given enough priming examples.

3 Approach

Terminology. We follow standard practice in MPP, where we evaluate the *preference* ($\mathcal{P}$) of a language model M towards acceptable sentence (x) over its unacceptable counterpart (x'), with respect to log-likelihood, and compute the value over the full evaluation dataset D. D typically consists of several *test suites*, each of which instantiates a particular linguistic phenomenon. We denote the particular test suite under evaluation as the *target suite*: $S \subset D$. Each target suite consists of k pairs of acceptable and unacceptable sentences, $(x, x')_{i=1}^{k} \in S$, and may have multiple conditions. Within each target suite, we compute the acceptability judgements on one or more *experimental conditions*, comparing a given LM's log-likelihood preference $\mathcal{P}$ for the acceptable and unacceptable sentence in each condition. The accuracy ($\mathcal{A}$) over a test pair from a single condition is defined as:

$$\mathcal{A}(x_i, x'_i) = \mathbb{1}[\mathcal{P}(x_i) > \mathcal{P}(x'_i)], \quad (1)$$

where $\mathbb{1}$ is the indicator function which returns 1 if the inequality is satisfied and 0 otherwise. Depending on the dataset, it can have either one or multiple conditions evaluated for each test item.

To simulate increasing length of input, we prepend a prefix sequence c to both x and x', and compute the preferences over the concatenated sequence, $\mathcal{P}([c, x_i])$ and $\mathcal{P}([c, x'_i])$, where c can be arbitrarily long.

Datasets. We focus on the standard targeted syntactic evaluation datasets of BLiMP (Warstadt et al., 2020a, licensed CC-BY) and SyntaxGym (Hu et al., 2020, MIT license). BLiMP is a large-scale MPP dataset consisting of 67 different subsets of 1000 English sentence pairs each. Each BLiMP subset targets a particular linguistic paradigm that belongs to 12 different overarching linguistic phenomena—for instance, *subject-verb agreement*, *argument structure*, etc. SyntaxGym is a syntactic evaluation benchmark designed with more stringent evaluation criteria. For 34 different linguistic phenomena, the SyntaxGym benchmark defines test items with two to four different conditions, consisting of minimal structural variations on the same sentence which render the sentence either grammatical or ungrammatical. Model log-likelihoods are measured at a *critical region* within each sentence, rather than across the whole sentence, and models are expected to produce log-likelihoods that satisfy

[1] This is related to but differs from the operationalization of priming as finetuning/adaptation as developed by van Schijndel and Linzen (2018); Prasad et al. (2019).

multiple inequalities across all conditions. SyntaxGym is smaller than BLiMP (with about 20 items per phenomenon on average) and all of the examples are hand-written. We adapt 23 of the 34 test paradigms in SyntaxGym whose structure was compatible with the prefixing analyses of this paper.[2] These two datasets offer complementary value to the analyses in this paper: BLiMP's large scale allows us to make general conclusions about the average effect of prefix interventions, while SyntaxGym's stringent evaluation allows us to verify that the effects are sustained under more rigorous experimental conditions.

To better understand whether our results are specific to syntactic evaluation MPP datasets, we also replicate a portion of our experiments using the CrowS-Pairs dataset for stereotype evaluation (Nangia et al., 2020, licensed CC-BY-SA). CrowS-Pairs examples fall into 9 bias types (e.g., race, gender, age) and consist of minimal pairs with one stereotypical sentence and one less stereotypical sentence about a historically disadvantaged group. We view the bias types in CrowS-Pairs as analogous to particular linguistic test suites in BLiMP or SyntaxGym for the purposes of our replication: we re-code "less-stereotypical" as "acceptable" and "more-stereotypical" as "unacceptable".[3] More discussion of the dataset and further methodological information is provided in Appendix A.

Method. We compute the log-likelihood of the given input using the `minicons` library (Misra, 2022),[4] which is based on `huggingface` (Wolf et al., 2020). For each dataset D, we first compute the baseline acceptability accuracy according to Equation 1. Next, we re-evaluate the acceptability accuracy as we steadily increase the token length of the input. Following prior work on priming (§2), we analyze how prepending the test examples with additional context affects a given model's acceptability judgements.

To increase the token length while maintaining the MPP formulation, we introduce a context c by prepending the same sequence to each target x and x' in S. To construct a context c, we sample from several possible sources (acceptable sentences, unacceptable sentences, and control sentences) discussed below. We also gradually increase the length of the context c by sampling multiple sentences from a known set, and concatenating them with periods and spaces as delimiters.

Next, we recompute the log-likelihood over the stimuli (x or x') by conditioning on c, i.e., $\mathcal{P}([c, x_i]) = \log p(x_i \mid c)$.[5] For each item pair (x_i, x'_i) in target suite $S \in D$, we first sample *acceptable* sentences to construct context c as follows:

- *Matched*: Contexts are sampled from the same test suite (or bias type) as the target suite S: $x, c \in S \mid x \neq c$.
- *Mismatched*: Contexts are sampled outside the target suite (or bias type) S: $x \in S, c \in D \mid c \notin S$.

For each $x \in S$, we construct the context c by sampling N sentences (without replacement) from each group, concatenating them, until the input reaches 1000 tokens.[6]

Traditionally, most work on priming has only considered grammatically acceptable sentences as the context. While there has been some work on syntactic priming in humans showing they can be primed with ungrammatical sentences to produce other ungrammatical sentences (Kaschak and Glenberg, 2004; Pickering and Garrod, 2017; Yang and Stocco, 2019), there is little evidence in the NLP literature about how a model would react to grammatically *unacceptable* sentences in the input. Therefore, we perform our evaluation on both acceptable prefixes ($c \in x$) and unacceptable prefixes ($c \in x'$), drawn from the same phenomena (*matched*, $c \in S$) or from a different phenomena (*mismatched*, $c \notin S$).

For evaluation, we compute the Δ *accuracy* of acceptability judgements for each model:

$$\frac{1}{|D|}\sum_{i}^{|D|}\mathcal{A}([c, x_i], [c, \hat{x}_i]) - \frac{1}{|D|}\sum_{i}^{|D|}\mathcal{A}(x_i, \hat{x}_i), \tag{2}$$

where $|D|$ is the total number of samples in a given dataset (D). Taking this difference allows us to quantify the precise contribution (in terms of the

[2] See Appendix F for more technical details on the SyntaxGym analysis.

[3] Our definition of "unacceptable" for the CrowS-Pairs does not imply grammatically ill-formed, but instead it implies socially inappropriate. We are aware that recoding in this way does some terminological violence to the well established psycholinguistic term *(un-)acceptable* (c.f. Chomsky 1965; Schütze 1996), which we chose to do for reasons of space.

[4] `https://github.com/kanishkamisra/minicons`

[5] Since c is held constant for every item, the difference in the conditional measure is equivalent to that in the full sequence log-likelihood.

[6] Since GPT and OPT models have a context window of 1024 tokens, we investigate 1000 tokens as an approximate.

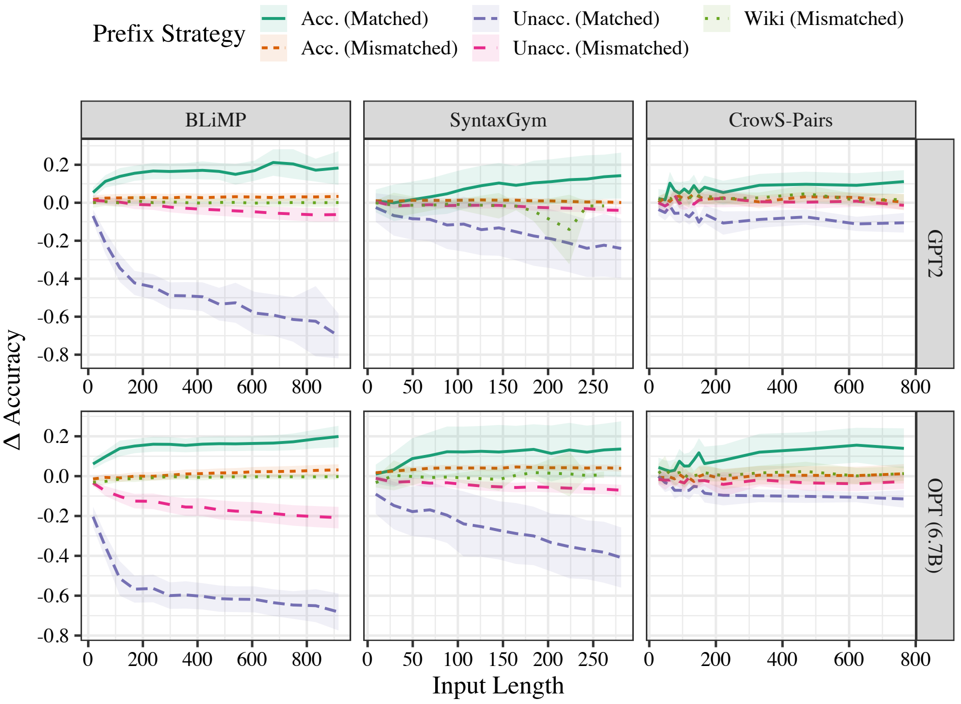


Figure 2: Prefixing type affects model performance (Δ Accuracy) for GPT2 and OPT (6.7B) on BLiMP, SyntaxGym and CrowS-Pairs datasets. Longer prefixes tend to elicit performance enhancement, an effect which is modulated by whether the prefixes are acceptable, and whether the prefixes match the test suite/bias.

gain or loss in accuracy of the LM on the acceptability task) of the priming contexts (c), which are held constant for a given pair of test samples. It further allows us to report a unified measure across our systematic manipulations of the context.

