How much do language models copy from their training data? Evaluating linguistic novelty in text generation using RAVEN

R. Thomas McCoy,[*] [1] **Paul Smolensky,**[2,3] **Tal Linzen,**[4] **Jianfeng Gao,**[2] **Asli Celikyilmaz**[†] [5]
[1]Princeton University [2]Microsoft Research
[3]Johns Hopkins University [4]New York University [5]Meta AI
tom.mccoy@princeton.edu, psmo@microsoft.com, linzen@nyu.edu, jfgao@microsoft.com, aslic@meta.com

Abstract

Current language models can generate high-quality text. Are they simply copying text they have seen before, or have they learned generalizable linguistic abstractions? To tease apart these possibilities, we introduce RAVEN, a suite of analyses for assessing the novelty of generated text, focusing on sequential structure (n-grams) and syntactic structure. We apply these analyses to four neural language models trained on English (an LSTM, a Transformer, Transformer-XL, and GPT-2). For local structure—e.g., individual dependencies—text generated with a standard sampling scheme is substantially less novel than our baseline of human-generated text from each model's test set. For larger-scale structure—e.g., overall sentence structure—model-generated text is as novel or even more novel than the human-generated baseline, but models still sometimes copy substantially, in some cases duplicating passages over 1,000 words long from the training set. We also perform extensive manual analysis, finding evidence that GPT-2 uses both compositional and analogical generalization mechanisms and showing that GPT-2's novel text is usually well-formed morphologically and syntactically but has reasonably frequent semantic issues (e.g., being self-contradictory).

1 Introduction

There are many abstract properties that characterize well-formed text, from grammatical properties (e.g., subject-verb agreement) to discourse properties (e.g., coherence). How can we tell which of these properties have been learned by a language model (LM)? One popular approach is to analyze text generated by the LM (Dai et al., 2019; Zellers et al., 2019; Brown et al., 2020; Zhang et al., 2022). The assumption underlying this approach is that, if the text displays a particular linguistic property (e.g., coherence), then the LM must have captured that property.

We argue that this approach has an important flaw: The generated text could have been copied from the LM's training data, in which case it does not provide clear evidence for any linguistic abstractions. For example, suppose an LM achieves coherence by copying a paragraph from its training set. In this case, the entity that deserves credit for being coherent would not be the LM but rather the human who originally wrote that paragraph. To address this concern, it is important to check whether LM-generated text duplicates from the training data. That is, we argue that LM-generated text must have two traits to be clear evidence that the LM has learned some abstraction A:

(1) **Quality:** The text must be well-formed with respect to A.

(2) **Novelty:** The text must not have been copied from the training data.

Much prior work has discussed how to evaluate various aspects of quality (Gatt and Krahmer, 2018; Celikyilmaz et al., 2020). Our central point is that novelty also merits careful consideration.

In this work, to quantify the novelty of generated text, we introduce a suite of analyses called RAVEN (**RA**ting **VE**rbal **N**ovelty).[1,2] These analyses cover both sequential structure (n-grams)

[*] Work done while at Microsoft Research and Johns Hopkins University.

[†] Work done while at Microsoft Research.

[1] https://github.com/tommccoy1/raven

[2] *Verbal* here uses its broad definition of "linguistic" rather than the narrow definition of "verb-related." This acronym refers to "The Raven" by Edgar Allan Poe, in which the narrator encounters a mysterious raven which repeatedly cries out, "Nevermore!" The narrator cannot tell if the raven is simply repeating something that it heard a human say, or if it is constructing its own utterances (perhaps by combining *never* and *more*)—the same basic ambiguity that our paper addresses. This acronym is also a nod to Bender et al.'s (2021) comparison of LMs to another utterance-memorizing bird, the parrot.

and syntactic structure. We apply these analyses to text generated by an LSTM, a Transformer, Transformer-XL, and all 4 sizes of GPT-2 (the largest LM for which we had access to the training data). Because there are many ways to generate text from LMs, we test 12 generation methods and 4 prompt lengths. As a baseline, we also analyze human-generated text from each model's test set.

Summary of findings: We find that models display novelty for all aspects of structure that we analyze: they generate novel n-grams, novel morphological combinations, and novel syntactic structures. For instance, GPT-2 coins several types of novel words, including inflections (e.g., *Swissified*) and derivations (e.g., *IKEA-ness*), and 74% of sentences generated by Transformer-XL have a syntactic structure that no training sentence has. Thus, **neural language models do not simply memorize; instead they use productive processes that allow them to combine familiar parts in novel ways**. Nonetheless, when considering small n-grams, these models are less novel than the baseline. For example, for each model, the baseline human-generated text has 1.4 to 3.3 times as many novel bigrams as the model-generated text does. For n-grams larger than 5-grams, models are *more* novel than the baseline, but they still occasionally copy extensively: GPT-2 sometimes duplicates training passages that are over 1,000 words long.

Significance of findings: Our main finding is that LMs do not copy much. This finding is a welcome one because it shows that a confound present in many prior analyses (the possibility that LMs might mainly be copying) is unlikely to be a major concern in practice. On the other hand, the fact that LMs sometimes copy substantially shows that it is not safe to assume that a particular piece of generated text is novel—we must specifically check for novelty if we want to draw general conclusions about an LM from text it has generated.

