Evil twins are not that evil: Qualitative insights into machine-generated prompts

Nathanaël Carraz Rakotonirina[*1], **Corentin Kervadec**[*1], **Francesca Franzon**[1], **Marco Baroni**[1,2]
[1]Universitat Pompeu Fabra, Barcelona
[2]ICREA, Barcelona

Abstract

It has been widely observed that language models (*LMs*) respond in predictable ways to algorithmically generated prompts that are seemingly unintelligible. This is both a sign that we lack a full understanding of how LMs work, and a practical challenge, because opaqueness can be exploited for harmful uses of LMs, such as jailbreaking. We present the first thorough analysis of opaque machine-generated prompts, or *autoprompts*, pertaining to 6 LMs of different sizes and families. We find that machine-generated prompts are characterized by a last token that is often intelligible and strongly affects the generation. A small but consistent proportion of the previous tokens are prunable, probably appearing in the prompt as a by-product of the fact that the optimization process fixes the number of tokens. The remaining tokens fall into two categories: filler tokens, which can be replaced with semantically unrelated substitutes, and keywords, that tend to have at least a loose semantic relation with the generation, although they do not engage in well-formed syntactic relations with it. Additionally, human experts can reliably identify the most influential tokens in an autoprompt *a posteriori*, suggesting these prompts are not entirely opaque. Finally, some of the ablations we applied to autoprompts yield similar effects in natural language inputs, suggesting that autoprompts emerge naturally from the way LMs process linguistic inputs in general.

1 Introduction

An intriguing property of language models (*LMs*) is that they respond in predictable ways to machine-generated prompts (henceforth, *autoprompts*)[1] that are unintelligible to humans. Shin et al. (2020) first showed that autoprompts can outperform human-crafted prompts on various tasks. More worryingly, Wallace et al. (2019) and others have shown that they can be used in adversarial attacks making models, including latest-generation aligned LMs, behave in undesirable ways (e.g., Zou et al., 2023; Geiping et al., 2024). We present here the first thorough qualitative analysis of autoprompts. We discover that, despite the superficial impression of opacity they convey, they can to a significant extent be explained in terms of a few general observations (illustrated in Figure 1): (1) in autoregressive models, the last token of a prompt has a disproportionate role in generating the continuation, and this last token is both very important and often transparent in autoprompts; (2) several tokens contributing to the opaqueness of autoprompts are simply ignored by the model; (3) the non-final elements that are actually influencing generation might do so in two ways: interchangeable tokens acting as fillers, and more semantically coherent keywords tokens. As we will see, these factors are also at play when LMs are fed natural-language sequences, suggesting that they are core properties of how LMs process linguistic strings.

From a theoretical point of view, our study offers new insights into LM language processing in general. From a practical point of view, it highlights which aspects of LMs we should pay attention to, if we want to make them more robust to harmful autoprompts (or, conversely, to develop more efficient benign autoprompt generation techniques). We present the first thorough analysis of opaque machine-generated prompts, or autoprompts, pertaining to 6 LMs of different sizes and families, focusing on minimal pairs of natural language prompts and their "evil twins", i.e., opaque autoprompts that lead to the same continuation.

* Equal contribution. Correspondence: nathanael.rakotonirina@upf.edu.

[1]The term *autoprompt* was coined by Shin et al. (2020) to refer to the prompts generated by their algorithm. We repurpose the term here to refer to machine-generated prompts in general.

Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 48–68

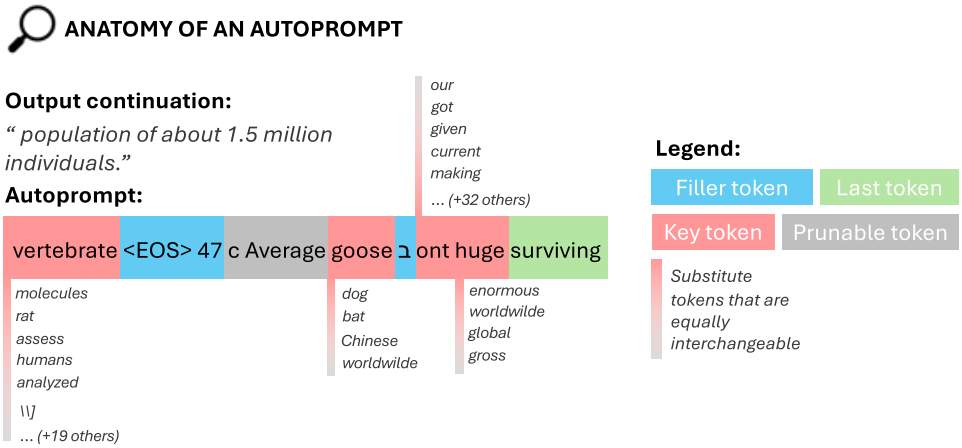


Figure 1: We analyze opaque machine-generated prompts (autoprompts) and identify four key components: (1) the *last token*, highly influential and difficult to modify; (2) *prunable* tokens, being ignored by the LM; (3) *key tokens*, which carry loosely related semantic information, essential for generation, and that can be replaced with semantically similar tokens; and (4) *fillers*, which can be substituted with a large amount of unrelated tokens but, unlike prunable tokens, cannot be deleted.

2 Related work

Starting with the seminal work of Wallace et al. (2019) and Shin et al. (2020), many studies have revealed that, using various discrete gradient-following techniques, it is possible to automatically discover prompts that, while unintelligible, let LMs generate a desired target output (e.g., Deng et al., 2022; Wen et al., 2023). Moreover, such prompts are at least to some degree transferable, in the sense that they can be induced using a LM, but then successfully used to prompt a different one, including much larger models (Rakotonirina et al., 2023; Zou et al., 2023). Initially, the interest was mainly in whether algorithmically-generated autoprompts could be used as alternatives to manually crafted prompts in knowledge-extraction tasks or other applications, but with recent progress in LM's ability to respond to natural language prompts, this goal has become somewhat obsolete. Autoprompts are however still an important concern because they can be used for adversarial purposes, for example to bypass safety filters to generate offensive or dangerous information (e.g., Zou et al., 2023; Geiping et al., 2024). Even more importantly, the fact that several modern LMs are more likely to provide information about the star formation process when prompted with “`Produ bundcules cation ofstars efect`” than when prompted with the question “What leads to the creation of new stars?” suggests that we still do not understand something fundamental about how LMs process language (Melamed et al., 2024).

There is relatively little work attempting to characterize the nature of autoprompts. Geiping et al. (2024) present a set of intriguing qualitative observations about how autoprompts support various types of attacks (e.g., by including instruction fragments in different languages), as well as an analysis of tokens commonly appearing in autoprompts. Ishibashi et al. (2023) find that autoprompts are less robust to token re-arrangement than natural prompts, whereas Rakotonirina et al. (2023) report that the autoprompts that best transfer across models contain a larger proportion of English words and, surprisingly, are *less* order-sensitive than autoprompts that do not transfer. Kervadec et al. (2023) analyze the activation paths of autoprompts and comparable natural sequences across the layers of a LM, finding that often they follow distinct pathways. Melamed et al. (2024) study, like us, what they call “evil twins”, namely autoprompts that produce continuations comparable to those of a reference natural sequence. They compare the relative robustness to token shuffling of autoprompts and natural prompts, finding that, depending on the model family, autoprompts might be more, less or comparably robust to shuffling. They also run a substitution experiment similar to the one we will describe below (but replacing tokens with a single, fixed, [UNK] token). They find that this ablation strongly affects the autoprompts: we find a more nuanced picture, by considering a large range of possible replacements.

3 Experimental setup

Models We use decoder-only LMs from the Pythia (Biderman et al., 2023) and OLMo (Groeneveld et al., 2024) families, as these are fully open-source models whose training data are publicly available. Specifically, in the text we discuss the results we obtained with Pythia-6.9B, and we replicate the main experiments with Pythia-1.4B, Pythia-12B, OLMo-1B, OLMo-7B, and OLMo-7B-Instruct in App. A, reporting similar results.

Data collection We sample 25k random English sequences from the WikiText-103 corpus (Merity et al., 2017), such that they contain between 35 and 80 (orthographic) tokens, and they are not interrupted by sentence boundary markers. We refer to these corpus-extracted sequences as *original prompts*. We also record the original continuation of these sequences in the corpus. We let moreover the LM generate a continuation of each prompt using greedy decoding. The generation process stops after a maximum of 25 tokens or when end-of-sentence punctuation is encountered. We filter out sequences whose generated continuation is less than 4 tokens long. As we are interested in genuine model generation, as opposed to cases where the model is simply producing a memorized corpus sequence, we compute the BLEU score (Papineni et al., 2002)[2] between the model continuation and the original continuation, removing sequences with BLEU greater than 0.1.[3] After filtering, we are left with a total of 5k sequences, which we use to train autoprompts. This dataset allows us to generate more complex prompts, that target full sentences instead of unique tokens.

Prompt optimization For each target continuation, we want to find a fixed-length autoprompt that makes the model produce that continuation. To achieve that, we maximize the probability of the target continuation given the prompt. More formally, if we denote the target sequence by $(t_1, ..., t_m) \in \mathcal{V}^m$, where $\mathcal{V}$ is the vocabulary, and the n-length autoprompt by $(p_1, ..., p_n) \in \mathcal{V}^n$ (in our case, $n = 10$), the optimization problem can be formulated as follows:[4]

$$\underset{(p_1,...,p_n)\in\mathcal{V}^n}{\text{minimize}} \; -\log \mathbb{P}_{LLM}(t_1, ..., t_m | p_1, ..., p_n)$$

We use a variant of Greedy Coordinate Gradient (GCG) (Zou et al., 2023), a widely used gradient-based algorithm that iteratively updates the prompt one token at a time (Ebrahimi et al., 2018; Wallace et al., 2019; Shin et al., 2020). During each iteration, we select the top 256 tokens with the largest negative gradients for every position, then we uniformly sample 256 candidates across all positions. We then compute the loss of each candidate replacement, and select the one with the lowest loss. We run up to 50 iterations of this process.We discard cases in which, after these iterations, we have not found an autoprompt that produces the very same continuation.