Models. We study autoregressive language models at varying sizes—we consider GPT2 (small, 124M parameters) (Radford et al., 2018), and a subset of the OPT family (125M, 350M, 1.3B, 2.7B and 6.7B parameters; Zhang et al. 2022).

Control. While we define matched and mismatched with respect to the phenomena or bias type provided by the dataset (target suite, S), we are still in the regime of *in-distribution* prefix sentences, as the context is drawn from the same MPP dataset. By design, these sentences are lexically constrained, and constructed to be as simple as possible while still testing for the relevant phenomena. To simulate an *out-of-distribution* context relative to the BLiMP/SyntaxGym test examples, we sample prefix sentences from a completely unrelated Wikipedia domain, the WikiText-103 test set (Merity et al., 2017).

Regression Analysis. We define and test our claims about the effect of length on acceptability with a mixed-effects logistic regression for each combination of model and dataset. The regression predicts a model's acceptability judgement accuracy for a given phenomenon as a function of the three previously introduced properties of the prefix c: its length, whether it is matched or mismatched, and its acceptability. The model includes a three-way interaction term and all lower-order terms for these variables, with sum-coded categorical variables and log-transformed prefix lengths, along with a random intercept term for the phenomenon (controlling for variation in baseline accuracies per phenomenon).

4 Main Results

Figure 2 presents the summary results of our prefixing manipulation, charting models' accuracy on MPP evaluations as a function of the prefix (1) length (x-axis), (2) acceptability (teal and orange vs. red and purple), and (3) whether it is drawn from a domain that is matched (teal and purple), mismatched (orange and red), or unrelated Wikipedia (light green). We further explore the main qualitative findings in the following paragraphs, plotting results on the BLiMP dataset for simplicity. Detailed results on SyntaxGym and CrowS-Pairs are available in Appendix F and A, respectively.

Model acceptability judgements are largely robust across lengths—for unrelated, control prefixes. We first investigate the impact of increasing context length on model acceptability judgement performance. We start with the control case defined in §3, simulating lengthy context windows with no other notable grammatical properties by drawing sentences from Wikipedia, an out-of-distribution text domain for the target MPP datasets. As we increase the context length, acceptability judgement results do not significantly change (Figure 3, long dashed lines), suggesting that LMs, in general, are *very robust* to unrelated changes in their context window. Quantitatively, no main effect of prefix length is significant ($p > 0.2$ for all models) for Wikipedia sentences.

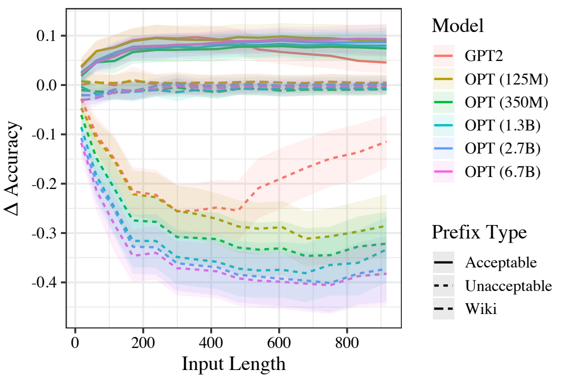


Figure 3: Interaction of length and prefix type on BLiMP (collapsed across match/mismatch). Across all tested models, accuracy improves for acceptable prefixes and worsens for unacceptable ones, as length increases ($p < 10^{-11}$ for all models). Shaded regions are the 95% confidence interval.

The length of the context matters when the prefix is related to the acceptability task. We next investigate the effect of long context on acceptability by drawing prefixes that are in-distribution (from the same MPP dataset). As prefix length grows, model performance on average changes monotonically from baseline accuracy (Figure 2: rising for acceptable sentences, falling for unacceptable ones). When the prefix consists of acceptable sentences (teal, orange) for example, Δ accuracy increases up to 10–20 percentage points for all datasets, and mostly uniformly across all model sizes. However, unacceptable prefixes (purple, red) elicit the opposite effect: Δ accuracy falls as context context length grows (Figure 2, dashed lines).

Scale amplifies this effect only for unacceptable sentences (Figure 3). For example, OPT 6.7B suffers the largest degradation of acceptability task accuracy with increasing length of ungrammatical context, compared to GPT2. Surprisingly, GPT2 recovers some percentage of the degradation on very long sequences, while also showing attenuated the matched gains. We speculate that this effect derives from a relative weakness of GPT2 to learn in-context, as it is trained on markedly less data (8B tokens, as estimated by Warstadt et al.) than models from the OPT family (180B tokens).

Quantitatively, this interaction between prefix length and acceptability is highly significant for all models and evaluations ($p < 0.002$ for all models on BLiMP and SyntaxGym). Overall, we observe length can influence LM's acceptability judgement performance for *in-distribution* contexts, and more so when the contexts contain acceptability violations. One possible driver for these results could be that longer contexts are more conducive to large LMs' in-context learning abilities, and mimic their k-shot learning scenario. This would mean that the length of preceding context matters only insofar as length is a proxy for the number of acceptable (or unacceptable, with an opposite effect) matched prefixes in the context (see §5 for a related analysis).

Matched context impacts acceptability judgements more than mismatched contexts. We now dig into the interaction between length and acceptability, investigating whether the magnitude of the effect is modulated by whether the phenomena are matched or not. In case of BLiMP, the average effect of acceptable prefixes is ≤ 12 Δ accuracy points (Figure 3). However, matched prefixes drive this improvement more ($\Delta \geq 15$) than mismatched ones ($\Delta \leq 5$) (Figure 4, left subfigure). Conversely, while the average effect of unacceptable prefixes is between 30–40 Δ accuracy points (Figure 3), this too is more heavily impacted by the effect of matched prefixes ($50 \leq \Delta \leq 80$) than by mismatched ones ($\Delta \leq 20$) (Figure 4, right subfigure). These effects manifest quantitatively in a three-way interaction between prefix (un-)acceptability, (mis-)match, and length ($p < 0.007$ for all models on BLiMP and SyntaxGym).

The effects of unacceptable prefixes are amplified substantially when they are consistent—i.e., when they violate the grammatical rules (of English) in the same way (matched), as opposed to

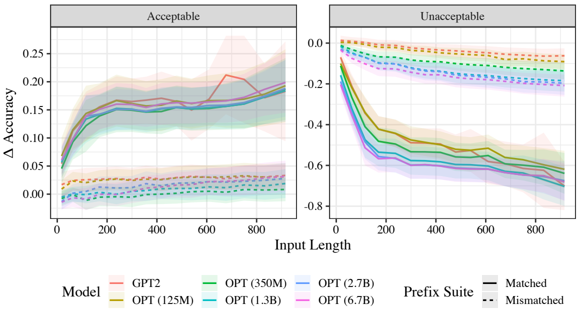


Figure 4: Interaction of length and prefix suite (matched versus mismatched) in two separate prefix types (acceptable/unacceptable) in BLiMP.

in more diverse ways (mismatched).[7] These results could explain why in-context learning ability works: perhaps prepending contexts that are syntactic similarity can help the model *learn* or *unlearn* acceptability at a higher rate.

5 Prefix Similarity Analysis

We have observed that length effects on acceptability judgements are conditional on the similarity between the prefix phenomenon and the test phenomenon. However, we have only analyzed prefixes that are either very similar (i.e. contain predominantly the same abstract syntactic structure as the test sentence, matched prefixing), or are almost entirely unrelated (mismatched prefixing, or unrelated prefixing such as Wikipedia). This leads us to wonder about the nature of the similarity driving our results thus far: are the models responding to the presence of shared syntactic structure in the prefix? Or are they responding to something more shallow and brittle, such as the exact match in sentence templates between the prefix content and the test? If the former is true, we should see a smooth relationship between prefix syntactic similarity and length effects, such that slight changes in the syntactic structure of the prefix content results in similarly slight modulations of length effects. If models are using more shallow template-based comparisons between the prefix content and the test content, we might see a more discontinuous response, in which even small changes to prefix content result in large changes in length effects.

