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

Anonymous ACL submission

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 3 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 fillers that probably appear in the prompt as a by-product of the fact that the optimization process fixes the number of tokens. The remaining tokens tend to have at least a loose semantic relation with the generation, although they do not engage in well-formed syntactic relations with it. We find moreover that some of the ablations we applied to machine-generated prompts can also be applied to natural language sequences, leading to similar behavior, suggesting that autoprompts are a direct consequence of the way in which 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 several other studies after them 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., 2023b; Geiping et al., 2024).

In this paper, we present 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. First, in autoregressive models the last token of the prompts has a disproportionate role in generating the continuation, and this last token is both very important and often quite transparent in autoprompts. Second, several tokens contributing to the opaqueness of autoprompts act as fillers that are ignored by the model. Third, the non-final tokens that are actually influencing generation might do so in a keyword-like way, and even occasionally display a loose form compositionality, in a sense we'll make precise below. 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).

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., Shin et al., 2020; Deng et al., 2022; Wen et al., 2023; Melamed et al., 2024),. Moreover, such prompts are at least to some degree transferable, in the sense that they

[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.

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., 2023b; Melamed et al., 2024).

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 LM applications (e.g., Shin et al., 2020; Deng et al., 2022; Rakotonirina et al., 2023). With the recent astounding progress in LM ability to respond to natural language prompts, this goal has become somewhat obsolete, but autoprompts are still an important concern because they can be used for adversarial purposes, for example to bypass LM security filters in order to generate offensive or dangerous information (e.g., Zou et al., 2023b; 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 the string "Produ bundcules cation ofs̈tars efect" than when prompted with the question "What leads to the creation of new stars?" suggests that there is something fundamental we still do not understand about how LMs process language.[2]

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.

[2]Example from Melamed et al., 2024.

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-1.4B, and we replicate the main experiments with Pythia-6.9B and OLMo-1B in Appendix B, 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 (period, exclamation mark, question mark) 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)[3] between the model continuation and the original continuation, removing sequences with BLEU greater than 0.1.[4] After completing the filtering processes, we are left with a total of 5k sequences, which we use to train autoprompts.

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

[3]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.

[4]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.

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:

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

We use a variant of Greedy Coordinate Gradient (GCG) (Zou et al., 2023b), 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. In our experiments, we run up to 150 iterations of this process.[5] We discard cases in which, after this number of iterations, we have not found an autoprompt that produces the very same continuation as the original prompt.

Data-set statistics The final data-set we use for the Pythia-1.4B experiments reported in the main text consists of 2473 triples of original prompt, autoprompt and continuation. The average original prompt length is of 38.6 tokens (s.d. 11.7); that of the continuations is of 9.4 tokens (s.d. 2.7).[6]

4 Experiments

4.1 Pruning autoprompts

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. It is possible to shorten the original autoprompt in a clear majority of the cases (60%), with the average pruned autoprompt having lost 1.9 tokens of 10 (s.d.: 1.1). Table 1 shows randomly picked examples of autoprompts with the pruned tokens 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. The view that pruned tokens are filler-like is supported by the following observation. We roughly classified the tokens into the autoprompts 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 (32.9%) than among kept tokens (24.5%).

Table 2 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.

As expected if they are somewhat filler-like, pruned tokens are easier to ignore when randomly interspersed into natural sequences than non-pruned tokens are. To quantify this claim, we extracted the set of tokens that are always pruned in our autoprompt corpus, as well as the set of tokens that are never pruned. We then inserted in each of the original prompts a random sample of 3 always-pruned or always-kept tokens, in random positions. We measured how this affected the continuation by computing the average BLEU scores for the continuations after the insertion, with the original continuations as reference. Not surprisingly, in both cases adding 3 new tokens does affect generation, but adding pruned tokens has a *lower* effect than adding kept tokens: the average pruned-token insertion BLEU is at 0.40 (s.d. 0.39); the kept-token insertion BLEU is at 0.37 (s.d 0.39). The difference is highly significant according to a paired t-test ($p < 0.001$).