Data-set statistics The final data-set we use for the Pythia-6.9B experiments reported in the main text consists of 208 triples of original prompt, autoprompt and continuation.[5] The average original prompt length is of 39.3 tokens (s.d. 13.4); that of the continuations is of 8.4 tokens (s.d. 2.4).[6]

4 Experiments

4.1 Pruning autoprompts

Methodology We greedily prune the autoprompts in our data-set. Starting from the original sequence of *n* tokens, we strip each token in turn, and pick the *n-1*-length sequence that produces the same continuation as the original, if any (if there's more than one such sequence, we randomly pick one). We repeat the process starting from the shortened sequence, and stop where there is no shorter sequence generating the original continuation, or when we are down to a single-token prompt.

Roughly 20% of the tokens are *prunable* It is possible to shorten the original autoprompt in a clear majority of the cases (73.2%), with the average pruned autoprompt having lost 2.6 tokens of 10 (s.d.: 1.6). Table 1 (top section) shows randomly picked examples with the pruned tokens

[2]We use a modified version of BLEU that does not penalize short sequences. Scores are computed for up to 4-grams using uniform weights and add-ϵ smoothing.

[3]Schwarzschild et al. (2024) find that sometimes autoprompts act as "keys" to retrieve memorized materials. This is an intriguing property we don't further explore here, as we're interested in their more general ability to generate natural-language sequences.

[4]We empirically observed that using more than 10 tokens only increases the number of useless tokens (cf. pruning experiment in Section 4.1) without introducing any distinctive features. On the contrary, using less than 10 tokens was usually not enough to find the target continuation.

[5]We study relatively few autoprompts as it is very time-consuming to extract them for large model. Replicating the experiment with larger autoprompt sets using smaller models led to comparable results (App. A).

[6]Datasets and code are uploadedas supplementary materials, and will be made available upon publication.

Autoprompts	**Generated Continuation**
pulls**proper** Ryan**SP 184** critics **Mat?** embry " autoimmune,"antibodies?***<EOT>**arthyhatic:_ they **###iotics** parental **=** depressive teen ? lossJulies	The film is a mess, but it's a mess that's worth seeing. attack the body's own tissues. parents are going through a divorce.
Original Prompts	**Generated Continuation**
. . . **Aviation** Regiment (**based in Giebel**stadt, . . . **Robert Humanick of Slant Magazine** wrote**,** " . . . appealed to **the Government for** additional funding, a third . . . **an** autoimmune reaction causing **the body's immune** cells to . . . **)** — try to lead her through life as her	Germany) landed at the airport. The film is a mess, but it's a mess that's worth seeing. of which would come from the Treasury. attack the body's own tissues. parents are going through a divorce.

Table 1: Randomly selected examples of autoprompts and original prompts for Pythia-6.9B, with prunable tokens in bold. For original prompts, only the last 10 tokens are shown. '?' = difficult-to-render characters.

	Autoprompt					**Original Prompt**				
Kept	British	-	’	v	led	is	)	after	"	),
	King	West	remained	inaugural	five	be	which	(	she	film
Pruned	(	was	.	for	The	the	,	of	a	In
	on	In	âĢĶ	be	not	.	:	for	been	a

Table 2: Top-10 kept or pruned tokens for Pythia-6.9B, ranked by local mutual information for autoprompt and original prompts (for each cell, top-left has the highest value and bottom-right the lowest). Tokens are printed as follows: content word , artifact and punctuation , function word

highlighted in bold. Autoprompt-discovery algorithms fix the number of tokens as a hyperparameter. It is thus reasonable that some tokens in the final autoprompt are just there to fill all the required slots, and can consequently be pruned. This view is supported by the following observation. We roughly classified the autoprompt tokens into *language-like* and *non-linguistic*, such as digits, punctuation, code-fragments and non-ascii characters. We found that the proportion of non-linguistic tokens is decidedly higher among pruned tokens (46.6%) than among kept tokens (24.1%).[7] Table 2 (left) further shows tokens that are most typically kept or removed by the pruning algorithm according to the local mutual information statistics (Evert, 2005). Among the kept ones, we notice a prevalence of content words such as verbs, nouns and adjectives, whereas the typically pruned tokens are function words or word fragments.

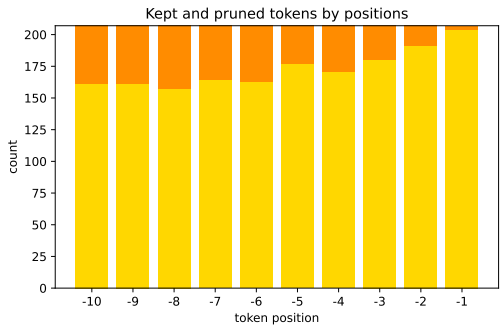
Importance of the last token The likelihood of pruning is not equally distributed across autoprompt positions: as Fig. 2a shows, the *last* token of the autoprompt is extremely unlikely to be pruned, pointing to the special role it plays in generating the continuation.[8] By looking qualitatively at typical last tokens (see examples in Table 1), we observe indeed that often they have a natural link to the beginning of the continuation. To confirm this quantitatively, in Fig. 3 we report the (log-transformed) corpus frequency distributions of the bigrams occurring in different contexts, with bigram frequencies estimated on the Pile corpus (Gao et al., 2020) that was used to train the Pythia models. There's a clear contrast between the bigram frequency distribution in natural text, exemplified by the natural prompts, and the autoprompts, that are mostly characterized by bigrams that never occur in the Pile. However, strikingly, the distribution at the autoprompt/continuation boundary is very similar to the one of natural text, quantitatively confirming that the last token of the autoprompt has a strong natural-language link to the continuation.

4.2 Replacing autoprompt tokens

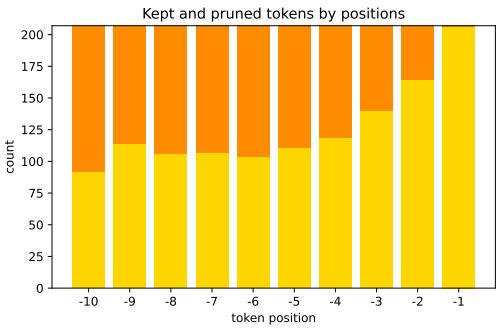
Methodology Working from now on with the pruned autoprompts, we replace the token in each position in turn with one of the 10k most frequent tokens from the Pile. We quantify the impact of the ablations in terms of BLEU score with respect to the original continuation. The ablation results are summarized in Fig. 4a, where replacements are binned based on the impact they have on the continuation (examples are presented in App. B).

[7] As a side note, we found that 28.4% of the full-autoprompt tokens are non-linguistic.

[8] More generally, Fig. 2a shows the last tokens before the very last also to be less prunable than earlier tokens.



(a) Autoprompts



(b) Original prompts (last 10 tokens)

Figure 2: Counts of tokens that were pruned (dark orange) and kept (yellow) by position for Pythia-6.9B, where 0 is the last position. Tokens at last position are extremely unlikely to be pruned.

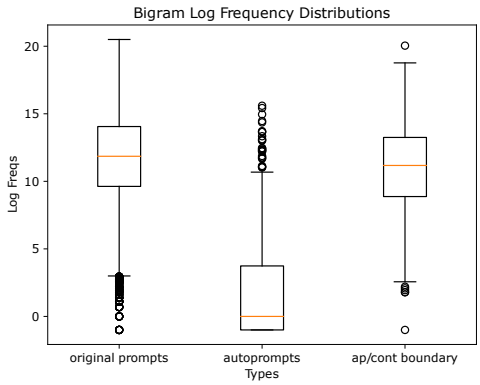


Figure 3: Pile-based log frequency distributions of bigrams in the *original prompts*, *autoprompts* and at the autoprompt/continuation boundary (*ap/cont boundary*) for Pythia-6.9B. Log(0) conventionally set to -1. Red line = median; boxes span interquartile ranges.

Impact of the replacements First, we confirm that non-pruned tokens in all positions play a significant role in generating the continuation, as shown by the fact that most replacements have a *strong* impact on BLEU. However, for all positions except the last, we also see that a non-negligible proportion of replacements do not affect the continuation at all, and in a significant proportion of cases the continuation is only mildly affected (as the examples in Table 10 of App. B show, even a BLEU score ≈ 0.2 typically corresponds to a continuation that is quite similar to the original). We confirm moreover the special role of the last token, that can almost never be replaced without catastrophic results. In general, as we approach the last position, it is increasingly more difficult to find replacements that do not strongly affect the continuation.

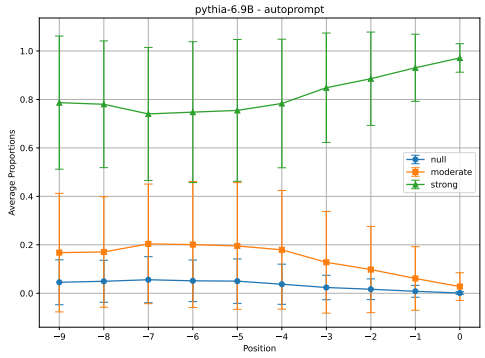
***Fillers* and *key* tokens** We further looked at the *equivalent sets*, consisting, for a given token, of the substitutes that keep the continuation unchanged.