To test this, we narrow our focus to the top 20 BLiMP phenomena which responded most strongly to matched prefixing in our previous analyses.[8] We perform controlled perturbations on each prefix sentence c that preserve the presence of the original syntactic structure, but incorporate mild structural variations or additions. These perturbations increase prefix length and shift the position of certain tokens (e.g., the main verb) in c relative to their counterparts in the test sentence. This enables us to test whether the models are merely learning to associate fragile token-position pairings between the prefix and test sentences, or whether they are relying on relevant abstract syntactic information. We leave the test sentence x unchanged.

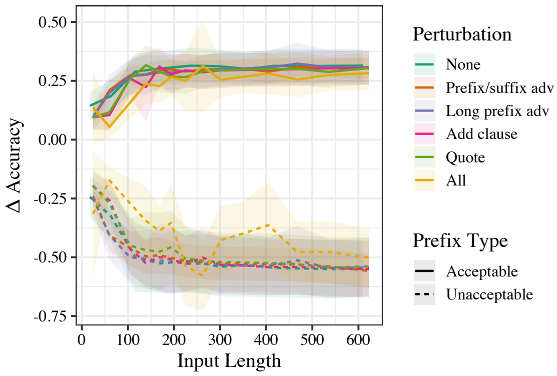


Figure 5: Perturbation analysis on OPT 6.7B, using BLiMP prefixes.

Our perturbations include the following, all of which preserve both the grammaticality and the relevant overarching syntactic structure of the BLiMP phenomena:

- *Prefix/suffix adverbs*: add a single-word sentential adverb to the start or end of the sentence (e.g., “However, c.”).
- *Long prefix adverb*: add an adverbial phrase to the start of the sentence (e.g., “First and foremost, c.”).
- *Add clause*: Add a dependent clause to the start or end of the sentence (e.g., “Regardless of what {NAME} thinks about it, c.”)
- *Quote*: Embed the sentence in a quotation (e.g., “Yesterday, {NAME} said, ‘c.’”).

We also combine all of these strategies into a single large perturbation, referred to as *All*.[9]

[7] Note, however, (i) we assumed that all Wikipedia sentences are acceptable, and (ii) we found that acceptable prefixes have a generally weaker effect on the acceptability task. Were we to test *un*acceptable Wikipedia sentences as well, we might expect a small priming effect.

[8] We selected the phenomena which showed the greatest change in accuracy (averaged across models) between their baseline accuracy and their accuracy after matched prefixing at the greatest lengths tested in the analysis (Appendix Table 1).

[9] We exclude short prefix adverbs from the *All* perturbation in favor of long prefix adverbs. Combining these sometimes

Our findings (Figure 5) show that minor perturbation of the prefixes results in only very minor reductions to length effects, suggesting that matched prefixing effects do not require identically structured prefixes. Increasingly aggressive perturbations result in increasing (if small) reductions to Δ accuracy magnitudes, especially when using fewer prefixes. We correlate Δ accuracies with the mean similarity of prefix sentences before and after a perturbation, where similarity is an ordinal variable assigned to each perturbation based on how many tokens it adds to the sentence; see App. B for details. The Spearman rank-order correlation ($\rho_s = 0.93$, $p < .001$) is significantly positive for acceptable prefixes; it is weaker but still significantly *negative* ($\rho_s = -0.7$, $p < .05$) for unacceptable prefixes. Thus, there is a smooth relationship between prefix similarity and length effects.

This perturbation analysis shows that model judgments are mostly robust to syntactic variations in the prefix content, with a smooth relationship between degrees of syntactic variation and model performance. Appendix D investigates whether these similarity effects can be described in terms of lexical overlap or matches in low-level syntactic features between the prefix and test content; we find no clear relationship between these low-level features and models' acceptability judgment performance. Taken together, these results suggest that the changes we observe in models' acceptability judgments are likely due to an abstract comparison between structural features of the prefix content and test content. In other words, language models are sensitive to latent syntactic features, and the syntactic similarity of the context to the test examples.

6 Discussion

Short and single-sentence inputs may not be representative of language models' true abilities. Our results have implications for interpreting results from MPP benchmark datasets, as these datasets often consist of shorter inputs that are not what many pre-trained language models expect, given that their pre-training procedures often entail packing many sentences into a single training example (Brown et al., 2020; Liu et al., 2019). This strengthens prior findings showing that reformatting train and test inputs in a way that more closely resembles the pre-training setup can boost performance (Hupkes et al., 2020; Newman et al., 2020; Varis and Bojar, 2021; Chada and Natarajan, 2021).

More broadly, our work adds to the literature on prompt sensitivity in pre-trained language models, which found that LMs are sensitive to individual prompts (Kojima et al., 2022), and that the ordering of in-context examples (Lu et al., 2022) can greatly affect model performance. Smaller LMs are also sensitive to the choice of prompt and output verbalizer (Schick and Schütze, 2021a; Gao et al., 2021), and we indeed observe that a variety of model sizes and prefixing strategies elicit prefix sensitivity. To our knowledge, our study is the first to consider structural priming in concert with in-context learning; we have found quantitative, graded effects of structural priming on string probabilities, subject to the length of the context.

Language models are sensitive to latent syntactic features, as well as syntactic similarities across multiple sentences. Our analyses add to a literature that has found that language models are sensitive to more than just lexical or surface-level syntactic features (Warstadt et al., 2020b; Mueller et al., 2022). Indeed, LMs are capable of leveraging abstract syntactic features, and are sensitive to latent syntactic similarities between the context and test examples. Strengthening this finding, we also observe that models are capable of adapting to the structures of both acceptable *and* unacceptable examples: LMs show marked improvements on acceptability tasks when prefixed by matched acceptable sentences, and they also (more substantially) show the opposite behavior—preferring unacceptable sentences—when prefixed by matched unacceptable sentences (§4). This shows that LMs are sensitive enough to sentence acceptability to be able to produce not just systematically grammatical outputs, but also *systematically ungrammatical* outputs. While this is not a practical application, it does demonstrate how well LMs capture this important linguistic feature. Furthermore, our perturbation analysis demonstrated that this two-way adaptation was robust to irrelevant syntactic variations in the context (§5). The present work bolsters the findings of other recent work that only explores this behavior in the grammatical direction (Lampinen, 2022; Sinclair et al., 2022).

Our finding of models' reliance on abstract structural features that are made available in their context can be further strengthened by controlling for lexical exposure (Kim et al., 2022). That is, fu-

results in unacceptable sentences.

ture work can augment our contexts by replacing real lexical items—especially content words—with nonsense words (e.g., *wug*, *dax*, etc.), following recent works (Dasgupta et al., 2022; Misra et al., 2023, *i.a.*). Doing so would maintain the structural features of the context while also more strictly controlling for superficial cues such as lexical overlap or similarity, and would make our conclusions stronger.

7 Conclusion

In this work, we study how robust the acceptability judgements of autoregressive Transformer language models are to manipulations of the context. We find that acceptability judgements are generally robust when the test sentences are preceded by randomly sampled linguistic contexts. However, when the contexts contain syntactic structures closely matching those in the test sentence, that can significantly improve or degrade the models' performance. This effect is amplified as we lengthen the context provided to the model. Our results demonstrate in-context learning in a highly specific way: models are sensitive to granular syntactic properties of the context when making predictions over a target sentence, such that they can be driven to produce both correct and reliably *incorrect* outputs.

Limitations

The prefixes we use are semantically independent from the test sentences, and also semantically implausible when chained together. This is the opposite of what we typically expect in natural language, where sentences follow from some pragmatically licit prior context. While our findings are theoretically relevant to any NLP task that leverages natural language inputs, we may see qualitatively different trends in more naturalistic settings.

Our results are currently limited to English. Certain languages have grammatical features (such as case marking) that could strongly impact on language models' acceptability judgments, and this could affect the trends we have observed. Future work should investigate similar phenomena across languages to ensure that these findings suitably general.

Acknowledgments

We thank the ACL reviewers for their insightful questions and comments. We would also like to thank Marten van Schijndel, Allyson Ettinger, Tiwalayo Eisape, Jennifer Hu, Peng Qian and Alex Warstadt for their feedback and comments on draft versions of this paper.

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*, 25(4):543–557.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. 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 1004–1015, Online. Association for Computational Linguistics.

J. Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18(3):355–387.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Rakesh Chada and Pradeep Natarajan. 2021. FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6090, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noam Chomsky. 1965. *Aspects of the theory of syntax*. MIT Press, Cambridge, MA.

Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*.

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

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, Berlin, Germany. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer Language Models without Positional Encodings Still Learn Positional Information. *ArXiv preprint*, abs/2203.16634.

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.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Denis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2022. State-of-the-art generalisation research in nlp: a taxonomy and review. *arXiv preprint arXiv:2210.03050*.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

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.