Importantly, the likelihood of pruning is not equally distributed across autoprompt positions: as Fig. 1 shows, the *last* token of the autoprompt is extremely unlikely to be pruned, pointing to the special role it plays in generating the continuation. It seems that, when analyzing autoprompts, we can establish a 3-way distinction, in terms of importance, between pruned tokens, kept tokens in all positions but the last, and the token that occurs in

[5] We set the number of candidates to 256 following Zou et al. (2023b). We converged on using 150 as the maximum number of iterations based on a few exploratory runs, without extended hyperparameter search.

[6] We include data-sets and code as supplementary materials. They will be made publicly available upon publication a under a CC-BY-SA and a CC-BY license, respectively.

autoprompt:	*continuation:*
Billboard franchise<EOT> Large venues**It** 1897 comfortablycontained what	was then the largest venue in the world.
shareholders discontinued visual impairment schools subsequently **allegedly** ???**atically** lead	to a decline in the number of visually impaired students.
Scott Brock)<EOT> Magazine ϵ finaleuntil Lisa put	the finishing touches on the cover.
California drones operate airlines**??** **Bigletters**** license from	the Federal Aviation Administration (FAA).
<EOT><EOT><EOT> **Bal** sank**CCA**!" did aircraft regain	power and the plane crashed into the sea.

Table 1: Random autoprompts examples with prunable tokens in bold. Difficult to render characters replaced by "?".

kept:	*pruned:*
represents	I
denotes	also
premiered	they
filmmaker	time
Their	th
means	out
infamous	G
guitarist	T
His	S
soundtrack	well

Table 2: Top-10 kept (left) or pruned (right) autoprompt tokens ranked by local mutual information.

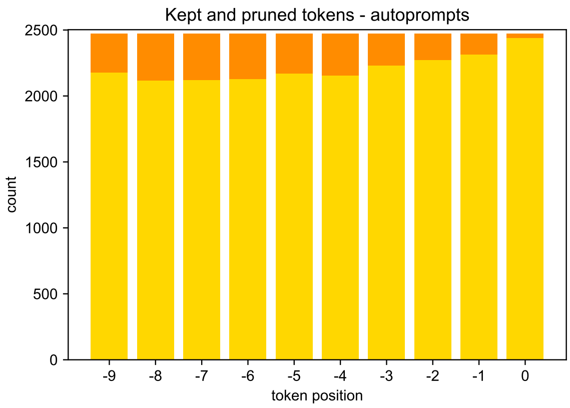

Figure 1: Counts of autoptompt tokens that were pruned (dark orange) and kept (yellow) by position, where 0 is the last position.

the last position.[7]

In support of this analysis, we conducted the following experiment. For each autoprompt, we measured the proportion of tokens that also occur in the corresponding original prompt (and are thus likely to be meaningfully related to the continuation), distinguishing between pruned tokens, kept tokens except last, and last tokens. We found a significant difference in overlap between pruned and kept-non-last-token overlap: 0.66% (s.d. 6.75%) vs. 2.25% (s.d. 5.84%), $p < 0.001$. However, the very last token is much more likely to overlap with the last token of the original sequence than the other kept tokens are to overlap with *any* token in the latter (11.10% vs. 2.25%).

By looking qualitatively at typical last tokens (see the example 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. 2 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

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. 3, where replacements are binned based on the impact they have on the continuation (examples are presented in the tables of Appendix A).

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 8 of Appendix A show, even a BLEU score around 0.2 typically

[7]This is a somewhat coarse distinction, since, as Fig. 1 shows, the last few tokens before the very last also tend to be less prunable than earlier tokens.

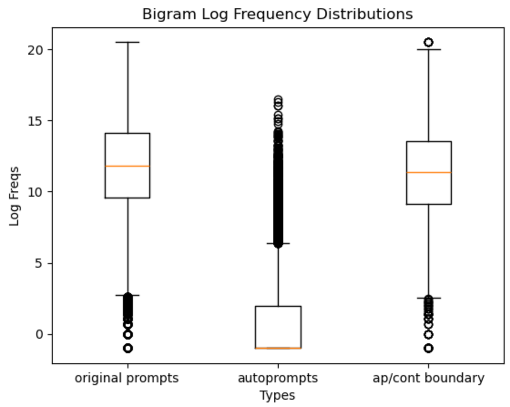

Figure 2: Pile-based log frequency distributions of bigrams in the *original prompts*, *autoprompts* and at the autoprompt/continuation boundary (*ap/cont boundary*). Log(0) conventionally set to -1. The red line represents the median; boxes span interquartile ranges.