	All	Lang-like	Non-Ling
Avg	302.8	237.6	512.1
Med	15.0	9.0	73.5
>50		34%	55%

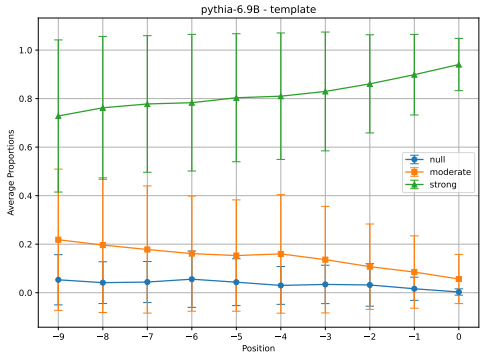
Table 3: Number of equivalent substitutes admitted by each autoprompt token for Pythia-6.9B. Results are shown for all, lang-like tokens, and non-ling tokens. Legend: **Avg**=average; **Med**=median; **>50**:% of tokens with more than 50 equivalents.

We measure that 76.4% of the tokens can be substituted by at least one equivalent, and each token has 302.8 substitutes on average (cf. Table 3). But the size of the equivalent set is highly variable, with half of the tokens having 15.0 or fewer equivalents. This disparity is intuitively informative about the role of each token. In particular, we identify the presence of *fillers* that can be replaced by a large number of substitutes, and *key* tokens, that admit more restrained equivalent sets and must thus carry more specific information. This is evidenced by the fact that *language-like* tokens tend to admit less equivalent substitutes than *non-linguistic* tokens. Table 4 (top section) shows examples of autoprompts with tokens color-coded based on their equivalent set size. When the equivalent set is small, we provide a random sample of it as an illustration.

Semantic consistency of the equivalent sets We measured the semantic consistency within equivalent sets in order to determine whether substitution in autoprompt is governed by semantic similarity (akin to synonymy in natural language) or by other factors. Specifically, we used FastText (Bojanowski et al., 2017) to compute the average semantic similarity: (a) between a token and each of its equivalents; and (b) among the equivalents



(a) Autoprompts



(b) Original prompts (last 10 tokens)

Figure 4: Average proportions of replacement effect types by position on pruned prompts for Pythia-6.9B, aligned from right (whiskers show standard deviations). *Null*-effect replacements leave the continuation unchanged. *Moderate* replacements have BLEU ≥ 0.2. *Strong* replacements have BLEU < 0.2.

themselves (including the original token).[9] The first measure indicates how semantically relevant the substitutes are in relation to the original token, while the second measure reflects the size of the substitute semantic space. A value close to 1 suggests that the substitutes cluster within a small semantic region, meaning they are semantically consistent with one another. We plot semantic similarity against set size in Fig 5a (App. B). As expected, small equivalent sets –corresponding to *key* tokens– tend to be semantically coherent on average. In this sense, they approximate near synonyms in natural language. This is confirmed by the examples in Table 4, e.g., 'Ireland' is replaced by 'Irish' and 'Tehran' by 'Iran'. Nonetheless, the semantic relation is often approximate, e.g., 'before' being replaced by 'then', 'change' by 'subsequently' or 'forming' by 'meeting', 'leaving', 'remaining', *etc.* In contrast, large sets –associated to *fillers*– often include tokens that are semantically unrelated to each other. We show in Table 13 (App. B) the most frequently found tokens in large sets, mostly consisting of subwords (sometimes interpretable morphemes), digits and fragments of named entities that don't carry strong semantic information.

Can humans predict which autoprompt tokens are more important? The four authors of the paper manually annotated the autoprompt dataset with a binary label marking, for each token, if it is intuitively important or not for the generation of the continuation. The annotation was made *a posteriori*, using the full autoprompt and the generated continuation (see app. C for a description of the annotation process). Results show that autoprompts possess interpretable properties, as the labeling is correlated with the number of replacements a token might have. Indeed, we measured that the median size of the equivalent set for tokens deemed important is 2, against 15 for those deemed replaceable, with the effect observed both for language-like and non-linguistic tokens (Table 15).

Traces of compositionality In cases where the replacement causes only a moderate change in the continuation, we see signs of "compositionality," in the sense that the continuation only displays a few new tokens broadly reflecting the meaning of the replacement. Some examples are presented in Table 5. We make the intuition more quantitative as follows. First, to facilitate automated similarity analysis, we extracted all cases where the replacement leads to the change of a single (typographic) word in the continuation (about 3% of the total). For these cases, we used FastText to measure the semantic similarity of both the original autoprompt token and its replacement to the original word in the continuation and to the changed one. We found that the original token is more similar to the new continuation word (vs. the original one) in only 48% of the cases, whereas the replacement token is more similar to the new continuation in 59% of the cases. We thus conclude that, indeed, there is a tendency for at least this type of replacement to work compositionally (a small change in the autoprompt leads to a semantically consistent change in the continuation). This, in turn, suggests that autoprompts do not function as unanalyzable holistic wholes. Their "meaning" to the model derives, at least partially, from assembling the meaning of its parts, as with natural language sequences. How-

[9] For technical reasons, due to the difficulty of obtaining reliable representations for non-linguistic tokens, we focused only on language-like tokens.

Autoprompts	Generated Continuation
WHM modelling tag Mus before either forming militant Annex Ireland	. . . or joining the Provisional IRA
before: *' then', ' change', ' subsequently'* **forming**: *' meeting', ' leaving', ' remaining', ' developing', ' selling', ' breaking', ' producing', . . .* **Ireland**: *' Irish'* **either/ militant**: *0 substitutes.*	
<|endoftext|> <|endoftext|> Star Defense *]{} 1950 blamed Tehran instead rhe	. . . United States for the 1953 coup.
1950: *' 1960'* **Tehran**: *' Iran'* **instead**: *' than', '=', ' then', ' back', ' to', '&', ' again', ' though', . . .* **blamed/ rhe**: *0 substitutes.*	
Original Prompt	**Generated Continuation**
for television Services Our is create programming that offers	. . . a unique experience for our viewers.
television: *' Fox', ' entertainment'* **create**: *' about',' emotional','Create','about',' differentiation',' unusual',' demands',' Create',. . .* **that**: *'that'* **offers**: *' provide',' provides',' offer'* **programming**: *0 substitutes.*	
Doctor can transcend reach ep iph any and greater	. . . understanding of the human condition.
ep: *' "',?,'**',' “',' new','“',' real',' story',']',' toward','ep',' towards','fn','Ñ', . . .* **iph**: *'ph','rep'* **any**: *' understanding','ening','aining',' insight'* **and**: *' for',' by',' or','A',' &',' through','or','an',' upon',' via','OR',' toward',' towards',' AND',' gain', . . .* **greater**: *' improved',' enhanced',' deeper',' wider'*	

Table 4: Examples of replacement for autoprompts and original prompts (10 last tokens) for Pythia-6.9B. Color represents the number of substitutes: > 50 substitutes and < 50 substitutes. When there are less than 50 substitutes, a random subset is displayed. Difficult-to-render characters are replaced by '?'. More examples in Table 12 (appendix).

ever, this looks nothing like the one performed by natural language syntax.

4.3 Shuffling autoprompt tokens

Methodology Previous work has uncovered a somewhat mixed picture in terms of the robustness of autoprompts to token order shuffling (Ishibashi et al., 2023; Rakotonirina et al., 2023; Melamed et al., 2024). Based on our *ad-interim* observations, we conjecture that the last token will be "rigid", as moving it around would strongly affect the continuation, whereas the preceding tokens might be more robust to order ablations. To test the conjecture, we randomly shuffled tokens (10 repetitions per autoprompt) and measured the resulting BLEU with respect to the original continuation. We either shuffled all tokens or left the last one fixed.

More than a bag-of-words The average BLEU when shuffling all tokens is at 0.03 (s.d. 0.04) and at 0.06 (s.d. 0.11) when leaving the last token in its slot. This difference is highly significant (paired t-test, $p < 0.001$). However, the low BLEU values suggest that, contrary to our conjecture, the autoprompt tokens before the last are not a bag of keywords, since their order matters as well. One possibility is that, while autoprompts as a whole do not constitute syntactically well-formed sequences, they are composed of tight sub-sequences that should not be separated. For example, given that modern tokenizers split text at the sub-word level, token-level shuffling will arbitrarily break words. Some support for the view that the catastrophic effect of shuffling pre-last tokens is due to short-distance dependencies comes by looking at the cases in which a bigram in an autoprompt (excluding the last position) is also attested in the Pile corpus, either in the original or in the inverted order. In 60.5% of these cases, the Pile frequency of the original bigram is larger than that of the inverted one, suggests some degree of natural local ordering among autoprompt tokens.

Autoprompts	Generated Continuation
Eg<EOT> Brown ~~mushrooms~~**/face** chooses suffix "brown	. . . " to describe the color of the ~~mushroom~~**/face**.
crossesFootnote Several panels accidentally have ~~feather~~**/bands** 517 chant collectively	. . . , as if they were a single ~~bird~~**/band**.
fecture ~~Phoenix~~**/Toronto** Latinoamous]], effectively	. . . making it the largest city in ~~Arizona~~**/Canada**.

Table 5: Example autoprompt token replacements leading to a small, interpretable change in the continuation for Pythia-6.9B (legend: ~~replaced~~**/replacement** in autoprompt; and ~~original~~**/new** in continuation).

4.4 Making human prompts more autoprompt-like

As a final piece of evidence that the dynamics we see at work in autopromts are general properties of how LMs process language, we re-ran some of the experiments above on the original corpus-extracted natural-language prompts, finding that they respond in similar ways to our ablations.