Beyond these broad takeaways, the specific types of novelty illuminated by our analyses provide evidence that several important linguistic abstractions have been captured by the LMs we investigated. These abstractions include:

- Constituency structure (§6)
- Dependency structure (§6)
- Plural and possessive morphology (§7.1)
- Spelling-change rules (§7.1)
- Subject-verb agreement (§7.2)
- Incrementation and ordering (§7.2)
- Novelty (i.e., in addition to the fact that *we* study novelty, there is evidence that *GPT-2* encodes whether its text is novel, as shown by a tendency to enclose novel words in quotation marks: §7.2)

Our analyses also revealed two areas that were not well-captured by GPT-2, namely:

- Acronym structure (§7.2)
- The relation between morphology and meaning (§7.3)

Finally, our results provide evidence that GPT-2 uses two distinct types of generalization: compositional and analogical generalization (§7.4).

Though many of the abstractions that we study have been discussed in prior analyses of LMs, the only one for which prior work has enforced novelty is subject-verb agreement (Wei et al., 2021). Overall, by evaluating novelty, we gain a new window into how models have or have not succeeded at generalizing beyond their experience.

2 Background

Memorization and copying: The concern that LMs might copy extensively from their training data is widely recognized. For example, Bender et al. (2021) liken LMs to "stochastic parrots" that simply memorize seen examples and recombine them in shallow ways.[3] On the other hand, some prominent examples of LM-generated text have led others to assume that LMs are not heavily reliant on copying. For example, GPT-2 generated a story about scientists discovering talking unicorns in the Andes, which seems unlikely to have arisen via copying (Radford et al., 2019). Our goal is to adjudicate between these conflicting viewpoints.

It is clear that neural networks are capable of extensive memorization: they can memorize randomly-labeled examples (Zhang et al., 2021a) and can reveal training data when subjected to adversarial attacks (Shokri et al., 2017; Carlini et al., 2019, 2021, 2023). We study copying in text generated under standard, non-adversarial conditions, a topic which a few other recent works have

[3]This view even extends beyond the research community: A 2021 webcomic by Zach Weinersmith (https://languagelog.ldc.upenn.edu/nll/?p=52293) includes an AI system exclaiming, "The fools don't realize how many of my coherent phrases are verbatim from training data!"

touched on. Brown et al. (2020, Section 8.2), Lee et al. (2022b), and Kandpal et al. (2022) study copying of large n-grams in Transformer LMs, while Chen et al. (2021) and Ziegler (2021) look at copying of large n-grams in the code-generating model Codex. We perform a more comprehensive analysis of duplication: We look across the full range of n-gram sizes and analyze a range of architectures and generation methods. Beyond n-grams, we also evaluate copying of other linguistic structures (e.g., dependency arcs). Thus, we study linguistic generalization, while past work studied concerns of data privacy (Carlini et al., 2019) and plagiarism (Lee et al., 2022a).

Evaluating text quality: Prior work has proposed many approaches for evaluating the quality of generated text. Some approaches provide a single holistic score (Zhang et al., 2020a), while others give scores that focus on specific properties (Dou et al., 2022) such as fluency (Mutton et al., 2007) or factual accuracy (Kryściński et al., 2020).

Our focus is novelty rather than quality. The previously-studied attribute that is most similar to novelty is *diversity* (Zhu et al., 2018; Hashimoto et al., 2019): can a model generate a diverse range of output sentences? Like novelty, diversity is rooted in differences between pieces of text. Despite this superficial similarity, novelty and diversity are distinct. Novelty covers how the generated text differs from the training set, while diversity covers how the generated text is different from other generated text. A model could be diverse but not novel (by copying a diverse set of training sentences), or novel but not diverse (by repeatedly generating the same novel sentence).

Much discussion about evaluating LMs focuses on whether they *understand* language (Bender and Koller, 2020; Marcus, 2020), whereas we assess the novelty of surface text. Thus, our main analyses only test whether models have abstractions governing form (e.g., syntax), not meaning.

Our focus on considering a model's training data when evaluating that model fits with a broader trend of tracing model behavior back to the training set. Other papers in this direction include Akyurek et al. (2022), Han and Tsvetkov (2022), and Elazar et al. (2022).

3 Motivation and approach

Motivation: The analyses in RAVEN are inspired by a scientific question: To what extent do LMs have generalizable linguistic abilities? This question motivates our focus on novelty because only novel text can illustrate linguistic generalization. There may be some practical use cases for which novelty is not important—but for answering our scientific question, and for working toward general-purpose LMs that can handle unfamiliar situations (LeBrun et al., 2022), novelty is crucial.

Approach: We generate many samples of text from LMs and then evaluate how novel the text is. We assess novelty for two types of structure: n-grams and syntactic structure. We count a generated structure as duplicated if it appears in the training set or the context (the concatenation of the prompt and the text that the LM has already generated based on the prompt); otherwise, it is novel.

Copying is not necessarily undesirable (Khandelwal et al., 2020): some long n-grams, such as book titles, might reasonably be duplicated from the training set. To contextualize a model's degree of duplication, we compare the model-generated text to human-generated text from the model's (in-distribution) test set, which gives a baseline for how much duplication is expected in the model's training domain. If the model is at least as novel as the baseline, we conclude that it is not copying excessively. Two prior papers (Pannitto and Herbelot, 2020; Meister and Cotterell, 2021; Yamakoshi et al., 2022) have also analyzed models' linguistic abilities by comparing model-generated text to human-generated text, but none of these focused on novelty.