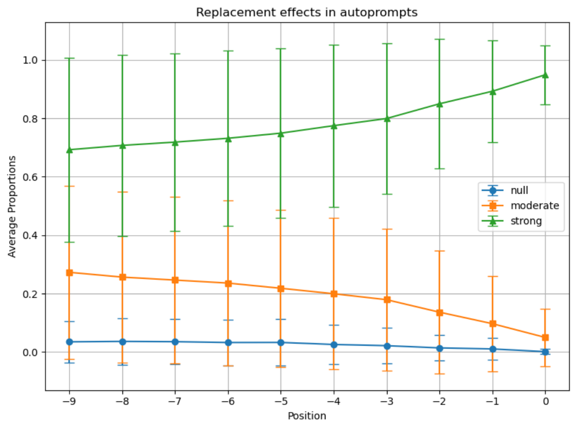

Figure 3: Average proportions of replacement effect types by position on pruned autoprompts, aligned from right (whiskers show standard deviations). *Null*-effect replacements leave the continuation unchanged. *Moderate* replacements have BLEU of at least 0.2. *Strong* replacements have BLEU below 0.2.

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 a catastrophic result on the continuation. The importance of the ending of the autoprompt is further shown by the fact that, as we approach the last position, it is increasingly more difficult to find replacements that do not strongly affect the continuation.

Furthermore, by manually inspecting the cases that lead to only a moderate change in the continuation, we observe that sometimes they show a degree of "compositionality", in the sense that the continuation stays the same except for one or a few tokens that are replaced with new tokens that reflect the meaning of the replacement, and/or drift away from the meaning of the replaced token. Some examples are presented in Table 3.

To make this intuition more quantitative, we ran the following experiment. 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 cases). For these cases, we used FastText (Bojanowski et al., 2017) 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 37% of the cases, whereas the replacement token is more similar to the new continuation in 55% of the cases. We thus conclude that, indeed, there is a tendency for at least this type of replacement to work compositionally, with a small change in the autoprompt leading to a semantically consistent change in the continuation. This, in turn, suggests that autoprompts do not function as unanalyzable holistic wholes, but their "meaning" to the model derives, at least partially, from assembling the meaning of its parts, as with natural language sequences. As the examples show, though, this assembling looks nothing like the one performed by natural language syntax.

4.3 Shuffling autoprompt tokens

The picture we get from the previous studies is one where autoprompts are composed of three types of tokens. A number of tokens are fillers that, being ignored by the LM, can simply be pruned. The final token is extremely important and hard to change, because, in autoregressive prediction, it determines the exact nature of the first token of the continuation, and consequently strongly affects the rest of the continuation. The other non-prunable tokens also have an impact on the continuation, but they seem to rather work as single "keywords" that affect the semantic content of what follows, without forming a tight syntactic bound with each other and what follows.

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

autoprompt:	*continuation:*
cake implies Norman meaning **LOVE/radical** journalism indicated	by the use of the word "**love/radical**" in the title.
s**Dad/meal** ———— Protection Many mans ruggedally understands	the need to protect his **family/food**
Grad^{ OTHERary soldier}\\}$ indicates auxiliary **baggage/work**	**carried/done** by the other soldiers.

Table 3: Example autoprompt token replacements leading to a small, interpretable change in the continuation (replaced/replacement tokens in the autoprompt and changed material in the continuation are highlighted in bold).

on what we just observe, 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 the 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.

The average BLEU when shuffling all tokens is at 0.02 (s.d. 0.03) and at 0.05 (s.d. 0.07) when leaving the last token in its slot. This difference is highly significant (paired t-test, $p < 0.001$).[8] 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 61.5% of these cases, the Pile frequency of the original bigram is larger than that of the inverted one. This suggests that there is at least some tendency towards a natural local ordering among autoprompt tokens.

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% can also be pruned, with 21.9 tokens removed on average. Considering the average token length of the original prompts is 38.57, this means that, strikingly, on average 57% 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 4. Prunable material often consists of modifiers whose removal does not affect the basic syntactic structure of the fragments ("***strategic*** *bomber*", "*section **of the pipeline***", "***replication*** *fork*"...), but this is not always the case, and in many examples pruning turns well-formed sentences into seemingly unstructured token lists or telegraphic texts at best ("*most section, since it*", "*fork mobile day but*"). Still, like in the case of the autoprompts, the coherence of the transition between the prompt and the continuation is generally preserved ("*... bomber Tu, / which was designed...*", "*... since it / was the only one...*").