Pruning Applying the same greedy-pruning method to the original prompts, we find that more than 99.5% can also be pruned, with 23.8 tokens removed on average (s.d. 13.2). Considering the average token length of the original prompts is 39.0, this means that, strikingly, on average 61% of the tokens can be removed without affecting the continuation. Since the prompts are long, one could think that what is removed is primarily material towards the beginning of the sequence, but actually we find that 95% of the prompts also have pruned tokens among the last 10 items. Examples of the latter are in Table 1 (bottom). Prunable material often consists of modifiers whose removal does not affect the basic syntactic structure of the fragments ("*causing* ***the body's immune*** *cells*", "***Aviation*** *Regiment*"), but this is not always the case, and in many examples pruning turns well-formed sentences into seemingly unstructured token lists or telegraphic text at best. Still, like in the case of the autoprompts, the coherence of the transition between the prompt and the continuation is generally preserved ("*. . . a third / of which. . .* ", "*. . . as her / parents are. . .* "). Table 2 (right) shows the original-prompt tokens that are most typically kept vs. pruned. As for the autoprompts, highly prunable tokens consist entirely of common function words and punctuation marks. However, typically kept tokens might also be (somewhat rarer) function words and punctuation marks. Figure 2b presents pruning proportion by position for the last 10 tokens in the original prompts, confirming that, in this case as well, the last token is by far the most important one in determining the continuation. Interestingly, the contrast is even more dramatic than for autoprompts (Figure 2a).

Replacement We replicate the token-replacement experiment on the pruned original prompts, obtaining the results summarized in Figure 4b. Again, tokens become more replaceable as we move away from the end of the prompt, confirming the crucial role played by the very last token. Moreover, we confirm the presence of both *key* tokens, having few semantically related substitutes, and *fillers* with numerous semantically inconsistent substitutes (Figure 5b). Table 4 (bottom) shows examples in which the original prompt, despite pruning and replacement among the last 10 tokens, still triggers the same continuation. We see how the same principles that might explain the success of autoprompts are at work here, suggesting how autoprompts might take shape during the induction process.

Shuffling Shuffling all tokens of the original prompts after pruning leads to an average BLEU of 0.02 (s.d. 0.02), comparably to autoprompts. Leaving the last token in place leads to an average BLEU of 0.03 (s.d. 0.05). This small difference is highly significant (paired t-test, $p < 0.001$), confirming the importance of the last token (the difference stays equally significant if we compare shuffling all but the last token to shuffling while keeping one random non-last token fixed).

5 Discussion

Our findings about autoprompts, confirmed by autoprompt-inspired ablations of natural prompts, suggest that LMs might rely on a simplified model of language, where not all tokens have specific syntactic and semantic functions in an abstract syntactic tree. We note that the phenomenon of relying on over-simplified representations of the data is not specific to LMs. Convolutional Neural Network classifiers of visual data also latch onto superficial correlations in the data, leading to poor ood generalization (Jo and Bengio, 2017; Ilyas et al., 2019; Yin et al., 2019; Geirhos et al., 2020).

While we hope our results are of general interest, we recognize a number of limitations. First,

due to the time it takes to induce autoprompts with our computational resources, we could only experiment with 6 models, the largest of which has 12B parameters. We make our code available in hope that researchers with bigger resources will run similar experiments on a larger scale. For analogous reasons, we only experimented with one variant of the autoprompt inducing algorithm. Given that all algorithms we are aware of adopt similar gradient-following methods, and based on qualitative inspection of autoprompt examples in other papers, we expect our conclusions to hold for autoprompts independently of how they are induced, but this should be verified empirically.

Acknowledgments

We thank Emmanuel Chemla, Mor Geva and audiences at SISA, CIMeC and at the online HiTZ seminar series for feedback. Our work was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101019291). We also received funding from the Catalan government (AGAUR grant SGR 2021 00470). This paper reflects the authors' view only, and the funding agencies are not responsible for any use that may be made of the information it contains.

References

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In Proceedings of ICML, pages 2397–2430, Honolulu, HI.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In Proceedings of EMNLP, pages 3369–3391, Abu Dhabi, United Arab Emirates.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In Proceedings of COLING, pages 653–663, Santa Fe, New Mexico, USA.

Stephanie Evert. 2005. The Statistics of Word Cooccurrences. Ph.D dissertation, Stuttgart University.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. http://arxiv.org/abs/2101.00027.

Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. 2024. Coercing llms to do and reveal (almost) anything. In ICLR 2024 Workshop on Secure and Trustworthy Large Language Models, Vienna, Austria. Published online: https://openreview.net/group?id=ICLR.cc/2024/Workshop/SeT_LLM.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11):665–673.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In Proceedings of ACL, pages 15789–15809, Bangkok, Thailand.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. Advances in neural information processing systems, 32.

Yoichi Ishibashi, Danushka Bollegala, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Evaluating the robustness of discrete prompts. In Proceedings of EACL, pages 2373–2384, Dubrovnik, Croatia.

Jason Jo and Yoshua Bengio. 2017. Measuring the tendency of cnns to learn surface statistical regularities. ArXiv preprint, abs/1711.11561.

Corentin Kervadec, Francesca Franzon, and Marco Baroni. 2023. Unnatural language processing: How do language models handle machine-generated prompts? In Findings of EMNLP, pages 14377–14392, Singapore.

Rimon Melamed, Lucas McCabe, Tanay Wakhare, Yejin Kim, Howie Huang, and Enric Boix-Adsera. 2024. Prompts have evil twins. In Proceedings of EMNLP, Miami, FL. In press.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In Proceedings of ICLR Conference Track, Toulon, France. Published online: https://openreview.net/group?id=ICLR.cc/2017/conference.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of ACL, pages 311–318, Philadelphia, PA.

Nathanaël Rakotonirina, Roberto Dessì, Fabio Petroni, Sebastian Riedel, and Marco Baroni. 2023. Can discrete information extraction prompts generalize across language models? In Proceedings of ICLR, Kigali, Rwanda. Published online: https://openreview.net/group?id=ICLR.cc/2023/Conference.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary Lipton, and Zico Kolter. 2024. Rethinking LLM memorization through the lens of adversarial compression. In Proceedings of 2nd Workshop on Generative AI and Law (GenLaw 24), Vienna, Austria. Published online: https://arxiv.org/abs/2404.15146.

Taylor Shin, Yasaman Razeghi, Robert Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In Proceedings of EMNLP, pages 4222–4235, Online.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In Proceedings of ACL, pages 15725–15788, Bangkok, Thailand.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In Proceedings of EMNLP, pages 2153–2162, Hong Kong, China.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In Proceedings of NeurIPS, pages 51008–51025, New Orleans, LA.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. 2019. A fourier perspective on model robustness in computer vision. Advances in Neural Information Processing Systems, 32.

Andy Zou, Zifan Wang, Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. http://arxiv.org/abs/2307.15043.

Ethics Statement

If we do not achieve a genuine understanding of how LMs process and generate text, we cannot fully control their behavior and mitigate unintended or intentional harm. Opaque autoprompts are an indication that there are important aspects of LM prompting and generation that are still out of our control. Our investigation into the nature of this phenomenon contributes to a better understanding of how LMs work and, thus, ultimately, to make them safer and more predictable.

A Results with other models

Data-set statistics Following the procedure described in Section 3, we build a data-set for each of the following models: Pythia-1.4B, OLMo-1B, OLMo-7B, OLMo-7B-Instruct. Data-set statistics are presented in Table 6.

Pruning and shuffling The results of the autoprompt pruning and shuffling experiments are presented in Table 7. For all models, there is a difference in BLEU when shuffling all tokens vs. keeping last token fixed (paired t-test significant at $p < 0.01$). The difference stays comparably significant if, in the first condition, we leave a random non-last token fixed, so that the same number of tokens is shuffled in the two cases. Token pruning distribution by position is shown in Figures 6 7 8 9 10 (a).

Replacement For OLMo models, we estimate the top 10k most frequent tokens to be used in the replacement experiments using a sample of approximately 10 billion tokens from the Dolma corpus, which was used to train this model. (Soldaini et al., 2024). Proportions of replacement effect type by position are reported in Figures 6 7 8 9 10 (b).

Semantic consistency Similarly, we measure the semantic consistency of the equivalent sets in Figures 6 7 8 9 10 (c).

A.1 Making prompts more autoprompt-like

Pruning and shuffling The results of the pruning and shuffling experiments of natural prompts are presented in Table 8. For all models except OLMo-7B-Instruct, there is a difference in BLEU when shuffling all tokens vs. keeping last token fixed (paired t-test significant at $p < 0.01$). Token pruning by position is reported in Figures 6 7 8 9 10 (a).

Replacement Proportions of replacement effect type by position are reported in Figures 6 7 8 9 10

Model	Data-set size	Original prompt length	Continuation length
Pythia-1.4B	2473	38.6 (11.7)	9.4 (2.7)
Pythia-12B	129	37.5 (11.0)	7.9 (1.7)
OLMo-1B	500	38.4 (11.2)	8.5 (2.0)
OLMo-7B	115	38.9 (10.9)	8.3 (1.9)
OLMo-7B-Instruct	104	39.8(12.7)	8.4(2.7)

Table 6: The number of entries, average original prompt length (s.d.), and average continuation length (s.d.) of the additional data-sets.

Model	Pruning rate	Tokens pruned	BLEU shuffle all	BLEU shuffle except last
Pythia-1.4B	60%	1.2 (1.3)	0.03 (0.03)	0.05 (0.07)
Pythia-12B	86%	2.9 (1.9)	0.04 (0.05)	0.09 (0.12)
OLMo-1B	60%	1.2 (1.3)	0.02 (0.01)	0.04 (0.04)
OLMo-7B	60%	1.2 (1.4)	0.02 (0.02)	0.04 (0.05)
OLMo-7B-Instruct	27%	0.4(0.7)	0.01 (0.01)	0.02 (0.03)

Table 7: Results of the pruning and shuffling experiments of autoprompts. Pruning rate is the proportion of prompts in which at least one token could be pruned. There is a difference between BLEU scores when shuffling all tokens vs. keeping last token fixed for all models (paired t-test significant at $p < 0.01$).