Michael P. Kaschak and Arthur M. Glenberg. 2004. This construction needs learned. *Journal of Experimental Psychology: General*, 133(3):450.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Najoung Kim, Tal Linzen, and Paul Smolensky. 2022. Uncontrolled lexical exposure leads to overestimation of compositional generalization in pretrained models. *arXiv preprint arXiv:2212.10769*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*.

Andrew Lampinen. 2022. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *arXiv preprint arXiv:2210.15303*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

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.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring BERT's sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635, Online. Association for Computational Linguistics.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

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. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. 2020. The EOS decision and length extrapolation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 276–291, Online. Association for Computational Linguistics.

Martin J. Pickering and Victor S. Ferreira. 2008. Structural priming: A critical review. *Psychological Bulletin*, 134(3):427–459.

Martin J Pickering and Simon Garrod. 2017. *Priming and Language Change*, pages 173–90. Cambridge University Press.

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.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Carson T. Schütze. 1996. *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. University of Chicago Press, Chicago, IL.

Arabella Sinclair, Jaap Jumelet, Willem Zuidema, and Raquel Fernández. 2022. Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050.

Koustuv Sinha, Amirhossein Kazemnejad, Siva Reddy, Joelle Pineau, Dieuwke Hupkes, and Adina Williams. 2022. The curious case of absolute position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4449–4472, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marten van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.

Dusan Varis and Ondřej Bojar. 2021. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

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

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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

Yuxue Cher Yang and Andrea Stocco. 2019. Syntactic priming depends on procedural, reward-based computations: evidence from experimental data and a computational model. In *Proceedings of the 17th International Conference on Cognitive Modeling*, pages 307–313.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Fairness Analysis

Datasets. CrowS-Pairs (Nangia et al., 2020) contains 1508 sentence pairs denoting stereotypes about nine types of demographics, including gender, age, nationality, etc. CrowS-Pairs differs from BLiMP and SyntaxGym in construction, since it was crowdsourced using untrained English speakers from Amazon Mechanical Turk. Despite this difference, the resulting test pair sentences still only minimally differ from each other (except for some instances where more than a few tokens differ due to annotation noise, see Blodgett et al. 2021 for more discussion). Thus, we can leverage CrowS-Pairs in the similar MPP paradigm and test whether our results are specific to syntactic evaluation.

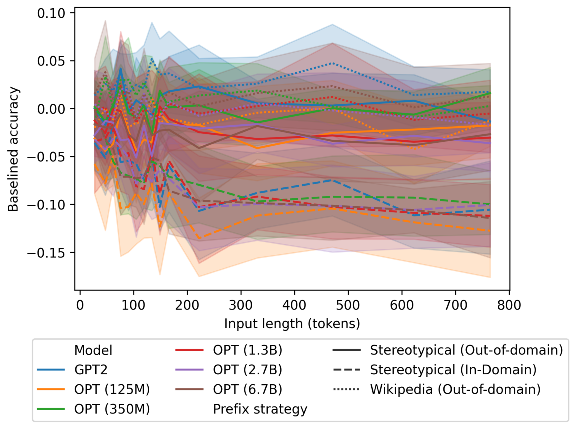


Figure 6: Effects of length and stereotypical prefixes per model. Shaded regions indicate the 95% confidence intervals across 9 demographics.

Similar to the approach in SyntaxGym, (Nangia et al., 2020) propose to measure fairness in masked language models by focusing only on the tokens which differ, computing the pseudo-log-likelihood of the sentences conditioned on those tokens. To maximize the comparability of the CrowS-Pairs results with our results on BLiMP/SyntaxGym, we compute the conditional log-likelihood, as described in §3. We then compute the acceptability of each test pair as described in equation 2, where we recode the definitions of *unacceptable* and *acceptable* items to *stereotypical* and *antistereotypical*, in-line with the definitions in this dataset. An ideal, fair model would show no special preference towards stereotypical sentences.

Method. We construct contexts using the same approach described in §3. In lieu of *phenomena* in SyntaxGym/BLiMP, Crows-Pairs dataset provides test pairs over multiple *demographies*. Thus, for a given test example, we construct *matched* contexts by sampling from the same demographic cohort the test pair belongs to, and conversely sample from different demographic subset to construct *mismatched* context. We re-use the same control experiment setup, i.e. sampling from Wikipedia for irrelevant contexts.

Analysis. Figure 6 compares stereotypical contexts from mismatched and matched demographics across models of varying sizes. The results show that mismatched contexts don't show any significant impact on the fairness scores. Across all model

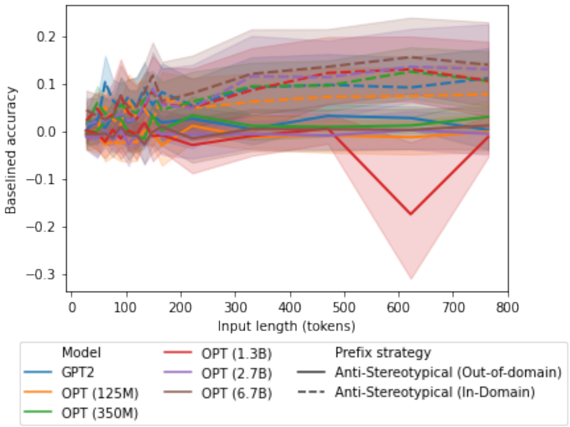


Figure 7: Effects of length and anti-stereotypical prefixes per model. Shaded regions indicate the 95% confidence intervals across 9 demographics.

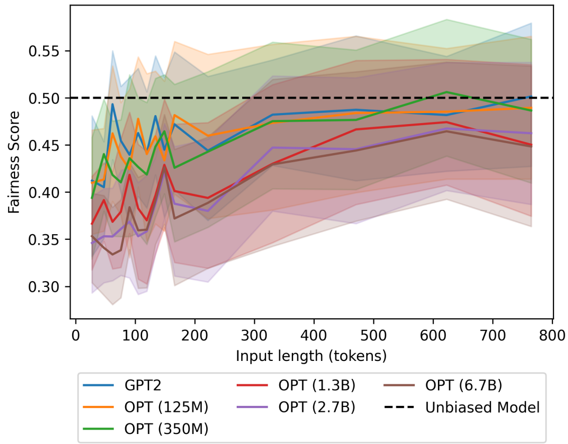


Figure 8: Effects of length and antistereotypical matched prefixes per model. Shaded regions indicate the 95% confidence intervals across 9 demographics.

sizes, we see a score decrease when prefixed with matched stereotypical context. Meanwhile, Figure 7 shows that prefixing with an antistereotypical context improves the fairness scores. This raises the question, can we prime models with antistereotypical contexts to reduce stereotypical bias? Figure 8 shows that prefixing with an antistereotypical matched context can enable models to reach the ideal score of an unbiased model, and even surpass it (i.e, making the model biased in the other direction). However, it is worth noting that this does not necessarily indicate that a model is unbiased, as there is significant variation between demographics, and more detailed examination is needed to evaluate the effects per demographic cohort.

B Prefix Similarity Analysis

Here, we provide more detail to support the experiment in §5. Specifically, we present the exact numbers we use to compute the rank-order correlation coefficients, and describe the implications of this finding for future work.

To compute the rank-order correlation, we first obtain mean accuracies across the 20 BLiMP phenomena that respond most strongly to matched prefixing. We do this for each perturbation strategy, as well as the non-perturbed matched prefixes. We then take the mean across all prefixing lengths for OPT 6.7B (i.e., we convert each line in Figure 5 into a single number by taking the mean along the x-axis). This yields a metric that approximately captures how much of a priming effect a given prefixing strategy has for this model; we use this as our dependent variable.

The independent variable is the strength of the perturbation prefix. It is difficult to define how strong a given perturbation is, as there are different notions of linguistic similarity that can be contradictory; for instance, embedding a sentence c into a quote, as in “Yesterday, Sarah said ‘c’”, does not add many lexical items to the sentence, but it significantly modifies the syntactic structure of the sentence. In our case, we simply measure the token F_1 score between the original prefix sentence and a perturbed prefix sentence; this metric captures the token similarity between the original and perturbed sentences. Future work could consider more sophisticated similarity metrics, such as syntactic or semantic similarities.

We summarize these results in Figure 9. Note the highly monotonic relationship when using acceptable prefixes, and the similarly (but slightly less) monotonic relationship with unacceptable prefixes. This visually displays the strong correlations we found in §5.