Table 5 shows the original-prompt tokens that are most typically kept vs. pruned. As for the autoprompts (cf. Table 2 above), the highly prunable tokens consist entirely of common function words and punctuation marks. However, the typically kept tokens tend to also consist of (somewhat rarer) function words and punctuation marks. The presence of quotation marks and brackets in this list should not surprise us, because removing these elements from the prompt will strongly affect the continuation (e.g., without the opening bracket in the prompt, the model might fail to close a parenthetical, producing a completely different continuation). However, what is the crucial distinction between the typically kept vs. pruned function words is not clear to us, and it deserves further investigation in

[8] 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.

prompt:	*continuation:*
… Soviet prototype **strategic** bomber **based on the** Tu **4**,	which was designed to replace the Tu-4.
… most **complex single** section **of the pipeline**, since it	was the only one that was not under contract with Fluor.
… **form**ers (**16–18 year olds**)	were recruited for the performance.
… **replication** fork **and the** mobile **Holl**iday **junction,** but	the structure of the DNA duplex was not known.
… the 74 gun **These**us, provided an escort and	fired a salvo of shells at the enemy's batteries.

Table 4: Randomly selected examples of original prompts with prunable tokens in bold. Only the last 10 tokens of each original prompts are shown. In the first example, the last token of the autoprompt is a comma, which is not pruned. In the third example, the brackets are not pruned, either.

kept:	*pruned:*
"	the
),	,
which	of
was	a
(	.
is	in
)	's
when	on
film	and
with	The

Table 5: Top-10 kept (left) or pruned (right) original prompt tokens ranked by local mutual information.

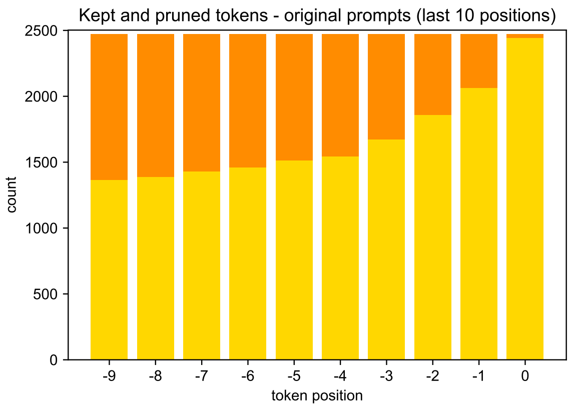

Figure 4: Counts of original prompt tokens that were pruned (dark orange) and kept (yellow) in the last 10 positions, where 0 is the last position.

the future.

Figure 4 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 (cf. Figure 1 above).

Replacement We replicate the token-replacement experiment on the pruned original prompts, obtaining the results summarized in Figure 5, where we used the same BLEU ranges as in Figure 3 above. 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.

Table 6 show 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 their induction process. For example, both "*… citing the "popular adventure book", attempted the first*" and "*… adventure regression first*" trigger the continuation "*ascent of the north face of Mount Everest.*" The last token is preserved, and determines the fact that the continuation will start with a noun. The term *adventure* probably contributes to determine that the continuation is something adventurous, but, as the materials surrounding the token have been deleted, it acts more like a keyword than a proper syntactic element. Finally, the irrelevant inserted token *regression* is ostensibly ignored.

Shuffling Shuffling all tokens of the original prompts after pruning leads to an average BLEU of 0.02 (s.d. 0.03), comparable to what observed for autoprompts. Leaving the last token in place leads to an average BLEU of 0.03 (s.d. 0.05). This small difference is again highly significant (paired t-test, $p < 0.001$), confirming the importance of the last token for the subsequent prediction (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

We show that seemingly opaque, machine-induced prompts possess, to some extent, interpretable properties, such as a strong reliance on the last token,

original: … one of the best examples of American surrealism and *modified:* … one of003 and *continuation:* one of the best films of the 1990s.	
original: … I Ever Wanted, and "Already Gone *modified:* … Ifold, and "Alreadyone *continuation:* " was the first single released from the album.	
original: … citing the "popular adventure book", attempted the first *modified:* adventure regression first *continuation:* ascent of the north face of Mount Everest.	