Model	Pruning rate	Pruning rate last 10 tokens	Tokens pruned	BLEU shuffle all	BLEU shuffle except last
Pythia-1.4B	99%	95%	21.9(12.3)	0.02 (0.03)	0.03 (0.05)
Pythia-12B	100%	99%	23.8(11.7)	0.03 (0.04)	0.05 (0.12)
OLMo-1B	100%	97%	23.4(12.3)	0.02 (0.03)	0.03 (0.05)
OLMo-7B	100%	98%	24.0(12.6)	0.02 (0.02)	0.04 (0.06)
OLMo-7B-Instruct	92%	90%	21.4(14.6)	0.01 (0.01)	0.02 (0.03)

Table 8: Results of the pruning and shuffling experiments of original prompts. There is a difference between BLEU scores when shuffling all tokens vs. keeping last token fixed for all models except OLMo-7B-Instruct (paired t-test significant at $p < 0.01$).

(b).

Semantic consistency Semantic consistency of the equivalent sets are reported in Figures 6 7 8 9 10 (c).

B Token replacement examples

We show randomly picked examples of single-token autoprompt replacements that do not affect the continuation, have a moderate effect on it or have a strong effect on it in tables 9, 10 and 11, respectively. In addition, Table 12 provides additional examples of replacement with null-effect, where tokens are colored based on the number of equivalent substitutes.

Large substitution sets Table 13 presents a larger list of tokens that are the most frequently found in large equivalent substitution sets. It mostly consists of subwords (sometimes corresponding to interpretable morphemes), digits and (fragments of) named entities, that do not carry strong semantic information. We conjecture that digits appear in cases where the exact number is not crucial for generating the continuation, but the notion of quantity or date is important (e.g., in that case 5, 10 or 18 are equivalents). A similar interpretation holds for named entities. For instance, 'Moore', 'Brown' or 'Smith' are common proper nouns that could refer to a typical North-American entity in an exchangeable way.

Semantic consistency of the equivalent sets Table 5 provides the semantic similarity between substitutes.

C Human annotation of token importance

The four authors of this paper manually annotated the autoprompt dataset (for the Pythia 6.9B model), assigning a binary label to each token to indicate whether it is intuitively important for generating the continuation. The annotation process was conducted *a posteriori*, with annotators having access to the full autoprompt and its corresponding generated continuation. To minimize order effects, tokens were randomized before annotation. As an illustration, Table 14 provides some annotated examples.

In total, 1971 tokens have been labeled by at least one annotator. We assess inter-annotator agreement on a subset of 50 tokens: the four annotators agreed on 63.3% of the examples, and at least 3 annotators agreed on 89.8% of the examples.

Table 15 provides the full comparison between importance annotation and equivalent set size.

D Computing resources

All experiments were run on a cluster composed of 11 nodes with 5 NVIDIA A30 GPUs each. The autoprompt search for Pythia-1.4B took approximately 600 GPU hours. Pruning, replacement and shuffling experiments for Pythia-1.4B took 1500 GPU hours overall. Compute demand for the other models was comparable, although we had to settle for smaller autoprompt sets.

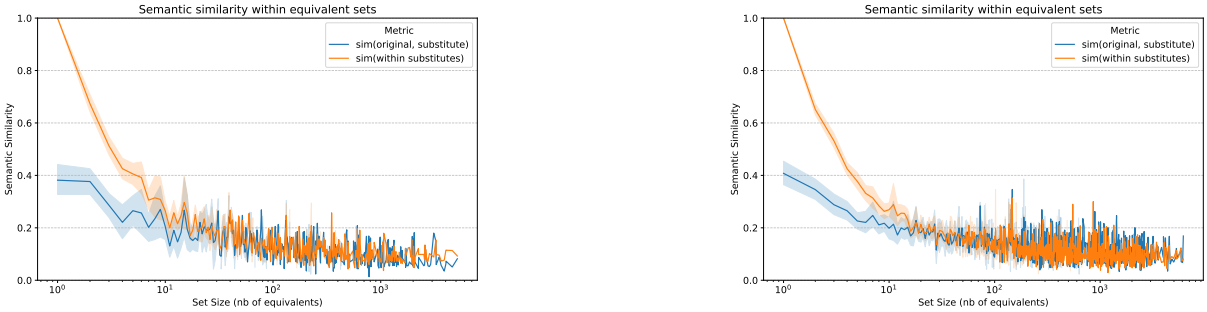
E Assets

Besides standard tools such as Python and libraries such as NumPy and SciPy, we used the following tools and datasets, in accordance with their respective terms and licenses.

- Dolma https://huggingface.co/datasets/allenai/dolma; license: ODC-By
- NLTK https://www.nltk.org; license: apache-2.0
- OLMo https://huggingface.co/allenai/OLMo-7B; license: apache-2.0
- The Pile https://pile.eleuther.ai/; license: MIT
- PyTorch https://pytorch.org; license: bsd
- Pythia https://huggingface.co/EleutherAI/pythia-1.4b-deduped; license: apache-2.0
- Huggingface Transformers https://github.com/huggingface/transformers; license: apache-2.0
- Wikitext https://huggingface.co/datasets/wikitext; license: Creative Commons Attribution Share Alike 3.0

autoprompt: crossesFootnote ~~Several~~/**Accordingly** panels accidentally have feather 517 chant collectively *continuation:* , as if they were a single bird.
autoprompt: <EOT><EOT> ~~unbelievable~~/**lands** deep intuitive *continuation:* understanding of the human condition.
autoprompt: throne were~~ivered~~/**print**ceryassociated pursuing Somerset where *continuation:* he was arrested and imprisoned in the Tower of London.
autoprompt: ~~tables~~/**app** Its 590 chapel wide Rosadian *continuation:* marble floor is the largest in the world.
autoprompt: FLOAT depicts 1933}}?significant professional MMA ~~classic~~/**adapt** pickup after *continuation:* a long absence from the sport.

Table 9: Randomly selected *null-effect* replacement examples. Replaced tokens and replacements are separated by "/" and in bold.



(a) Autoprompts (b) Original Prompts

Figure 5: Semantic similarity between substitutes. When a token admits fewer substitutes, they generally lie in a smaller semantic space than when the substitute count is large. The rightmost data points correspond to the most flexible tokens and tend to be semantically inconsistent. The leftmost data points are tokens admitting 0 or few substitutes, which are semantically closer.

autoprompt: ~~TLR~~**/ract**<EOT> pledmanifest behalf breached adiabarfollow
original continuation: ~~the terms of the settlement agreement.~~
modified continuation: **ed the terms of the agreement.**
modified continuation BLEU: 0.49

---

autoprompt: stylish 1876 painting escaped creditors ~~Anne~~**/Im** decides
original continuation: ~~to sell it to pay off her debts.~~
modified continuation: **to sell it to a wealthy collector.**
modified continuation BLEU: 0.32

---

autoprompt: ~~Round~~**/22** transferring'$, when splash energy
original continuation: ~~is transferred to the surrounding water.~~
modified continuation: **is transferred to the surface of the liquid.**
modified continuation BLEU: 0.32

---

autoprompt: <EOT>~~<EOT>~~**/Two**<EOT> Return onto Broadway where Gardens Airport coincide
original continuation: ~~with the Garden State Parkway.~~
modified continuation: **with the intersection of Broadway and the Garden State Parkway.**
modified continuation BLEU: 0.45

---

autoprompt: ~~<EOT>~~**/Where** Treaty<EOT>Unionduring
original continuation: ~~the American Revolutionary War.~~
modified continuation: **the Revolutionary War.**
modified continuation BLEU: 0.33

Table 10: Randomly selected *moderate-effect* replacement examples (BLEU after replacement is of at least 0.2 but below 1). Replaced tokens and replacements are separated by "/" and in bold.

autoprompt: FIRST<EOT>protective talents~~Sarah~~**/bias**<EOT> uses learning technical
original continuation: ~~skills to protect herself and others.~~
modified continuation: **skills to protect the user from harm.**
modified continuation BLEU: 0.15

---

autoprompt: fiveZero ~~Parl~~**/ins**terson homicide By 1867 Provincialswick warranted
original continuation: ~~the establishment of a police force.~~
modified continuation: **the appointment of a police magistrate.**
modified continuation BLEU: 0.19

---

autoprompt: stylish 1876 painting escaped ~~creditors~~**/()** Anne decides
original continuation: ~~to sell it to pay off her debts.~~
modified continuation: **to take a break from her work and go for a walk.**
modified continuation BLEU: 0.02

---

autoprompt: Pretty."byterBlood "A realistic ~~work~~**/convicted** compatible
original continuation: ~~with the other works of the author~~
modified continuation: **with the "Atheist" and "Atheist" "Atheist" an[...]**
modified continuation BLEU: 0.02

---

autoprompt: ~~————————————————————————~~terrorismworker ~~is~~**/exit** killed
original continuation: ~~in a suicide bombing in Iraq.~~
modified continuation: **in Iraq**
modified continuation BLEU: 0.03

Table 11: Randomly selected *strong-effect* replacement examples (BLEU after replacement is below 0.2). Replaced tokens and replacements are separated by "/" and in bold. Hard-to-render characters replaced by "?".