Why are language models being more primeable with longer contexts given more similar prefixes? Perhaps models can determine whether tokens are meaningfully similar between multiple sentences in the same context; this would be expected given the implications of the distributional hypothesis. Alternatively, the model could be effective at relating tokens that are similar in the pre-training corpus, as long as their positions are within some limited range of each other. Finally, perhaps the model is simply effective at ignoring (for example) adverbs and adjuncts that are semantically or

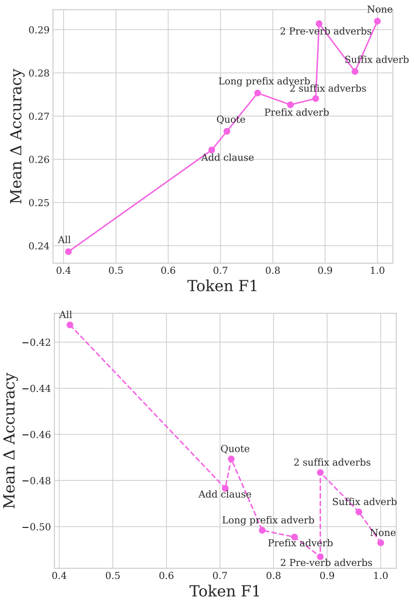


Figure 9: Token similarity before and after perturbing the prefixes vs. Δ accuracy across BLiMP phenomena for OPT 6.7B. We show Δ accuracies using acceptable (top) and unacceptable (bottom) prefixes.

syntactically irrelevant, and thus otherwise views the perturbed prefixes and test sentences as more or less structurally identical. Our results cannot currently disambiguate between these possibilities, but future work could investigate perturbed prefixing in significantly more depth to better understand *why* we observe these effects and correlations.

C Metric correlation analysis

To what degree can the priming effects discussed in this paper reveal facts about a model's capacities not already evident from existing MPP evaluations? To test this, we evaluated (for every model and dataset) the correlation between a model's baseline performance and its performance with a maximal amount of acceptable or unacceptable matched prefixes. For example, we evaluated the correlation between a model's accuracy in un-prefixed BLiMP phenomena, and its accuracy on each of the phenomena after prefixing with the maximal amount of possible unacceptable prefixes (start vs. end of dashed purple line in Figure 2).

Figure 10 and Figure 11 show the results of this analysis. A single point in any of these scatterplots indicates the relationship between a particular model's performance on a particular suite at baseline (no prefix, x-axis) and its performance with a maximal-length prefix, either acceptable (Figure 10) or unacceptable (Figure 11; y-axis). If our prefixing results reveal facts about model capacities not already present in MPP evaluations, then we should see substantial variance in the y-axis not explained by the x-axis on these plots. This is apparent in most of the plots, especially in the BLiMP evaluations (leftmost plots).

We also see variation among models: GPT2 has a prefixing response which is relatively predictable from its baseline performance (correlation with acceptable prefixing effect, mean across datasets: $r = 0.85$; unacceptable: $r = 0.79$). In contrast, OPT 2.7B is far less predictable in its prefixing response (correlation with acceptable prefixing effect, mean across datasets: $r = 0.59$; unacceptable: $r = 0.52$).

Overall, this analysis suggests that there are nontrivial variations in the way that models respond to these prefixing interventions which is not captured by models' baseline performance on matched stimuli. This suggests that prefixing reveals new aspects of model capacity not exactly captured by existing MPP evaluations.

D BLiMP Phenomenon Similarities

Length effects are conditional on the similarity of the prefix to the target BLiMP phenomenon. Does some specific kind of similarity (e.g., syntactic or lexical similarity) explain length effects? Perhaps the prefix is syntactically priming the model for the target sentence (Sinclair et al., 2022), in which case we would expect the syntactic similarity of the sentences to correlate smoothly with accuracy when using grammatical prefixes. Another possibility is that a more spurious feature—such as lexical overlap—is responsible (Misra et al., 2020; Kassner and Schütze, 2020). To test this, we can correlate syntactic similarity and lexical overlap with accuracies on each example.

To measure lexical overlap, we use F_1 scores to measure how many tokens[10] in the prefix and test sentences are shared. To approximate syntactic overlap, we can compute the F_1 score over *dependency labels* in two sentences, rather than across tokens. If multiple prefix sentences are present, we can take the mean similarity with the target sentence across prefixes. Then, we compute the point-biserial correlation[11] (ρ_p) between the sim-

[10] We tokenize the inputs using GPT2's tokenizer before computing overlap.

[11] The point-biserial correlation coefficient measures the strength of the relationship between a continuous variable (e.g., our overlap metrics) and a binary variable (accuracy on

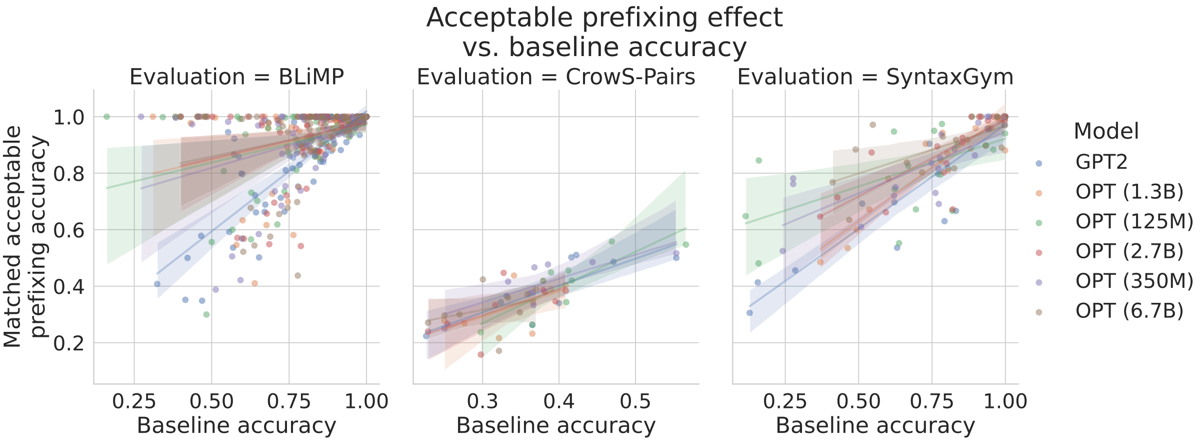


Figure 10: Correlation between accuracy with maximum-length acceptable prefix and baseline accuracy, for each model and dataset.

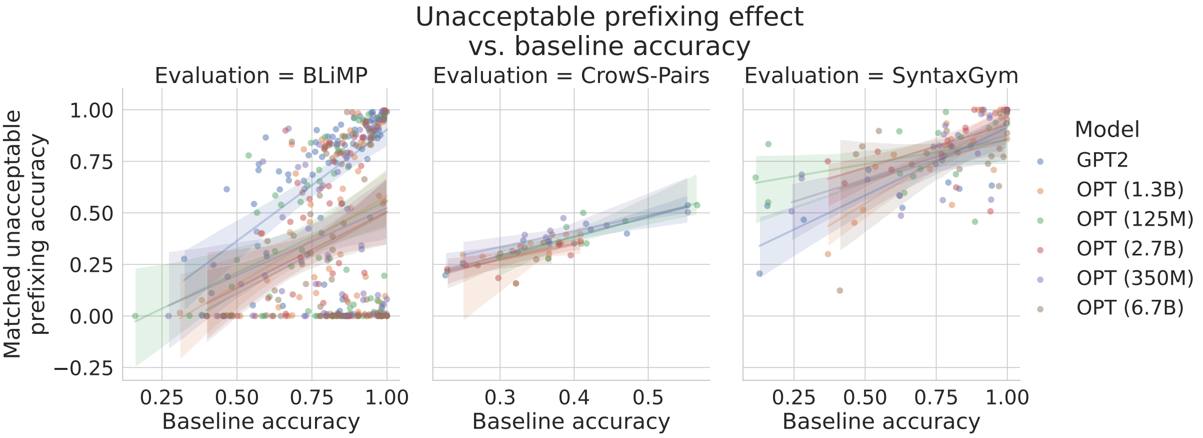


Figure 11: Correlation between accuracy with maximum-length unacceptable prefix and baseline accuracy, for each model and dataset.

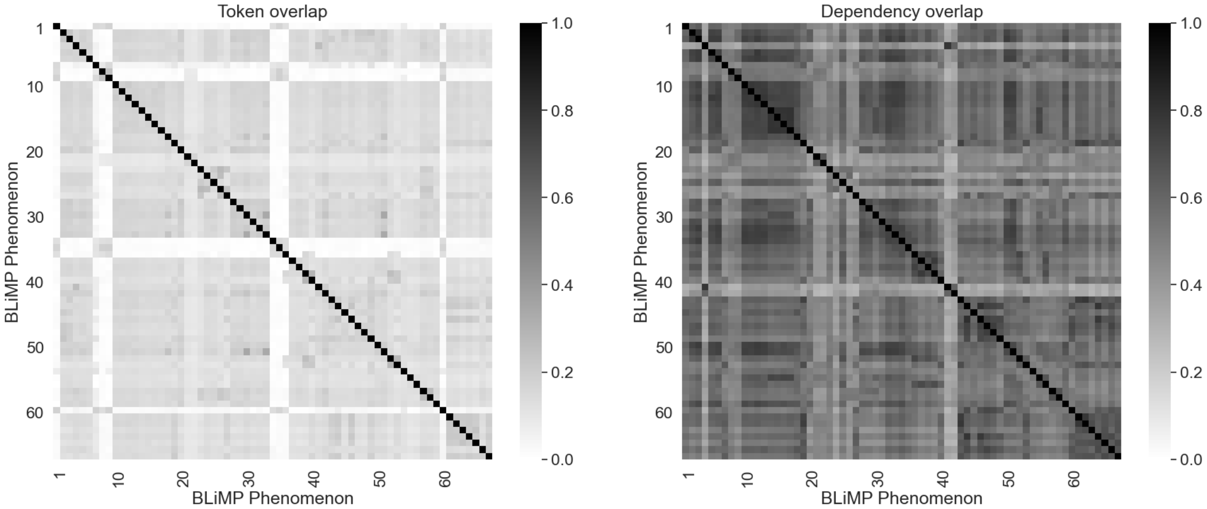


Figure 12: Token overlap (left) and dependency overlap (right) across BLiMP phenomena. We compute these using a sample of 10,000 sentences from the target phenomenon and from the prefix phenomenon. The phenomena are ordered alphabetically.

ilarity metric and accuracy on a given example, averaging similarities across prefix sentences. We compute the correlation separately for each model size and each prefixing strategy. Note that we only use grammatical prefixes; thus, we expect positive correlations if priming explains the length effects.