Table 6: Examples where pruning and replacing a token in an original prompt does not affect the continuation. The *original* row shows the last 10 tokens of an original prompt; the *modified* row shows the equivalent prompt suffix after pruning and replacement.

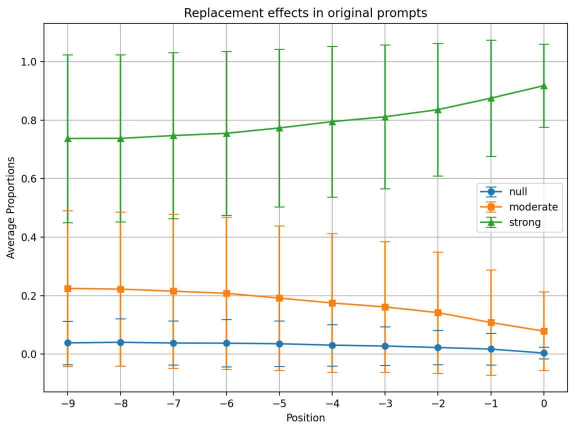

Figure 5: Average proportions of replacement effect types by position on pruned original prompts, aligned from right, limited to the last 10 tokens (whiskers show standard deviations).

the presence of filler tokens that are ignored by the model, and the compositional-like behavior of some keyword tokens. We further observe that some of these properties are also present in natural prompts.

These findings might shed some light on how LMs process language in general. They seem to rely on a simplified model of it, 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 have also been shown to latch onto superficial correlations in the data, leading to poor out-of-distribution generalization (Jo and Bengio, 2017; Ilyas et al., 2019; Yin et al., 2019; Geirhos et al., 2020).

Identifying and characterizing the features that deep learning models respond to are crucial steps in understanding their inner workings and making them more robust. In future work, besides addressing the issues discussed in the Limitations section below, we aim to extend our analysis beyond discrete tokens, focusing either on circuits through mechanistic interpretability methods or on representations using a more top-down approach, such as representation engineering (Zou et al., 2023a).

Limitations

- Due to the time it takes to induce autoprompts with our computational resources, we could only experiment with 3 models, the largest of which has 6.9B 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, and we fixed the number of tokens in the induced prompt to 10. Given that all algorithms we are aware of adopt similar gradient-following methods, and based on qualitative inspections 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.

- Our autoprompts most closely resemble adversarial attack where an obfuscated sequence is used to retrieve one specific piece of information from the LM. However, autoprompts might be also induced for other purposes, such as to improve factual knowledge retrieval when combined with a query sequence (Shin et al., 2020). It remains to be explored if different classes of autoprompts possess significantly different properties.

- We have now a basic understanding of how an autoprompt determines its continuation, but we still need a better characterization of which tokens are more likely to be pruned, and of the means by which randomizing non-last tokens affects the continuation so strongly.

Ethics Statement

If we do not achieve a genuine understanding of how LMs process and generate text, we cannot fully control their behaviour 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.

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, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 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, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 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, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023a. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

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

A 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 7, 8 and 9, respectively.

B Results with other models

B.1 Data-set statistics

Pythia-6.9B As it is very time-consuming to extract autoprompts for this larger model, we limited the data-set to 208 entries. 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).

OLMo-1B The data-set contains 500 entries. The average original prompt length is of 38.4 tokens (s.d. 11.2); that of the continuations is of 8.5 tokens.

B.2 Pruning autoprompts

- Proportion of prunable autoprompts and average (s.d.) tokens pruned:

  Pythia-6.9B 73.2% of autoprompts are pruned, with 2.6 (s.d. 1.6) tokens removed on average.

  OLMo-1B 60.0% of autoprompts are pruned, with 1.9 (s.d. 1.1) tokens removed on average.

- Token pruning distribution by position is shown in Fig. 6 (left: Pythia-6.9B; right: OLMo-1B).