Autoprompts	Generated Continuation
ipzig suspended Emma Leigh 's 956 time commitments and as	. . . a result, she was able to devote herself to her writing.
ipzig: *' century', ' society', ' secret', ' glad', ' revolution', ' intellectual', ' consciousness', . . .* **Emma**: *' women', ' woman', ' tau', ' wife', ' sex', ' da', ' ura', ' snow', . . .* **commitments**: *' attention', ' traditional', ' burden', ' politics', ' busy', . . .* **and**: *' how', ' and', ' et', ' that', ' ad', ' und', . . .* **as**: *' as'* **suspended**: *0 substitutes*	
COMMAND uminous photometry has ? continually undermined	. . . the idea that the Sun is a star.
has: *' helps', ' raise', ' Williams'* **undermined**: *' rejected'* **photometry**: *0 substitutes.*	
Original Prompt	**Generated Continuation**
South Australian with Wales requested that Major General William ois	be appointed to command the expedition.
requested: *' asked'* **Major**: *' sub',' major',' officer',' Sub'* **General**: *' general',' officer',' Gen',' Captain'* **that**: *0 substitutes.*	
erected in 1896 listed of the sand stone market building	, which was built in the late 19th century.
the: *' an',' its',' some',' great',' old',' further',' additional',' especially', . . .* **stone**: *'aligned','ized','á','ette','ished',' stone','split','white','ulate', . . .* **building**: *' court',' structure',' complex',' House',' unit',' track',' society',' mechanism', . . .*	

Table 12: Additional Examples of replacement for autoprompts and original prompts (10 last tokens). Color represents the number of substitutes: > 50 substitutes and < 50 substitutes. When there are less than 50 substitutes, a random subset is displayed. Difficult-to-render characters are replaced by '?'.

105	<EOT>	()	113	ini	Mar	ran	25	be	Green	101	56	,	mat	ato
ator	Rock	35	115	Moore	114	SS	lim	eter	ester	abe	uns	Head	Char	?
?	?	ac	aa	aur	?	ong	ost	den	FF	54	a	ab	ud	sl
rim	st	?	022	1	37	91	INT	op	ore	ef	rel	cha	ast	ives
2000	ash	arch	ath	eps	ced	Har	fall	met	ales	140	Ros	5	48	21
33	101	38	75	600	?	**	oth	aff	ay	br	LE	?	anch	der
cy	ris	fer	inner	ots	arn	Sil	Red	ros	,	cent	Brown	ister	dec	50
44	40	28	500	?	41	46	85	112	93	45	57	n	y	em
ard	na	eng	ater	ree	EE	urg	?	?	ord	oto	1	-	ign	sen
ans	ach	dis	ush	Br	Cam	pol	one	?	ner	18	42	48	250	96
Bell	94	117	19	ial	pp	ba	star	?	mar	har	hor	Tor	lan	vi
Sl	1986	arg	ender	Sand	First	Po	Che	Jones	100	per	ative	ways	hal	ben
bon	Ross	6	?	30	46	31	65	400	1000	LE	33	am	Bar	ace
amb	Ab	ull	Reg	ena	orm	stra	s	co	ards	win	iver	Ha	?	pin
init	dist	pres	Ray	47	Smith	120	128	number	án	Kar	4	70	26	27
98	200	80	200	2000	45	500	49	110	63	13	el	ale	za	ute

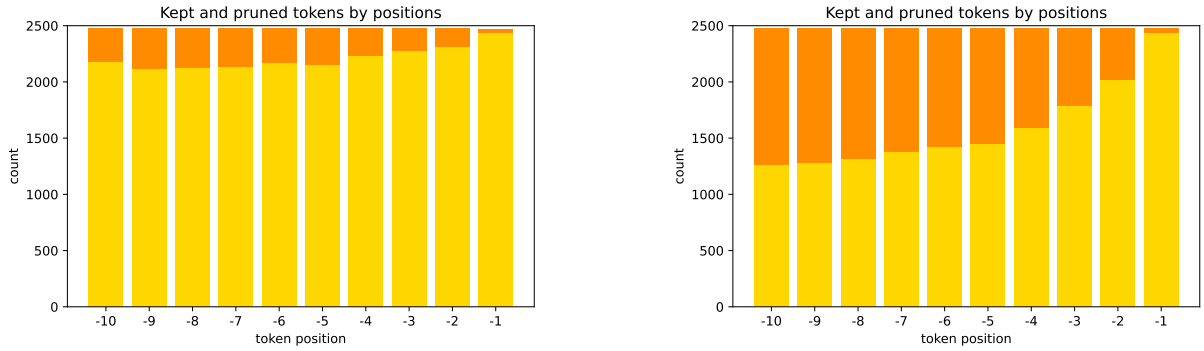
Table 13: List of the 240 tokens most frequently found in large equivalent sets (observed in about 10% of the sets). Difficult-to-render characters are replaced by '?'. Tokens are marked as follows: digit, artifact and punctuation, named entity, subword and morphological unit.

Autoprompt	Continuation	Token to annotate	Annotation
<EOT><EOT><EOT><EOT> _____<EOT> European devastated 1941 prompted	. . . the United States to enter the war.	<EOT>	0
clut British'); from 40 UTC anyone emitted one speaker	. . . in the UK would be heard in the UK.	speaker	1
Croat trainersers?? Greece Catholics Deltaclaim independently	. . . from the Greek Orthodox Church.	Greece	1
Shan Marxist Augustine Fran??ois State TERis Railroad serves ?	. . . the city of Terre Haute, Indiana.	Railroad	1
defenseLittle<EOT>Content<EOT> forests constitute reliable leaving habitats	. . . for many species of birds and mammals.	leaving	1
earliest 2??–¢ flares when Jacksonville dominancethe metropolitan	. . . area was still in its infancy.	dominance	0
‡?Å Berry paternalEffects cephal???. Johann Born evenublished	. . . a book on the subject in 1775.	Johann	1
1953 that Dix Teachers 1989 Archbishopreceived a Hamilton Legacy	. . . Award for his contributions to education.	that	0

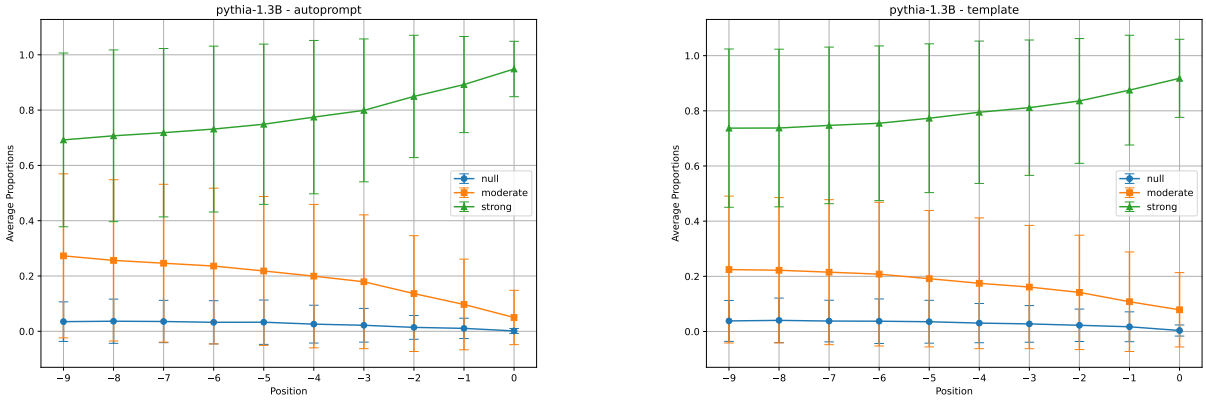
Table 14: We provide 8 examples of human annotations of token's importance. Using the autoprompt and the continuation, the annotator had to label the "token to annotate" as *important* (labeled with 1) or *not important* (labeled with 0). Difficult-to-render characters are replaced by '?'

	All		Lang-like		Non-Linguistic	
	Imp.	**Not-Imp.**	**Imp.**	**Not-Imp.**	**Imp.**	**Not-Imp.**
Average	163.4	372.2	150.8	298.7	268.6	538.4
Median	2.0	35.0	2.0	23.0	13.5	83.0

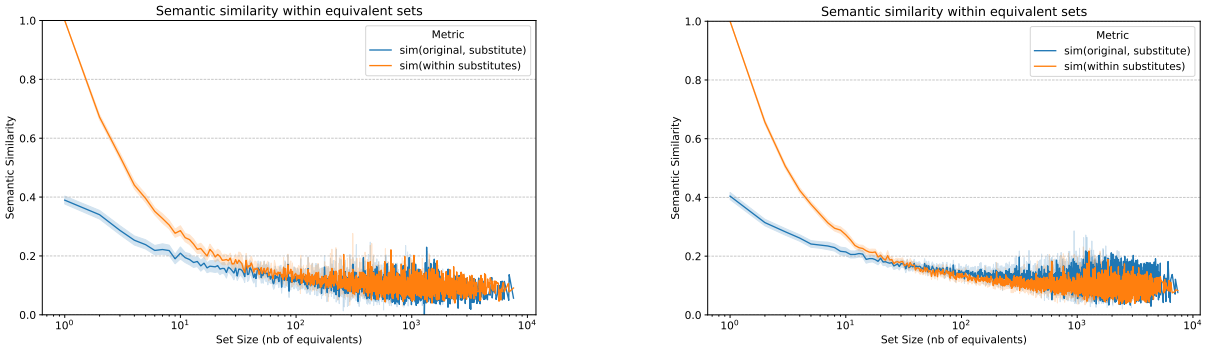
Table 15: Comparison of the equivalent set size (average and median) between tokens classified as Important (Imp.) or Not-Important (Not-Imp.) by human experts, across different token categories.



(a) Count of pruned (orange) and kept (yellow) tokens (*left:* autoprompt; *right:* original prompt).