However, this instance-level analysis could be confounded by the mixture of various phenomena in the prefixes. The model could be sensitive to sentences from certain phenomena more than others, or the varying lengths of sentences from each phenomenon. To more specifically measure whether priming can explain our findings, we focused on BLiMP and prefixed sentences from one phenomenon at a time with a given test phenomenon; in other words, we sample *mismatched* prefixes, but controlling which phenomenon we sample from. Using this approach, we can capture how structurally similar each BLiMP phenomenon is with each other BLiMP phenomenon, and how this correlates with accuracies.

Here, we present the lexical and syntactic similarity across each pair of BLiMP phenomena (Figure 12).[12] We find very low and non-significant correlations with dependency overlap and token overlap ($\rho_p < 0.05$, $p > 0.1$) regardless of prefixing strategy or model size. This could be evidence that the model is more sensitive to the length of the prefixes than any notion of syntactic or lexical similarity on this task. These are computed across each prefix and test phenomenon using a sample of 10,000 test sentences and 10,000 prefix sentences for each point in the confusion matrix. We find that dependency overlap is generally higher than token overlap across inputs, perhaps unsurprisingly given that the size of the set of possible dependency labels is much smaller than the size of the set of possible tokens in a given sentence.

We next try correlating these values with accuracies on each BLiMP phenomenon as a function of these phenomenon-level similarity metrics. Accuracies with prefixes (and changes in accuracies after after prefixing) for GPT2 are presented in Figure 13. Essentially, we are now measuring how similar the trends are across a similarity confusion matrix and an accuracy confusion matrix. As we are now measuring similarity across continuous variables, we compute the Spearman correlation (ρ_s). We find that correlations here are a bit stronger than when we mix mismatched prefixes ($\rho_s = 0.11$ for dependency overlap, and $\rho_s = 0.18$ for token overlap, $p < 0.001$ for both). While the magnitude of the correlations is very low, these are still significant. Thus, there is some relationship between the similarity of the prefix and test sentence with accuracy, but the relationship tends to be weak. Also, lexical overlap seems to be more strongly predictive of accuracies than structural similarities, indicating that the model may indeed be more sensitive to spurious lexical similarities than any deeper abstract notion of syntactic similarity between a prefix and the test sentence. Nonetheless, this is still preliminary evidence that priming effects do not explain much of the accuracy trends we observe with prefixing; instead, perhaps length itself makes a stronger difference than any specific notion of similarity between the prefix and test sentence.

This is preliminary evidence that **lexical overlap and low-level syntactic similarity effects *partially* explain accuracy increases with BLiMP prefixing, but most of the trends we observe cannot be explained by these effects alone.** Perhaps this is because the model is more sensitive to multiple similarities simultaneously than any one isolated type of similarity. Or, perhaps models are sensitive to some other latent feature that we did not analyze. Nonetheless, it is difficult to draw strong conclusions from the lack of a strong correlation, and correlations alone cannot causally implicate similarities in explaining our findings. Perhaps future work could disambiguate the relationship between these factors using causal methods.

D.1 Suite-by-suite prefixing performance

Figure 14 shows GPT2's improvement in prediction accuracy on different SyntaxGym test suites (rows) after drawing as many acceptable prefix sentences as possible from another SyntaxGym test suite (columns). The values are a percentage increase in prediction accuracy, relative to GPT2's baseline performance with no additional context. We see a substantial diversity in how different suites respond to prefixing of acceptable sentences. Some suites, such as an NPI licensing suite (`npi_src_any`) and a filler-gap dependency suite (`fgd_subject`), show across-the-board improvements in response to any prefixing at all. The suites labeled `reflexive_*_fem`, which test under-

an individual example).

[12] For visual conciseness across confusion matrices, we use indices rather than individual phenomenon names. For each confusion matrix in Figures 12 and 13, all phenomena are presented in alphabetical order.

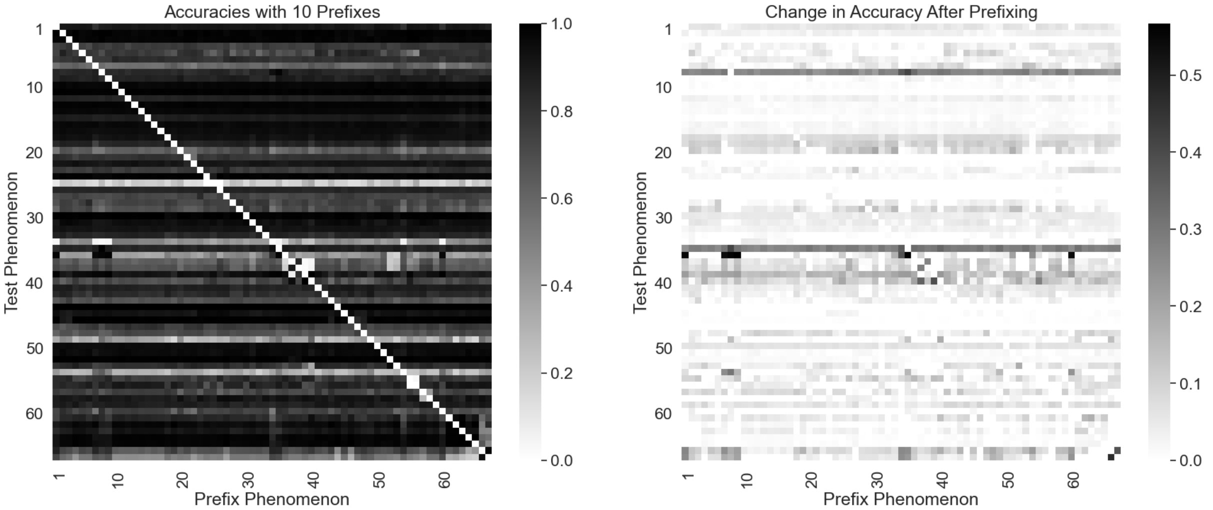


Figure 13: Accuracies for GPT2 on individual BLiMP phenomena after prefixing 10 sentences from a single BLiMP phenomenon (left). Change in accuracy from no prefix to 10 prefixes on each BLiMP phenomenon (right). We exclude the diagonal in both cases, as we are interested in *mismatched* prefixing effects.

standing of feminine reflexive anaphor agreement, demonstrate interesting unstable behavior: GPT2's predictions degrade when these particular tests are preceded by grammatical sentences containing masculine reflexive anaphors (see e.g. the blue boxes in the row labeled `reflexive_orc_fem`, but the same predictions are facilitated when preceded by feminine reflexive anaphors.

We also provide a snapshot of the top 10 suites in BLiMP (Warstadt et al., 2020a) which get the best and worst changes in accuracy (Δ Accuracy), when primed with acceptable (Table 1) and unacceptable prefixes (Table 2) respectively.