B.3 Replacing autoprompt tokens

For OLMo, 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 Figure 7 (left: Pythia-6.9B; right: OLMo-1B).

autoprompt: Processing<EOT> **Launch/life**},$ Watson saw1949mL bigger wing *continuation:* , a new engine, and a new propeller.
autoprompt: really **dwarfs/black**ados haben send extraordinarily overwhelmingly excessive$]{} abundance *continuation:* of heavy elements in their atmospheres.
autoprompt: approachò keep**_ mystery,. novel **reportedly/council**_** enjoys *continuation:* keeping the reader in suspense.
autoprompt: impressive character<EOT> galactic Avengers drops<EOT>**<EOT>/further** comics collected *continuation:* in the Marvel Cinematic Universe.
autoprompt: champ241<EOT> GE 1870ista“ **Japanese/rick** dance art *continuation:* form that was popular in the late 19th century.

Table 7: Randomly selected *null-effect* replacement examples. Replaced tokens and replacements are separated by “/” and in bold.

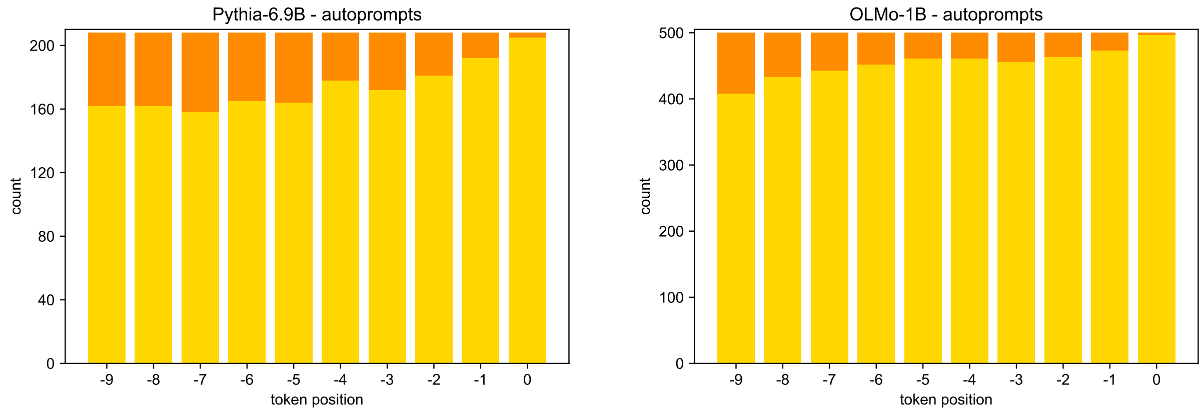

Figure 6: Counts of pruned (dark orange) and kept (yellow) tokens in the autoprompts by position, in Pythia-6.9B (left) and OLMo-1B (right).

B.4 Shuffling autoprompt tokens

Average BLEU (s.d.) when shuffling all tokens vs. keeping last token fixed:

Pythia-6.9B shuffling all tokens: 0.03 (s.d. 0.04); keeping last fixed: 0.06 (s.d. 0.11); paired t-test significant at $p < 0.001$ (also if last-fixed is compared to random-non-last-fixed).

OLMo-1B shuffling all tokens: 0.02 (s.d. 0.01); keeping last fixed: 0.04 (s.d. 0.04); t-test significant at $p < 0.001$ (also if last-fixed is compared to random-non-last-fixed).

B.5 Making prompts more autoprompt-like

Pruning

- Proportion of prunable prompts and average (s.d.) tokens pruned:

  Pythia-6.9B 99.5% of the original prompts are pruned, and the average number of pruned tokens is 23.8 (s.d. 13.2); 95.7% of the pruned prompts have at least one pruned token among the last 10.

  OLMo-1B 100% of the original prompts are pruned, and the average number of pruned tokens is 23.4 (s.d. 12.3); 97% of the pruned prompts have at least one pruned token among the last 10.

- Token pruning by position is reported in Figure 8 (left: Pythia-6.9B; right: OLMo-1B).

Replacement Proportions of replacement effect type by position are reported in Figure 9 (left: Pythia-6.9B; right: OLMo-1B).

Shuffling Average BLEU (s.d.) when shuffling all tokens vs. keeping last token fixed:

Pythia-6.9B shuffling all tokens: 0.02 (s.d. 0.02); keeping last fixed: 0.03 (s.d. 0.05); paired t-test significant at $p < 0.001$ (also if last-fixed is compared to random-non-last-fixed).