4 Experimental details

Models: To compare architectures in a controlled way, we used three models trained on the same dataset, namely Wikitext-103 (Merity et al., 2017). Wikitext-103 is a collection of English Wikipedia articles tokenized at the word level. Its training set contains 103 million words. Holding this training set constant, we compared the LSTM (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017), and Transformer-XL (TXL; Dai et al., 2019) architectures, chosen because they give examples of the two most prevalent types of processing in language modeling: recurrence (used in the LSTM) and self-attention (used in the Transformer), with TXL using both mechanisms.

In addition to these systematic analyses, we also analyzed GPT2-XL, the largest size of GPT-2 (Radford et al., 2019), as an example of a larger-scale Transformer LM (GPT-2 was the model with the largest training set that we could gain access to). Unlike our other models, GPT-2 is trained on the WebText corpus, which is constructed from webpages linked to on Reddit, mainly in English. GPT-2 also differs from our other models in its tokenization scheme: All our other models use word-level tokenization (in which each token is a full word), but GPT-2 uses a subword tokenization scheme (Sennrich et al., 2016). The WebText training corpus contains 7.7 billion words, making it much larger than Wikitext-103. For more details about each model, see the online supplement.[4] Throughout this paper, unless otherwise stated, *GPT-2* refers to GPT2-XL, following Radford et al.'s terminology.

Prompts: To generate text from a model, we input a prompt drawn from that model's test set, which comes from the same distribution as its training set. For Wikitext-103, we use 1000 prompts of length 512 words and have models generate 1000 words following the prompt. For WebText, we use 1000 prompts of length 564 subword tokens, and have models generate 1100 subword tokens; these numbers are 1.1 times the corresponding Wikitext-103 numbers because there are about 1.1 subword tokens per word in WebText. As our baseline human-generated text, we use the text that follows the prompt in the corpus.

Decoding method: top-40 sampling: As its prediction about which word will appear next, a language model outputs a probability distribution over the vocabulary. There are many ways to select a word to generate from this distribution, which are called *decoding methods*.

A tempting choice for a decoding method would be pure sampling, in which we simply sample from the model's distribution. However, when evaluating a model's novelty, an important consideration is that novelty is not always positive: a model that generates random nonsense would be highly novel. Thus, we want a decoding method that gives high-quality text, because novelty is only positive when accompanied by high quality. Pure sampling is not suitable for this purpose because it yields "incoherent gibberish" rather than high-quality text (Holtzman et al., 2020).

Instead, the main decoding scheme that we use is top-k sampling with $k = 40$, where the model's distribution is truncated to the 40 highest-ranked words then renormalized and sampled from. We chose top-40 sampling because it is what Radford et al. (2019) used for GPT-2 and what Dai et al. (2019) used for TXL; because this method was selected by the creators of these models, we can be reasonably confident that it produces high-quality text from these models. In addition, using the same decoding method as prior work facilitates comparisons to that work, which is important for our goal of assessing whether prior results might have been confounded by a lack of novelty. For consistency, we use this same decoding scheme for our LSTM and Transformer, for which there is no established decoding method. For experiments with other decoding methods, see Section 5.2.

[4] https://arxiv.org/abs/2111.09509

5 N-gram novelty

We first investigate novelty at the level of n-grams, where an n-gram is a sequence of n words.

Motivation: Many prior papers (e.g., Dai et al., 2019; Zhang et al., 2022) use holistic demonstrations of the high quality of LM-generated text as evidence for the LM's overall strength. As discussed in Section 1, these conclusions would be undermined if the generated text were copied from the training data. One goal of our work is to test whether this concern is borne out in practice. In this section, we use analyses at the n-gram level as holistic measures of novelty, to match the holistic nature of the relevant prior demonstrations of quality. In later sections, we will conduct analyses that target specific linguistic properties.

5.1 How often are generated n-grams novel for various values of n?

We first discuss n-gram novelty for two decoding methods: top-40 sampling and pure sampling. As discussed in Section 4, top-40 sampling follows the precedent of prior literature, while pure sampling is rarely used but is interesting because it shows the LM's unaltered distribution. The next section then gives a more thorough investigation of a range of decoding methods.

Findings: For $n > 6$, n-grams generated by LMs are almost always novel, both for top-40 sampling and pure sampling. For smaller n-grams, these decoding methods diverge: Small n-grams generated with top-40 sampling are much less novel than small n-grams in the human-generated baseline text, but small n-grams generated with pure sampling are more novel than the baseline.

Details: We tokenize all text with the Moses tokenizer (Koehn et al., 2007), which treats punctuation marks as separate words but otherwise does not separate words into smaller units, and we then analyze n-grams formed from these tokens. Figure 1 shows the proportion of generated n-grams that are novel, for values of n from 1 to 10. We first note that the models are not merely copying: for all models, for n-grams of size 5 or larger, the majority of n-grams are novel.

We can obtain a more nuanced view by comparing the models to the baseline of text drawn from each model's test set. When using pure sampling, models are more novel than the baseline across n-gram sizes. When using top-40 sampling, small and large n-grams differ: For small n-grams ($n < 6$), models are less novel than the baseline. For instance, with Wikitext-103, the baseline has 6% of its bigrams being novel, while the models have 2% to 3% novelty; for trigrams, the baseline has 31% novelty while models have 17% to 22%. Thus, models are conservative at the small scale when using top-40 sampling, rarely deviating from previously-seen bigrams and trigrams. However, for larger n-grams ($n > 6$), the models are *more* novel than the human-generated baseline. Thus, at a larger scale, even when using top-40 sampling, models cannot be described as excessively copying n-grams they have seen before.