E Margin Analysis

How confident are LMs as input length increases? The results on length priming indicates that longer matched acceptable prefixes tend to induce better acceptability judgements to the target model. However, investigating the accuracies as computed in Equation 2 alone does not fully explain the nuances of the model confidence. To understand how model confidence values themselves differ in acceptable/unacceptable target sentences, we plot and investigate the perplexity margins in Figure 15. Specifically, we compute the difference in the model perplexities δ for each acceptable/unacceptable pair:

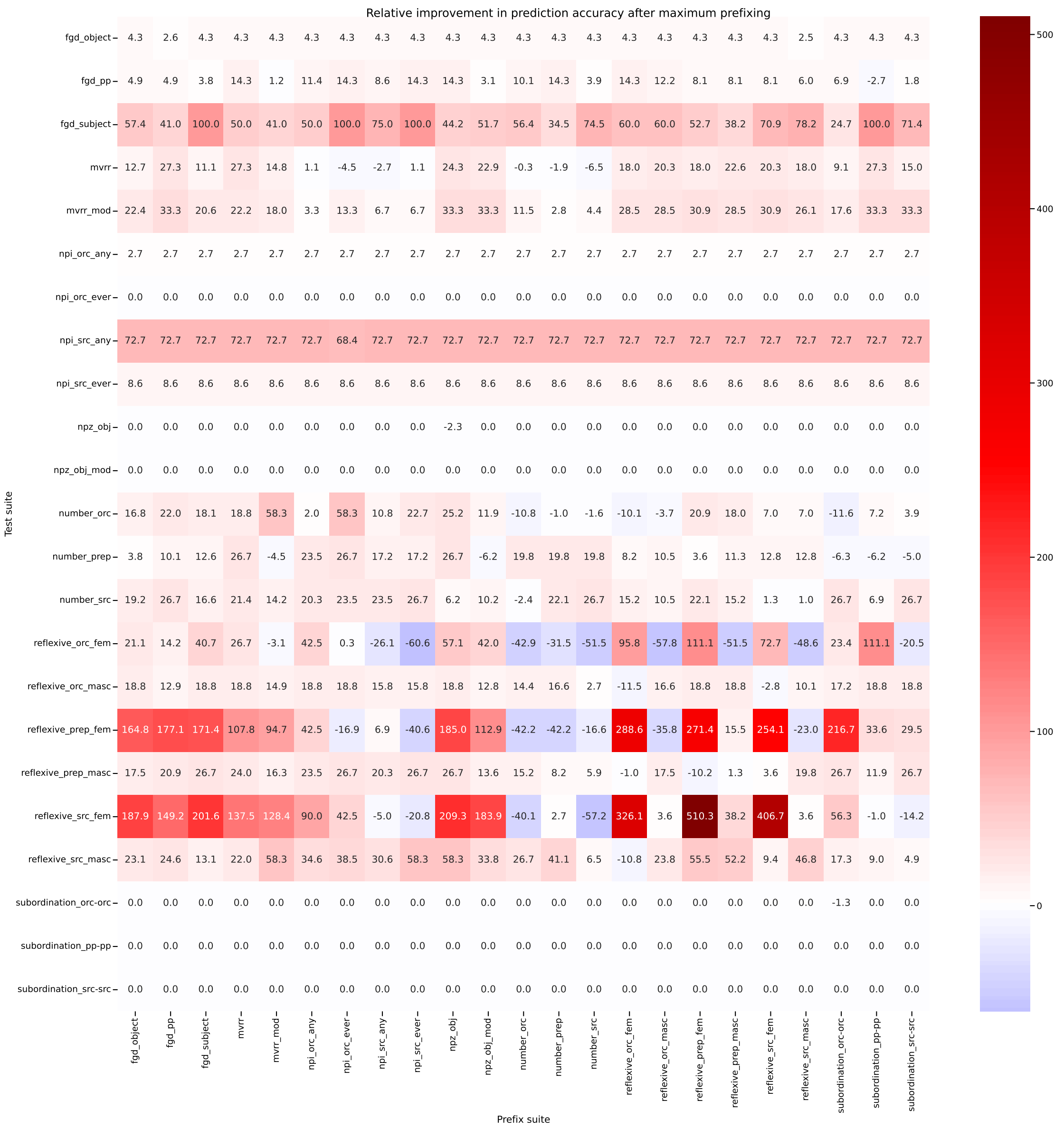
$$\delta(x_i, \hat{x_i}) = \mathtt{ppl}(x_i) - \mathtt{ppl}(\hat{x_i}), \tag{3}$$

We observe the margins on BLiMP for a candidate model, OPT 6.7B in Figure 15, for grammatical, ungrammatical and Wikipedia prefixes. For all cases, δ starts from a high value for short sequences, and approaches zero as the context length increases. There is a marked difference in δ values compared to Wikipedia and BLiMP prefixes: Wikipedia prefixes appear to display a high value, suggesting high surprisals. The average δ for Wikipedia also remains higher than the baseline value (without any priming), while δ is significantly lower for BLiMP prefixes. This behavior potentially explains why we observe almost no change in the accuracy of Wikipedia prefixes, as the margin remains high and stable with increasing length of tokens.

Within matched prefixes, we observe the δ to be significantly lower for unacceptable prefixes compared to the acceptable contexts, and it reduces with length. This behavior partially explains why we observe the trend of sharp decrease in acceptability accuracy for matched unacceptable prefixes, as the monotonically decreasing δ flips the acceptability judgement associations.

F SyntaxGym Results

We run our prefixing evaluations for 23 of the 34 SyntaxGym evaluations whose prediction structures are compatible with this paper's evaluation setup – that is, where model success is a function of one or more differences in surprisal measured between two experimental conditions. These applicable suites are shown in the axes of Figure 14. In contrast to BLiMP, model surprisal is measured only at a **critical region**, at which differing content



	fgd_object	fgd_pp	fgd_subject	mvrr	mvrr_mod	npi_orc_any	npi_orc_ever	npi_src_any	npi_src_ever	npz_obj	npz_obj_mod	number_orc	number_prep	number_src	reflexive_orc_fem	reflexive_orc_masc	reflexive_prep_fem	reflexive_prep_masc	reflexive_src_fem	reflexive_src_masc	subordination_orc-orc	subordination_pp-pp	subordination_src-src
fgd_object	4.3	2.6	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3	2.5	4.3	4.3	4.3
fgd_pp	4.9	4.9	3.8	14.3	1.2	11.4	14.3	8.6	14.3	14.3	3.1	10.1	14.3	3.9	14.3	12.2	8.1	8.1	8.1	6.0	6.9	-2.7	1.8
fgd_subject	57.4	41.0	100.0	50.0	41.0	50.0	100.0	75.0	100.0	44.2	51.7	56.4	34.5	74.5	60.0	60.0	52.7	38.2	70.9	78.2	24.7	100.0	71.4
mvrr	12.7	27.3	11.1	27.3	14.8	1.1	-4.5	-2.7	1.1	24.3	22.9	-0.3	-1.9	-6.5	18.0	20.3	18.0	22.6	20.3	18.0	9.1	27.3	15.0
mvrr_mod	22.4	33.3	20.6	22.2	18.0	3.3	13.3	6.7	6.7	33.3	33.3	11.5	2.8	4.4	28.5	28.5	30.9	28.5	30.9	26.1	17.6	33.3	33.3
npi_orc_any	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7
npi_orc_ever	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
npi_src_any	72.7	72.7	72.7	72.7	72.7	72.7	68.4	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7
npi_src_ever	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6	8.6
npz_obj	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
npz_obj_mod	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
number_orc	16.8	22.0	18.1	18.8	58.3	2.0	58.3	10.8	22.7	25.2	11.9	-10.8	-1.0	-1.6	-10.1	-3.7	20.9	18.0	7.0	7.0	-11.6	7.2	3.9
number_prep	3.8	10.1	12.6	26.7	-4.5	23.5	26.7	17.2	17.2	26.7	-6.2	19.8	19.8	19.8	8.2	10.5	3.6	11.3	12.8	12.8	-6.3	-6.2	-5.0
number_src	19.2	26.7	16.6	21.4	14.2	20.3	23.5	23.5	26.7	6.2	10.2	-2.4	22.1	26.7	15.2	10.5	22.1	15.2	1.3	1.0	26.7	6.9	26.7
reflexive_orc_fem	21.1	14.2	40.7	26.7	-3.1	42.5	0.3	-26.1	-60.6	57.1	42.0	-42.9	-31.5	-51.5	95.8	-57.8	111.1	-51.5	72.7	-48.6	23.4	111.1	-20.5
reflexive_orc_masc	18.8	12.9	18.8	18.8	14.9	18.8	18.8	15.8	15.8	18.8	12.8	14.4	16.6	2.7	-11.5	16.6	18.8	18.8	-2.8	10.1	17.2	18.8	18.8
reflexive_prep_fem	164.8	177.1	171.4	107.8	94.7	42.5	-16.9	6.9	-40.6	185.0	112.9	-42.2	-42.2	-16.6	288.6	-35.8	271.4	15.5	254.1	-23.0	216.7	33.6	29.5
reflexive_prep_masc	17.5	20.9	26.7	24.0	16.3	23.5	26.7	20.3	26.7	26.7	13.6	15.2	8.2	5.9	-1.0	17.5	-10.2	1.3	3.6	19.8	26.7	11.9	26.7
reflexive_src_fem	187.9	149.2	201.6	137.5	128.4	90.0	42.5	-5.0	-20.8	209.3	183.9	-40.1	2.7	-57.2	326.1	3.6	510.3	38.2	406.7	3.6	56.3	-1.0	-14.2
reflexive_src_masc	23.1	24.6	13.1	22.0	58.3	34.6	38.5	30.6	58.3	58.3	33.8	26.7	41.1	6.5	-10.8	23.8	55.5	52.2	9.4	46.8	17.3	9.0	4.9
subordination_orc-orc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.3	0.0	0.0
subordination_pp-pp	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
subordination_src-src	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 14: Relative improvement (in percentage points) in accuracy on SyntaxGym test suite evaluations (rows) after prefixing with sentences from other SyntaxGym test suites (columns) for GPT2.