OLMo-1B shuffling all tokens: 0.02 (s.d. 0.03); keeping last fixed: 0.03 (s.d. 0.05); paired t-test significant at $p < 0.001$ (also if last-fixed is compared to random-non-last-fixed).

autoprompt: cancer<EOT> están<EOT> **Card/Allen**ropical frig Jamaica describes humid *original continuation:* tropical climate of the Caribbean. *modified continuation:* tropical climate of the island of Jamaica. *modified continuation BLEU:* 0.36
autoprompt: hired locals budget,** climbing destinations Pull**Town/LR** oldest especially *original continuation:* popular destination for climbers. *modified continuation:* popular with climbers. *modified continuation BLEU:* 0.23
autoprompt: schenken clergy?? KosovoABA<EOT> pledge **regarding/loop** your constant *original continuation:* support of the Albanian Orthodox Church. *modified continuation:* support for the Albanian Orthodox Church in Kosovo. *modified continuation BLEU:* 0.43
autoprompt: <EOT>ITAL**<EOT>/Angeles**ño<EOT> Denote 0̆415perorachusetts as *original continuation:* the capital of the United States. *modified continuation:* the capital of the United States of America. *modified continuation BLEU:* 0.61
autoprompt: everyoneDaily **tracking/idea**>{{Self calendar??Its...exceedingly *original continuation:* difficult to keep track of everything. *modified continuation:* difficult to keep track of all the things that I want to do. *modified continuation BLEU:* 0.28

Table 8: 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.

C 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.

D 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

AI use disclosure: we used Copilot and ChatGPT for assistance in code writing and in manuscript typesetting.

autoprompt: laccyt<EOT>ALTHICAN Brown jazz**<EOT>/ST**an indispensable *original continuation:* part of the American folk tradition. *modified continuation:* to the development of the American style of jazz. *modified continuation BLEU:* 0.11
autoprompt: Off"arn careers Birmingham lion2005 **ballet/mediated** Barry starred *original continuation:* in the West End production of The Lion King. *modified continuation:* in the film, which was released in the United States in 2005. *modified continuation BLEU:* 0.03
autoprompt: Interview'————" Heisenberg **masterpiece/poverty** Summer Fire books brand *original continuation:* new introduction by the author. *modified continuation:* new. *modified continuation BLEU:* 0.02
autoprompt: tonnes **Catholics/i** Which Esc<EOT> have syn survived many factions *original continuation:* including the Roman Catholic Church. *modified continuation:* and are still in use. *modified continuation BLEU:* 0.04
autoprompt: >] Publishingigenous Cemetery Once Anventh **century/losing** coffin had *original continuation:* been buried in a pagan burial ground. *modified continuation:* been found in the woods, the family decided to bury it in the family plot. *modified continuation BLEU:* 0.01

Table 9: 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 "?".

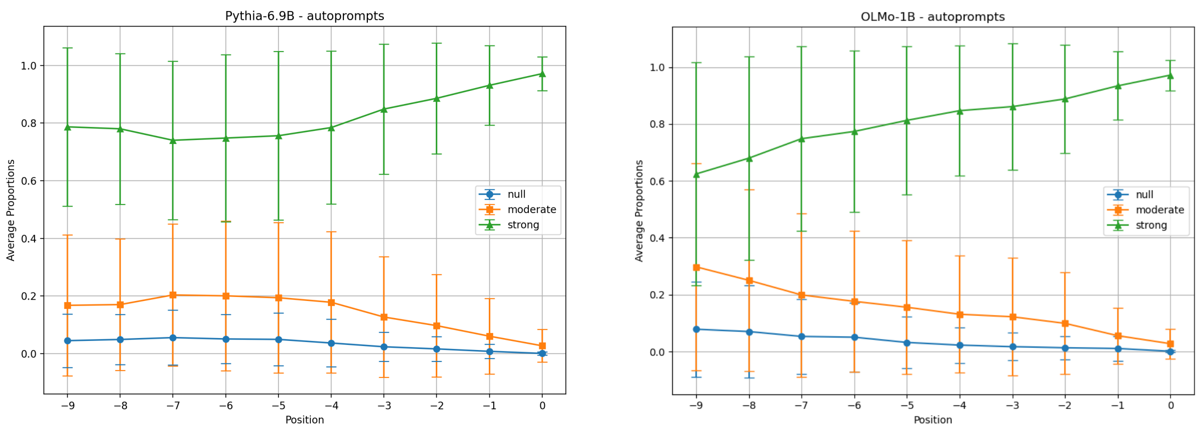

Figure 7: Average proportions of replacement effect types in the autoprompts by position, aligned from right for Pythia-6.9B (left) and OLMo-1B (right) (whiskers show standard deviations). *Null*-effect replacements leave the continuation unchanged. *Moderate* replacements have BLEU of at least 0.2. *Strong* replacements have BLEU below 0.2.