The insets of Figures 1a and 1b show that the LSTM and TXL are less novel for small n-grams than the Transformer. We conjecture the following explanation: Recurrence creates a recency bias (Ravfogel et al., 2019) which makes models likely to condition their predictions heavily on immediately preceding tokens, biasing them to memorize bigrams and trigrams. The LSTM and TXL both incorporate recurrence, whereas the Transformer does not, explaining why the Transformer duplicates the least of the three.

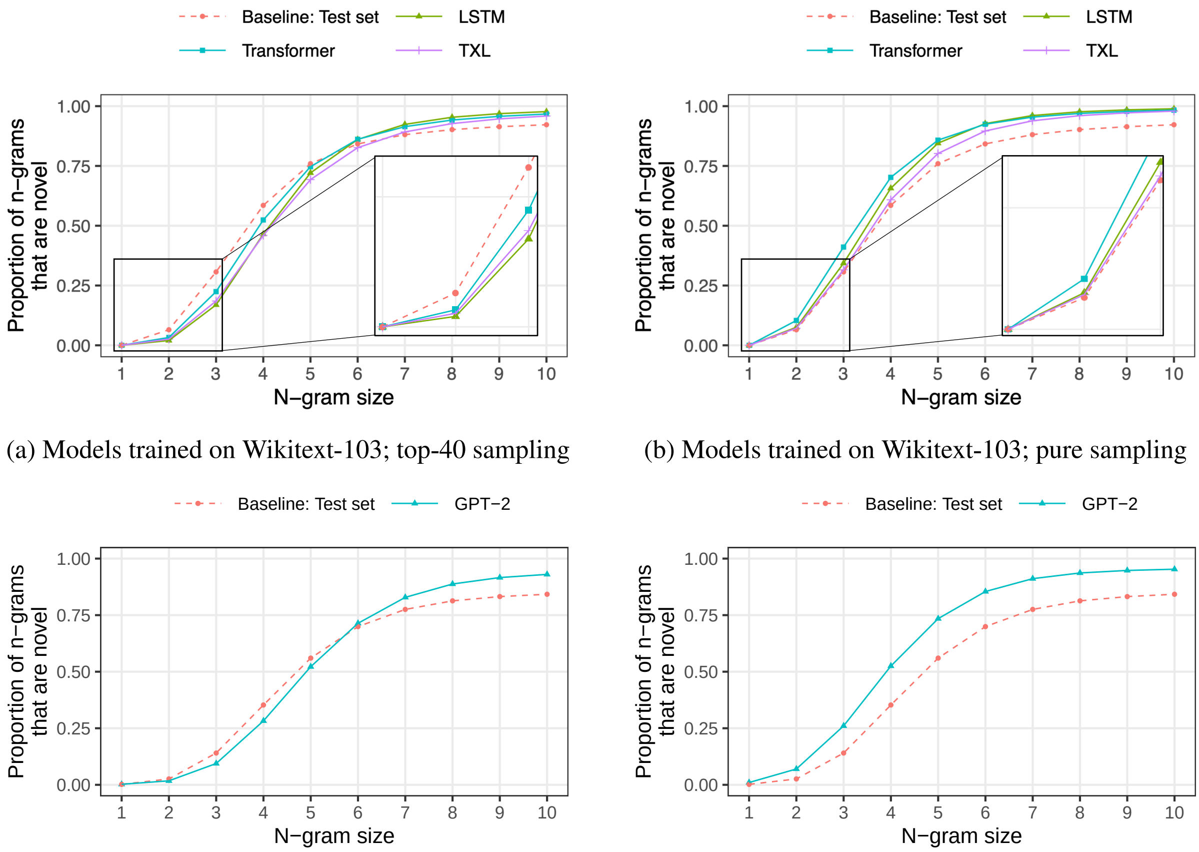
5.2 How is novelty related to the decoding scheme and the generated text's quality?

Findings: Changing decoding parameters can substantially alter a model's novelty: the novelty can be increased by increasing p in top-p sampling, k in top-k sampling, or the temperature. However, all modifications that increase the novelty of generated text also decrease the quality.

Details: To get a single number that summarizes novelty, we use a new metric called the *pointwise duplication score*: Each token gets a score quantifying the extent to which it duplicates previously-seen text. This score is equal to the size of the smallest novel n-gram that ends with this word. For example, if the word is the end of a novel 4-gram (e.g., *these <u>rules will not **be**</u>*), but all smaller n-grams ending with the word were duplicated (*will not **be***, *not **be***, and ***be***), then the pointwise duplication score is 4. The overall score is the average across tokens. A downside of this basic score is that the average can be heavily influenced by high values arising from the rare instances of long copied passages. To address this concern, we truncate each token's score at 5 before averaging (see the online supplement for untruncated results).

Using this score, we investigated a range of decoding methods. The online supplement shows in detail the effects of varying commonly-used decoding parameters. With top-k sampling (truncating the distribution to the k most probable tokens before sampling), increasing k also increases novelty. With top-p sampling (truncating the distribution to the the top p probability mass before sampling; Holtzman et al., 2020), increasing p increases novelty. When using a temperature (which scales words' scores before taking the softmax), increasing the temperature increases novelty. All of these trends make intuitive sense: a small k, p, or temperature upweights the head of the model's distribution, and it makes sense that statistical learners would assign higher probability to things they have seen than things they have not, which would lead to the head of a model's distribution being less novel than the tail.