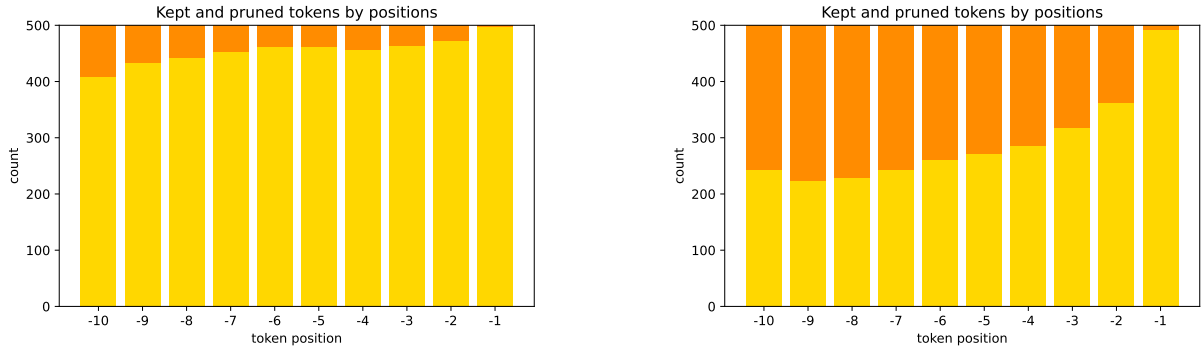


(b) Average proportions of replacement effect types by position (*left:* autoprompt; *right:* original prompt).

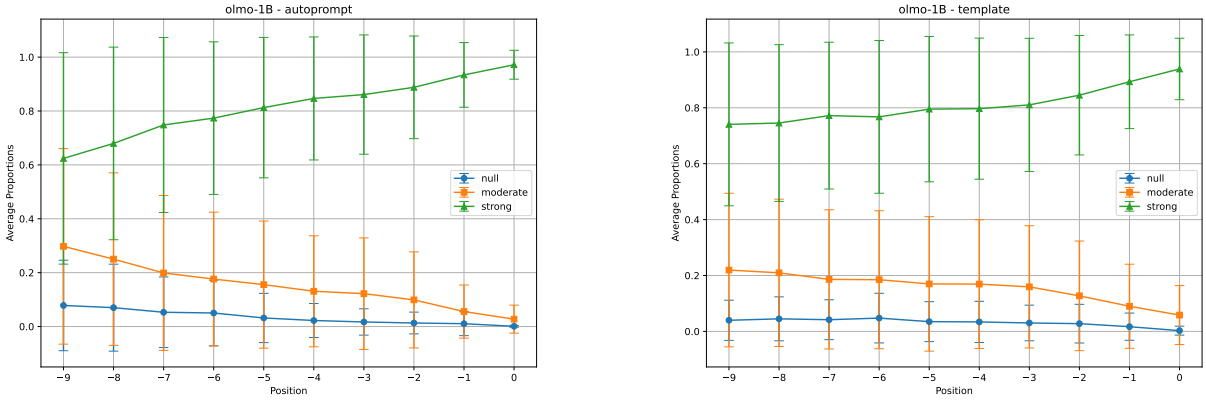


(c) Semantic similarity between substitutes (*left:* autoprompt; *right:* original prompt).

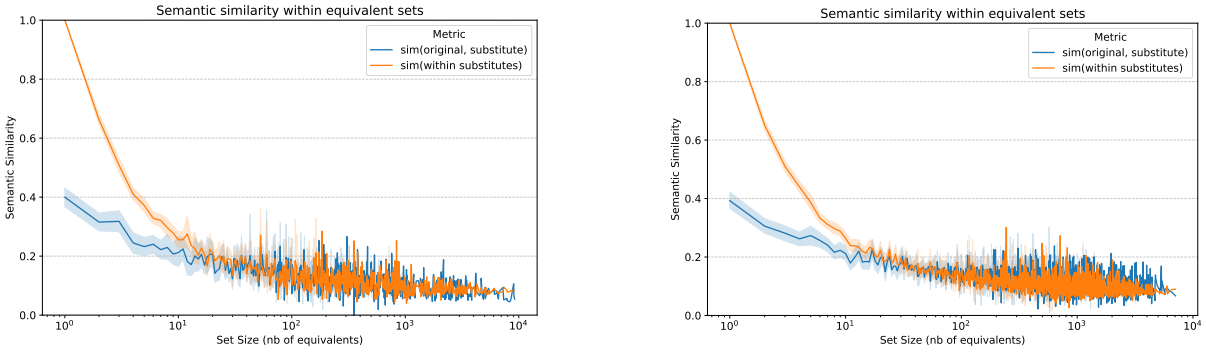
Figure 6: **Pythia-1.3B**: Reproducing pruning and replacement experiments.



(a) Count of pruned (orange) and kept (yellow) tokens (*left:* autoprompt; *right:* original prompt).

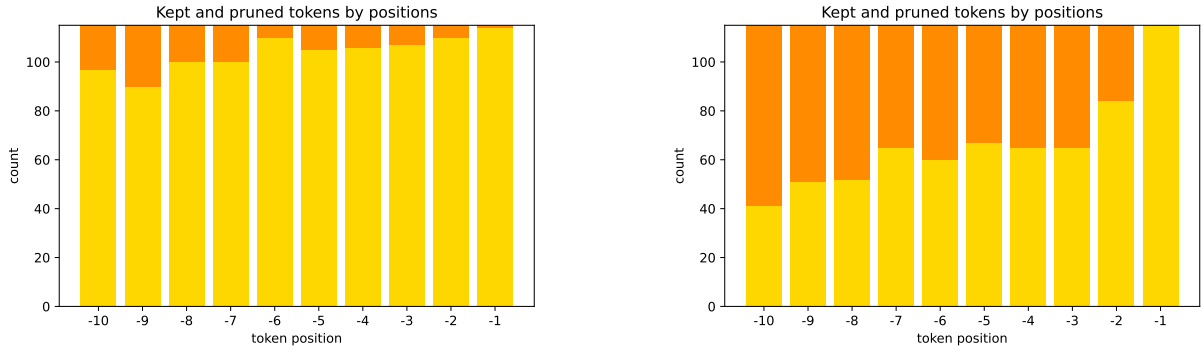


(b) Average proportions of replacement effect types by position (*left:* autoprompt; *right:* original prompt).

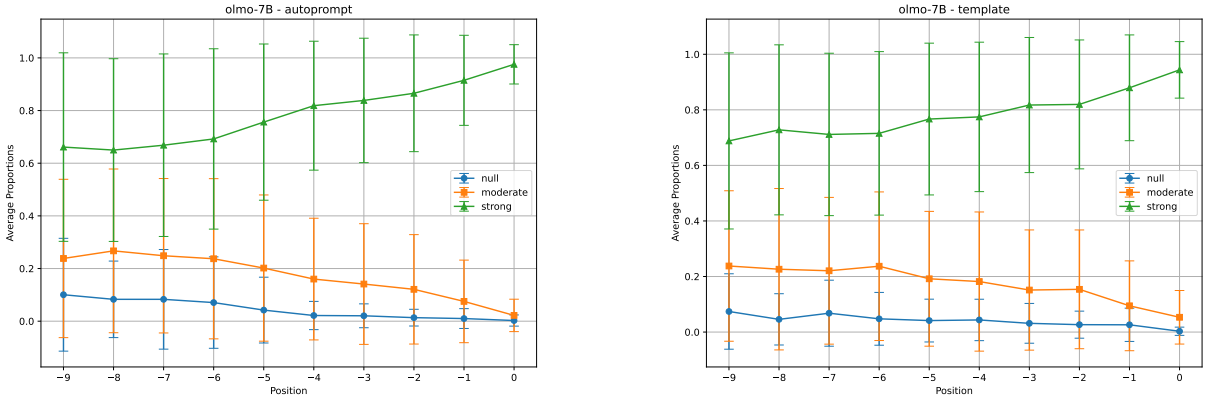


(c) Semantic similarity between substitutes (*left:* autoprompt; *right:* original prompt).

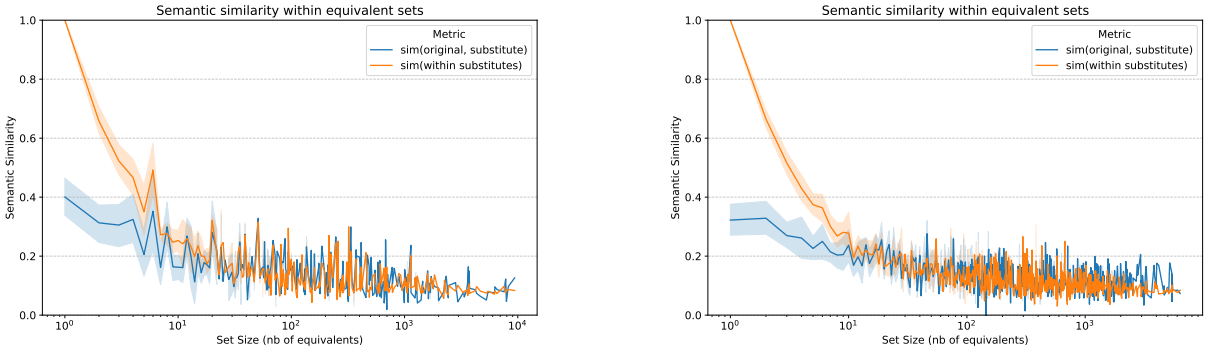
Figure 7: **OLMo-1B**: Reproducing pruning and replacement experiments.



(a) Count of pruned (orange) and kept (yellow) tokens (*left:* autoprompt; *right:* original prompt).