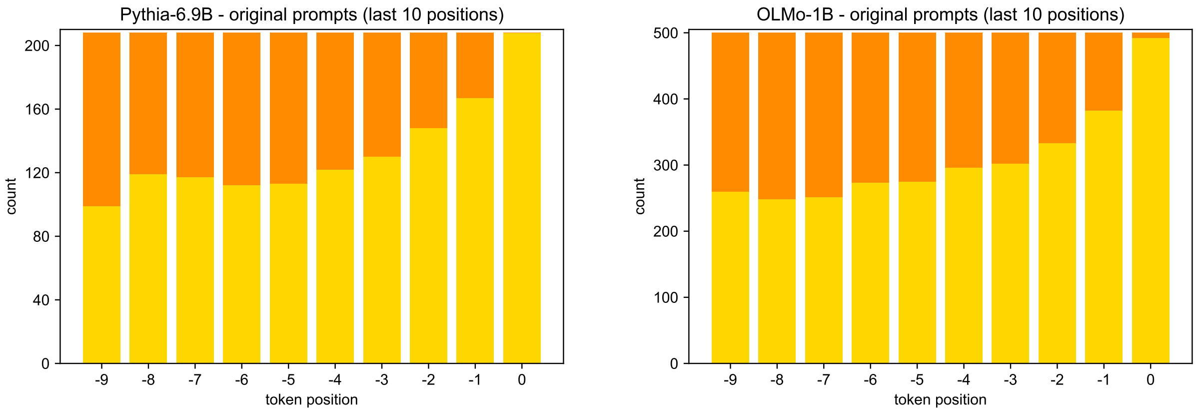

Figure 8: Counts of pruned (dark orange) and kept (yellow) tokens in the original prompts, by position, in Pythia-6.9B (left) and OLMo-1B (right).

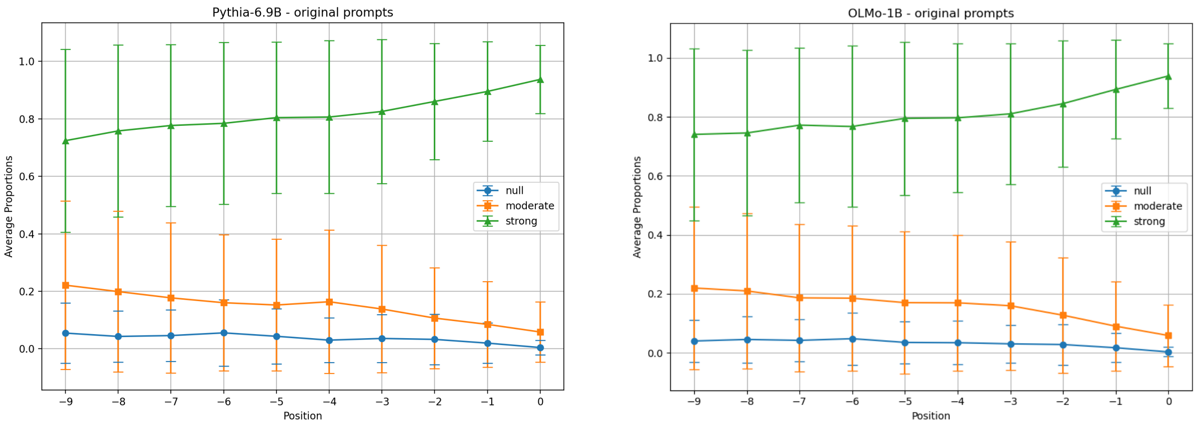

Figure 9: Average proportions of replacement effect types in the original prompts by position, aligned from right (whiskers show standard deviations) for Pythia-6.9B (left) and OLMo-1B (right). *Null*-effect replacements leave the continuation unchanged. *Moderate* replacements have BLEU of at least 0.2. *Strong* replacements have BLEU below 0.2.