Could we make models perfectly novel just by changing the decoding scheme? Unfortunately, the decoding methods that increase novelty also decrease quality. Measuring quality is challenging; ideally we would use human evaluations, but that is beyond the scope of this project because we have 48 conditions to evaluate (4 models with 12 decoding schemes). Instead, we use perplexity as a proxy for quality, under the assumption that high-quality text should have a low perplexity. This assumption is certainly imperfect: text can have a low perplexity for degenerate reasons such as being repetitive (Holtzman et al., 2020). Nonetheless, it can still give a rough initial sense of general trends. We use GPT-2 to measure the



(a) Models trained on Wikitext-103; top-40 sampling

(b) Models trained on Wikitext-103; pure sampling

(c) Models trained on WebText; top-40 sampling

(d) Models trained on WebText; pure sampling

Figure 1: Novelty of n-grams generated by LMs using top-40 sampling (which is the approach used in relevant prior literature) and pure sampling (which uses a model's unaltered distribution but is not standard because it produces low-quality text). As baselines, we use text drawn from models' test sets.

perplexity of text generated by the LSTM, Transformer, and TXL; we use TXL to measure the perplexity of GPT-2 text.

Figure 2 shows a clear tradeoff between novelty and quality. None of the models trained on Wikitext do as well as the baseline at managing this tradeoff. However, a model's perplexity does not entirely determine its level of novelty: Both Transformer architectures do better at this tradeoff than LSTMs, showing that it is possible to improve on this tradeoff using architectural innovations.

In contrast to the Wikitext-103 models, GPT-2 performs similarly to the baseline at the quality-novelty tradeoff. The GPT-2 decoding scheme that comes closest to the baseline is top-p decoding with $p = 0.95$; this achieves a perplexity of 93.7 (baseline: 89.4) and a truncated pointwise duplication score of 4.41 (baseline: 4.47). Why does GPT-2 (with the right decoding scheme) outperform the Wikitext-103 models at matching the quality and novelty of its baseline? It is unlikely that architecture is the reason because GPT-2 is similar in architecture to the Wikitext-103 Transformer. Although GPT-2 is our largest model, we also doubt that model size is the explanation: GPT-2 Small shows similar results even though it is smaller than TXL. It may be that training set size is the key factor, as WebText is much larger than Wikitext-103. Alternatively, the WebText baseline might be easier to meet than the Wikitext one, because the generic Internet text in WebText is generally lower-quality than the curated articles in Wikitext-103, meaning that the level of quality required to match the WebText baseline is lower than the level required to match the Wikitext baseline.

For the rest of this paper, all results are with top-40 sampling, for the reasons given in Section 4.

5.3 Do models ever duplicate large n-grams?

Finding: All models occasionally duplicate training set passages that are 100 words long or longer.

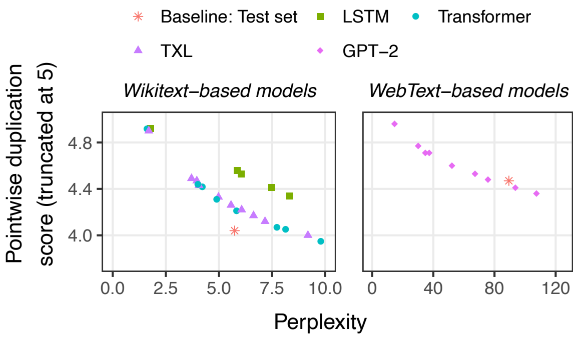


Figure 2: Manipulations of the decoding scheme that result in higher-quality text (i.e., lower perplexity; x-axis) also result in decreased novelty (i.e., a greater degree of duplication; y-axis). Each point shows a different decoding scheme.

Details: Models rarely duplicate n-grams larger than 10 tokens; across models, less than 5% of 10-grams are duplicated from the training set. However, there are occasional exceptions where models duplicate extremely long sequences. For instance, in our GPT-2 generated text, there are several cases where an entire generated passage (over 1,000 words long) appears in the training set. To refer to these extreme cases, we use the term *supercopying*, which we define as the duplication of an n-gram of size 100 or larger. See the online supplement for examples of supercopied text.

What causes supercopying? We hypothesize that models supercopy passages that appear multiple times in the training set. For instance, the Wikitext-103 training set contains 159 articles about instances of The Boat Race, a rowing competition: "The Boat Race 1861," "The Boat Race 2002," etc. These articles are formulaic, with many sentences repeated across articles, and some of the n-grams that were supercopied are indeed from these repetitive articles; e.g., the 100-gram in the supplement that was generated by all 3 Wikitext-103 models occurs 56 times in the training set. More generally, supercopied 100-grams appear, on average, over 10 times in the training set, whereas randomly-selected 100-grams typically appear only once. This is consistent with the findings of Lee et al. (2022b) and Ziegler (2021) that duplicated text tends to be common. Carlini et al. (2021) found that text can be extracted even if it only occurred once, but they used an adversarial method that deliberately tries to extract training data, instead of freely generating text.