Phenomena	GPT2	OPT 125M	OPT 350M	OPT 1.3B	OPT 2.7B	OPT 6.7B	Mean Δ
principle_A_reconstruction	0.528 (0.13)	0.62 (0.1)	0.699 (0.1)	0.599 (0.06)	0.585 (0.05)	0.585 (0.05)	0.603 (0.05)
existential_there_quantifiers_2	0.322 (0.07)	0.827 (0.04)	0.528 (0.04)	0.683 (0.02)	0.538 (0.03)	0.538 (0.02)	0.573 (0.15)
sentential_subject_island	0.58 (0.19)	0.556 (0.14)	0.536 (0.18)	0.491 (0.11)	0.402 (0.1)	0.48 (0.11)	0.507 (0.06)
wh_vs_that_with_gap_long_distance	0.457 (0.13)	0.48 (0.12)	0.465 (0.14)	0.481 (0.17)	0.446 (0.15)	0.514 (0.21)	0.474 (0.02)
matrix_question_npi_licensor_present	0.566 (0.01)	0.49 (0.03)	0.477 (0.03)	0.357 (0.01)	0.307 (0.01)	0.358 (0.01)	0.426 (0.09)
wh_vs_that_with_gap	0.353 (0.09)	0.376 (0.07)	0.394 (0.09)	0.435 (0.12)	0.447 (0.13)	0.468 (0.14)	0.412 (0.04)
left_branch_island_echo_question	0.443 (0.21)	0.462 (0.17)	0.357 (0.17)	0.359 (0.14)	0.361 (0.12)	0.344 (0.1)	0.388 (0.05)
only_npi_scope	0.38 (0.05)	0.511 (0.02)	0.198 (0.02)	0.321 (0.02)	0.375 (0.02)	0.347 (0.02)	0.355 (0.09)
npi_present_1	0.327 (0.1)	0.337 (0.11)	0.236 (0.09)	0.267 (0.07)	0.305 (0.09)	0.352 (0.08)	0.304 (0.04)
complex_NP_island	0.316 (0.12)	0.271 (0.09)	0.264 (0.1)	0.241 (0.1)	0.274 (0.09)	0.356 (0.1)	0.287 (0.04)

Table 1: Top 10 phenomena in BLiMP (Warstadt et al., 2020a) with largest average Δ increase in acceptability over the full context window, when primed with acceptable prefixes. The numbers in parenthesis reflect the standard deviation over length of the context.

Phenomena	GPT2	OPT 125M	OPT 350M	OPT 1.3B	OPT 2.7B	OPT 6.7B	Mean Δ
only_npi_licensor_present	-0.693 (0.21)	-0.726 (0.16)	-0.934 (0.19)	-0.953 (0.14)	-0.945 (0.1)	-0.961 (0.07)	-0.869 (0.11)
existential_there_quantifiers_1	-0.783 (0.27)	-0.871 (0.21)	-0.869 (0.21)	-0.856 (0.2)	-0.911 (0.21)	-0.906 (0.21)	-0.866 (0.04)
principle_A_case_1	-0.782 (0.42)	-0.863 (0.35)	-0.813 (0.33)	-0.871 (0.34)	-0.867 (0.35)	-0.872 (0.34)	-0.845 (0.03)
superlative_quantifiers_2	-0.817 (0.1)	-0.822 (0.07)	-0.847 (0.06)	-0.832 (0.05)	-0.862 (0.05)	-0.845 (0.04)	-0.837 (0.02)
sentential_negation_npi_licensor_present	-0.637 (0.25)	-0.733 (0.23)	-0.882 (0.23)	-0.907 (0.24)	-0.904 (0.22)	-0.911 (0.22)	-0.829 (0.11)
wh_questions_subject_gap	-0.731 (0.32)	-0.811 (0.27)	-0.804 (0.29)	-0.839 (0.28)	-0.837 (0.27)	-0.832 (0.27)	-0.809 (0.04)
wh_vs_that_no_gap_long_distance	-0.715 (0.33)	-0.806 (0.24)	-0.742 (0.23)	-0.806 (0.24)	-0.852 (0.25)	-0.889 (0.25)	-0.801 (0.06)
wh_questions_subject_gap_long_distance	-0.782 (0.24)	-0.802 (0.22)	-0.781 (0.22)	-0.828 (0.22)	-0.784 (0.23)	-0.833 (0.24)	-0.801 (0.02)
superlative_quantifiers_1	-0.685 (0.15)	-0.746 (0.06)	-0.832 (0.07)	-0.836 (0.11)	-0.849 (0.11)	-0.806 (0.07)	-0.792 (0.06)
irregular_past_participle_adjectives	-0.671 (0.33)	-0.788 (0.23)	-0.838 (0.24)	-0.834 (0.23)	-0.786 (0.22)	-0.829 (0.23)	-0.791 (0.06)

Table 2: Top 10 phenomena in BLiMP (Warstadt et al., 2020a) with largest average Δ decrease in acceptability over the full context window, when primed with unacceptable prefixes. The numbers in parenthesis reflect the standard deviation over length of the context.

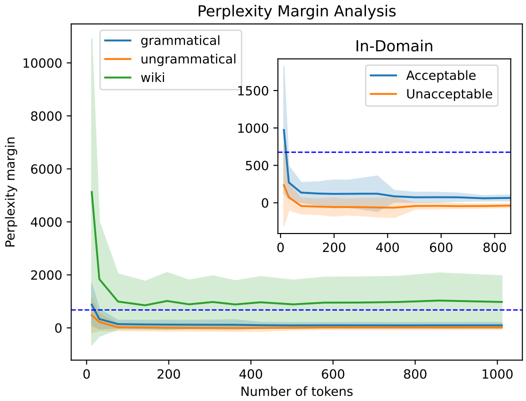


Figure 15: Perplexity margins of Grammatical, Ungrammatical and Wikipedia prefixes on BLiMP for OPT 6.7B model. The dashed lines represent the mean margin of the baseline without any context.

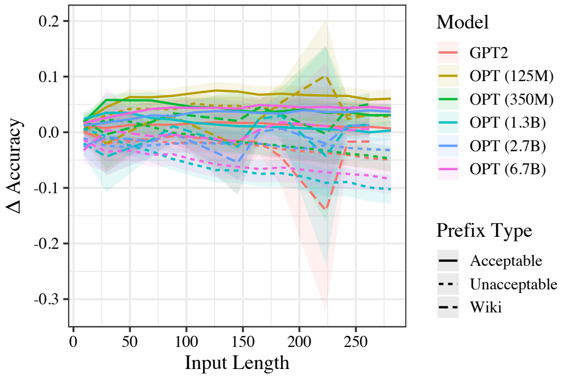


Figure 16: Interaction of length and prefix type on SyntaxGym (collapsed across match/mismatch). Shaded regions are the 95% confidence interval. Analogous to Figure 3 in the main text.

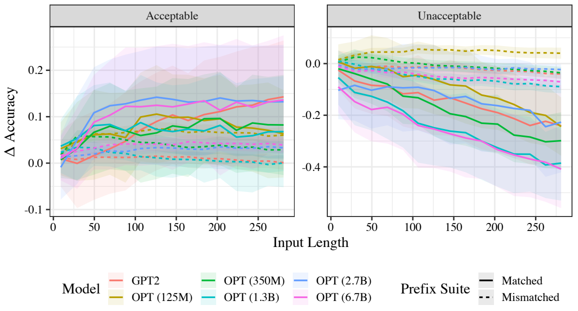


Figure 17: Interaction of length and prefix suite (matched versus mismatched) in two separate prefix types (acceptable/unacceptable) in SyntaxGym. Analogous to Figure 4 in the main text.

between conditions render minimal-pair sentences grammatical or ungrammatical. For example, the `number_prep` suite measures the surprisal difference at the underlined critical region between the following four conditions:

1. The farmer near the clerks knows many people.

2. * The farmer near the clerk know many people.

3. * The farmers near the clerk knows many people.

4. The farmers near the clerk know many people.

In this example test suite, model surprisals for the word *knows* in sentence 3 must be higher than in sentence 1, and surprisals for the word *know* must be higher in sentence 2 than in sentence 4. The full list of included suites is visible in Figure 14. Additional plots for SyntaxGym, analogous to Figure 3 and Figure 4, are provided at Figure 16 and Figure 17.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section after the Conclusion.

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Used: yes (Section 3) Created: no.

☑ B1. Did you cite the creators of artifacts you used?
3

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.