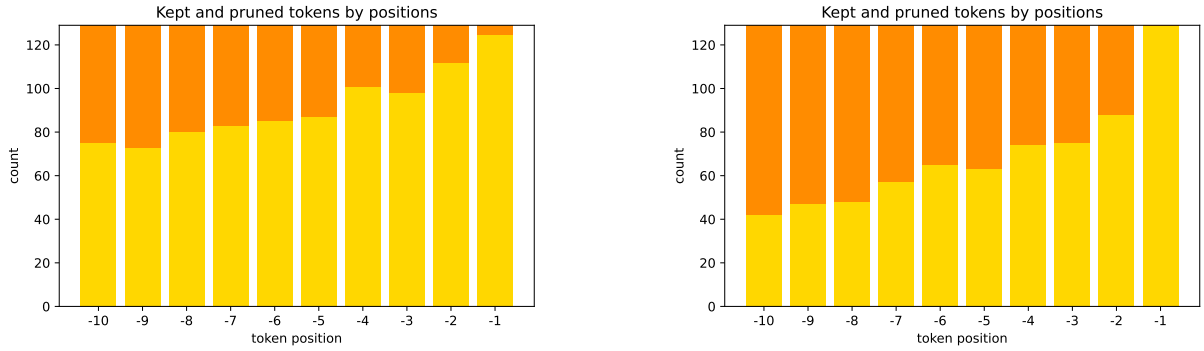


(b) Average proportions of replacement effect types by position (*left:* autoprompt; *right:* original prompt).

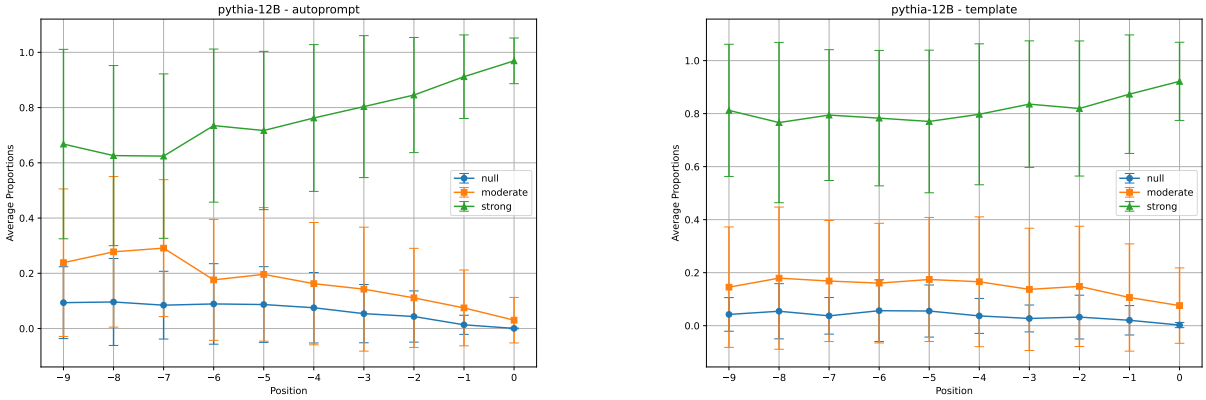


(c) Semantic similarity between substitutes (*left:* autoprompt; *right:* original prompt).

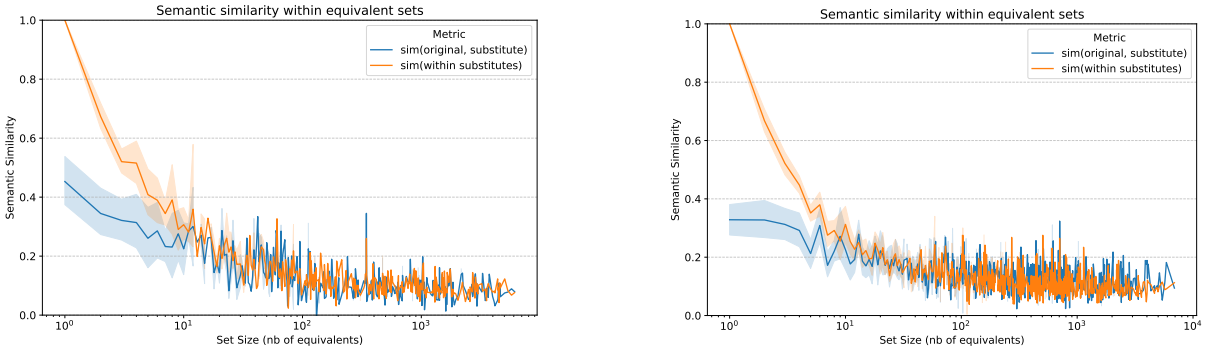
Figure 8: **OLMo-7B**: Reproducing pruning and replacement experiments.



(a) Count of pruned (orange) and kept (yellow) tokens (*left:* autoprompt; *right:* original prompt).

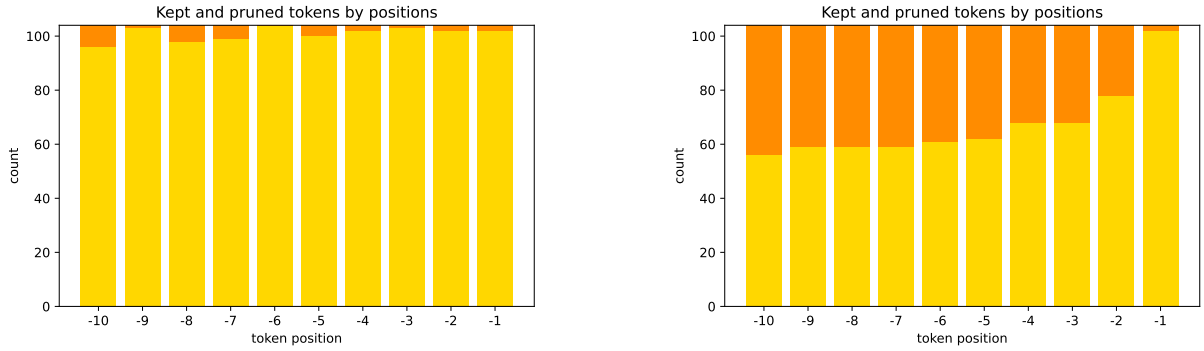


(b) Average proportions of replacement effect types by position (*left:* autoprompt; *right:* original prompt).

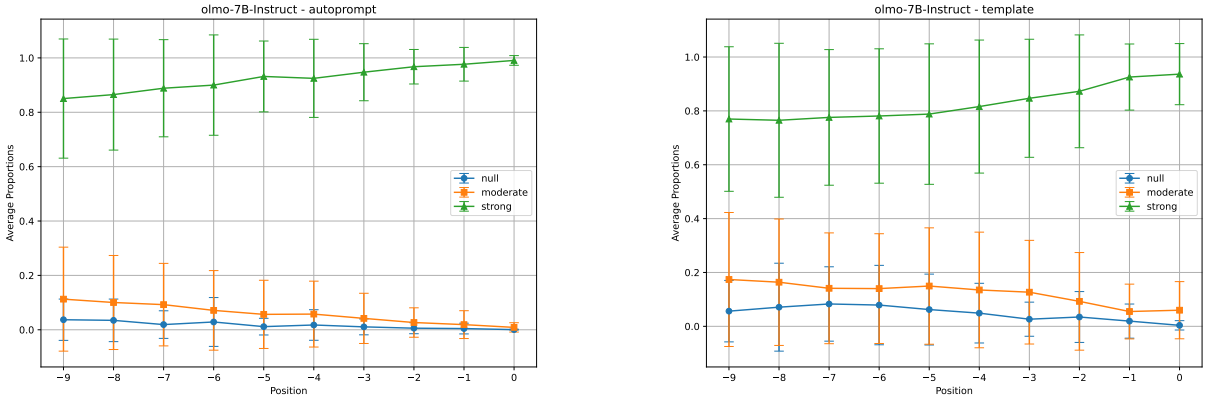


(c) Semantic similarity between substitutes (*left:* autoprompt; *right:* original prompt).

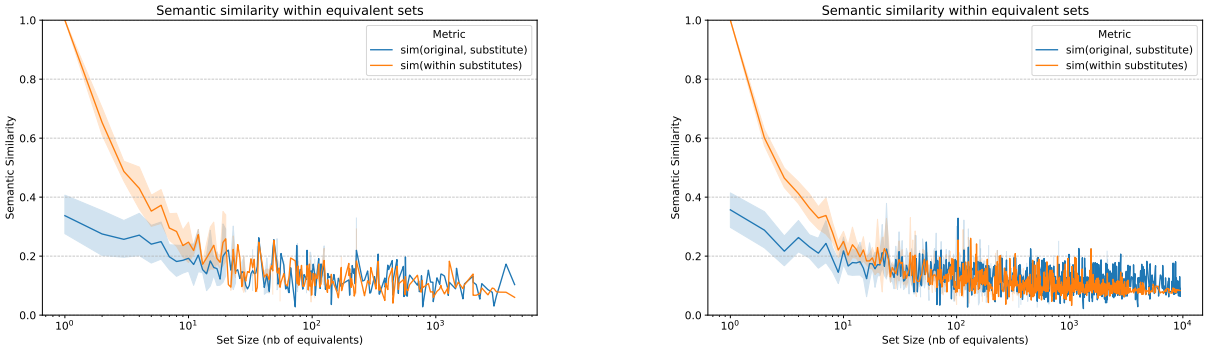
Figure 9: **Pythia-12B**: Reproducing pruning and replacement experiments.



(a) Count of pruned (orange) and kept (yellow) tokens (*left:* autoprompt; *right:* original prompt).



(b) Average proportions of replacement effect types by position (*left:* autoprompt; *right:* original prompt).



(c) Semantic similarity between substitutes (*left:* autoprompt; *right:* original prompt).

Figure 10: **OLMo-7B-Instruct**: Reproducing pruning and replacement experiments.