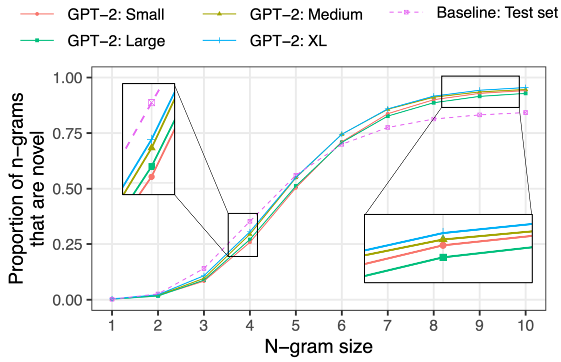


Figure 3: Effect of model size.

5.4 How does model size affect novelty?

Finding: Model size does not have a clear effect on novelty.

Details: It seems possible for model size to affect novelty in either direction. Larger models might be less novel due to having a greater capacity to memorize. On the other hand, larger models are generally stronger (Kaplan et al., 2020), which might include a greater ability to be novel.

Figure 3 shows the level of duplication observed for the 4 different sizes of GPT-2 (all using top-40 samping). There is not a clear, consistent effect of size. Across the various n-gram sizes, the most novel model is GPT-2 XL; however, GPT-2 Medium is more novel than GPT-2 Large, so it is not the case that larger models are always more novel than smaller models (or vice versa).

5.5 Other n-gram analyses

Additional analyses are in the online supplement. We find that prompt length does not have a clear effect on novelty; novelty is influenced by position within the generated text for some models, but the effect is small; and our novelty results do not change much if we only consider duplication from the training set rather than duplication from the context and/or training set.

6 Syntactic novelty

Motivation: We have seen that models display some novelty. How deeply does their novelty extend? Are they just inserting words into memorized templates, or performing deeper syntactic composition? Prior work has shown that the predictions of neural LMs have high **quality** with respect to syntax (Hu et al., 2020; Zhang et al., 2021b); e.g., the annotators in Dou et al. (2022)

marked less than 3% of LM-generated tokens as having grammatical errors. Here we evaluate syntactic **novelty** to address the possibility that the syntactic success of LMs is driven by memorization rather than by generalizable abstractions.

Findings: At the level of global sentence structure, models show a high degree of syntactic novelty, with the majority of generated sentences having an overall syntactic structure that no training sentence has (Table 1). Models also display some novelty for local structure (e.g., individual dependency arcs), but they have much less local novelty than the baselines do. Paired with the syntactic *quality* shown in prior work, the syntactic *novelty* in our analyses is evidence that the LMs we analyzed have captured abstract syntactic structure.

Details: We parsed our generated text and our models' training data using state-of-the-art constituency (Kitaev and Klein, 2018) and dependency (Zhang et al., 2020b) parsers. We then evaluated novelty for 7 aspects of syntax.

Though current parsers perform well, they are not perfect, so we cannot completely trust their output. This is particularly a problem because the cases that are important to us (novel ones) are especially likely to confuse parsers. To address this issue, we manually analyzed the examples identified as novel to estimate the parsers' error rates (details are in the online supplement). We concluded that 4 of the 7 attributes that we analyzed were handled accurately enough by the parsers for us to report numerical results, which are in Table 1. Here is a description of these attributes:[5]

- **POS sequence:** the sequence of part-of-speech tags for the words in the sentence.
- **Parse structure:** the sentence's constituency tree minus the leaves (the words).
- **Labeled dependency arc:** a 3-tuple of a dependency relation (e.g., *nsubj*) and the two words that hold that relation.
- **Dependency role:** a 3-tuple of a word, a dependency relation that the word is part of, and the word's position in that relation; e.g., "*watch* as the head of an *nsubj* relation."

These attributes give a window into whether models have captured compositional syntactic structure: Each attribute is composed of simpler units (e.g., a parse structure is composed of sub-trees, and a dependency arc is composed of the elements in its 3-tuple). Thus, producing novel examples for these attributes requires compositional generalization (combining familiar parts in novel ways).

	POS seq.	Parse struct.	Dep. arcs	Dep. roles
Wiki baseline	0.75	0.76	0.13	0.0050
LSTM	0.77	0.78	0.07	0.0015
Transformer	0.75	0.75	0.08	0.0024
TXL	0.74	0.74	0.07	0.0021
Web baseline	0.59	0.61	0.05	0.0015
GPT-2	0.62	0.64	0.03	0.0007

Table 1: Syntactic novelty. Abbreviations: *seq*=sequence; *dep*=dependency; *struct*=structure.

For POS sequences and parse structures, there is a high degree of novelty: across all models and baselines, the majority of sentences have an overall structure that no training sentence has. In addition, there is little difference between the models and the baselines. For the more local structure of dependency arcs and dependency relations, the baselines are far more novel than the models.

These syntactic findings are similar at a high level to our n-gram results, which showed that models are less novel than the baseline for local structure (small n-grams) but more novel than the baseline for larger-scale structure (large n-grams). To expand on this parallel, we considered dependency paths of varying lengths, analogous to n-grams of varying sizes. We define a dependency path as the labeled path in a dependency tree from a word to any of its ancestors or the root. Some example paths in Figure 4 are [*dog*], [dog_{nsubj}, *barked*], and [dog_{nsubj}, $barked_{root}$, *ROOT*], which have lengths 1, 2, and 3 (a length-2 path is equivalent to a dependency arc). Dependency path novelty (Figure 5) displays trends similar to those for n-gram novelty (Figures 1a and 1c): For short paths, models show little novelty and are less novel than the baseline, but for longer paths they are almost always novel and are more novel than the baseline. These results corroborate the general conclusion that models using top-40 sampling are rarely novel at small scales but usually novel at medium or large scales.

See the online supplement for specific examples of syntactic generalization (e.g., nouns that were

[5] The excluded attributes were CFG rules, word/POS tag pairs, and word/argument structure pairs (e.g., "*suffuse* used intransitively").

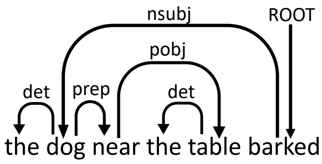
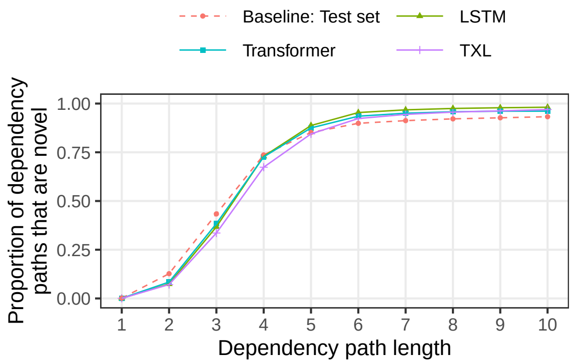
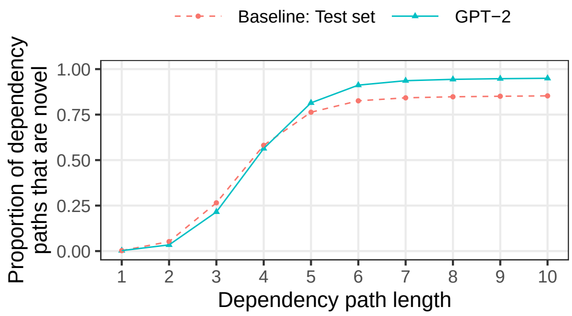


Figure 4: Example dependency tree.



(a) Models trained on Wikitext-103



(b) Models trained on WebText

Figure 5: Novelty of dependency paths in LM-generated text. An example path is [table$_{pobj}$, near], which has length 2 (Figure 4). As baselines, we use text drawn from models' test sets.

used as direct objects in generated text when they have never appeared as direct objects in training).

7 Manual analyses of specific phenomena

Our previous analyses focused on general sequential and syntactic structure. We now investigate some more specific linguistic phenomena by using manual analysis to verify both the quality and the novelty of relevant LM-generated text. Manual analysis is labor-intensive; to use this labor most effectively, we exclusively analyze GPT-2 because it is the strongest-performing model.

For this initial analysis, we study only the novel unigrams that GPT-2 generates; GPT-2 uses subword tokenization, so it can generate novel words by combining seen subwords in novel ways. We study novel words because they give a window into several levels of linguistic structure. Studying the words themselves provides insights into word-internal structure (morphology), while studying the context in which novel words appear provides insights into syntactic and semantic structure, since syntax and meaning use individual words as components. See the online supplement for a detailed taxonomy of the novel words generated by GPT-2. Here in the main paper, we focus only on 4 targeted questions about these novel words. Throughout this section, any word in boldface is novel.

	Morphology		Syntax	
	Baseline	GPT-2	Baseline	GPT-2
Correct	0.99	0.96	0.97	0.94
Incorrect	0.01	0.02	0.00	0.01
Unclear	0.00	0.02	0.03	0.05

Table 2: Syntactic and morphological usage of novel words

7.1 When GPT-2 generates novel words, are they morphologically well-formed?

Finding: The vast majority of GPT-2's novel words (96%) are well-formed (Table 2); this is, however, lower than the baseline (99%).

Specific categories: Forming English plurals requires a choice between two orthographic forms, *-s* and *-es*. In 72 of the 74 novel plurals, GPT-2 made the correct choice (e.g., *Brazilianisms*, *Fowleses*). The two incorrect examples were *1099es* and *SQLes*. Similarly, forming English possessives requires a choice between *-'s* and *-'*. Here, GPT-2 makes the correct choice in 135 out of 136 novel possessives (e.g., *Flexagons'*, *Runerealm's*), with the only error being *watchmakers's*.

Acronyms provide another case for which we can easily quantify well-formedness. Our GPT-2-generated text contains 75 examples of novel acronyms that appear along with the full version of what the acronym stands for. In 72% of cases, the acronym is not a suitable abbreviation (well-formed example in 3, ill-formed example in 4). There are valid reasons why an acronym might not match its expansion; e.g., sometimes English-language publications will translate a non-English phrase but not the acronym derived from it, giving results such as *Doctors Without Borders (MSF)*. However, in our baseline text, 17 of the 21 acronyms that appeared with expansions were suitable, so GPT-2 is still not suitable nearly as often as the baseline (28% vs. 81%).

(3) West of England Cricket and Athletics Club (**WECAC**)

(4) Extremely Large Interactive Neutrino Experiment (**ELIGO**)

Some additional examples of success involve suffixes that require the stem to change spelling, with GPT-2 successfully making the change (5). Some additional mistakes are the use of a plural noun as the first component of a compound (6) and overregularization, namely using the regular suffix *-th* instead of the exceptional suffix *-nd* (7).

(5) a. by " **cookying** " certain searches on the internet
b. **Summission** base camp
c. the **ridiculousities** of war

(6) The...rivers had their **headswaters** in a larger basin

(7) the **752th** year

7.2 When GPT-2 generates novel words, do they fit within their syntactic context?

Finding: The vast majority of GPT-2's novel words (94%) are used in grammatically-correct contexts (Table 2), but it does make more errors than we see in the baseline (e.g., 8, 9).

(8) the manicure that I did for **Sally-themed** a year ago

(9) Slicex **load-samples** provides a single button

Agreement: Despite these errors the vast majority of cases have proper syntax. Some particularly impressive cases involve novel plural words. First, (despite the one mistake in 9), GPT-2 generally does well at providing plural verbs (underlined) to agree with novel plural nouns, whether the verb appears after the noun (10) or before the noun in the context of a question (11). In (12), it correctly uses a plural verb for both verbs that agree with the novel plural subject—a verb within the relative clause, and a verb after it. The correct agreement with the verb after the relative clause is especially impressive because, in both sentences, there are 3 singular "distractors" (italicized) between the subject and the verb. See Haley (2020) for similar observations but with BERT instead of GPT-2.

(10) a. We know that **M-Sinks** need a target
b. **Torpexes** are small hardpoints

(11) Why do **SQLes** have to change

(12) a. The **Huamangas** , who are descendants of indigenous people who lived on the *Isthmus* of *Tehuantepec* before it was covered by *farmland* , have been demanding that the federal government address the issue of climate change .
b. **FOIA-requesters** who think an *agency* has a good *reason* for withholding *information* are not always given a second opportunity to press their case .

Overall, GPT-2 produces the correct verb inflection in 25 of the 26 cases where a novel plural noun is the subject of a present-tense verb (the only English verb type whose form is affected by its subject's number). The only error is in (9).

Other plural-relevant syntax: Beyond agreement, syntactic consequences of plurality are observed in a few other places as well: in using the plural possessive form that is just an apostrophe instead of the singular form of *-'s* (13); in having the pronouns that are coreferential with the noun be plural as well (14); and in following determiners that require a plural noun (15).

(13) The **Fowleses** ' lawyer

(14) a. I love **Klymits** , but it has been nearly impossible for us to find them in stores .
b. The **Sarrats** were lucky to have her as part of their lives

(15) a. these small **townites**
b. so many **Brazilianisms**

Across these three categories, GPT-2 makes no errors, but note that the sample size is small (we found one possessive example, seven examples with coreferential pronouns, and four with number-sensitive determiners).

Incrementing/ordering: Another type of interword relation that GPT-2 appears to have learned is incrementing/ordering, with examples in the online supplement. In one example, GPT-2 increments numbers from *Firstly* to *Fourteenthly*, with *Thirteenthly* and *Fourteenthly* being novel. In another example, it increments the letters at the ends of variable names in computer code, going from *multiplyx* to *multiplyy* to *multiplyz*. In a final example, the prompt ends with an alphabetical list of

	Baseline	GPT-2
p(novel)	0.0022	0.0022
p(novel | in quotes)	0.023	0.028
p(in quotes)	0.0016	0.0015
p(in quotes | novel)	0.016	0.019

Table 3: Quotation mark statistics. Computed over all word-level (not subword-level) unigrams.

	Baseline	GPT-2
Clearly suitable	0.327	0.209
Potentially suitable	0.643	0.587
Probably not suitable	0.002	0.044
Clearly unsuitable	0.001	0.072
Unclear	0.028	0.089

Table 4: How semantically suitable novel words are for their contexts.

companies, and GPT-2 continues this list, staying mostly in alphabetical order and including many novel words along the way.

Quotation marks: A final aspect of sentence structure that we analyze is putting words within quotation marks. In human-generated text, there is an association between novel words and quotation marks: words are much more likely to appear inside quotation marks if they are novel, and they are much more likely to be novel if they appear inside quotation marks. This association is also present in GPT-2's generated text (Table 3), e.g.:

(16) a. The “ **proto-poetry** ” of modern times
b. the “ **un-competition** ” that is happening

These results suggest that GPT-2 might encode some version of the concept “novel word” which it can access when determining whether to include quotation marks.

7.3 When GPT-2 generates novel words, do they result in reasonable meanings?

Finding: GPT-2 does less well in this area than in morphology and syntax, consistent with the claims of Bender and Koller (2020) that language models only learn form, not meaning (Table 4).

Examples: There are some generated examples for which there is clear evidence that the meaning is incorrect (17). One frequent source of mistakes is numbers, revealing a general lack of understanding of the quantities that these numbers represent. Numerical errors include incorrect conversions (18a), physical impossibilities (18b), and inconsistent exchange rates (18c):

(17) a. An old school English term is a **Brazilianism** .
b. ...adding an optional “ **no-knockout** ” version ... so you can actually be knocked out

(18) a. a **1,240-lb** . (**735-kg**) device
b. the ... 4ml tank holds **10.4ml** of e juice .
c. **KES50** (£ 3.50) ... **KES100** (£ 4.00) ... **KES300** (£ 4.50) ... **KES200** (£ 2.50)

Nonetheless, there are also some positive examples where GPT-2 essentially provides a clear and accurate definition of the novel word or otherwise makes use of all aspects of the word:

(19) a. ... the process of **re-nitrification** that gives them a new supply of nitrogen
b. the concept of ‘ **co-causation** ’ , in which effects are thought to be caused by causes that act in parallel
c. the “ **bondbreaking** enchantment ” , which...permanently breaks any binding .

7.4 What generalization mechanisms are used by GPT-2?

We have seen that GPT-2 generates some novel words. What types of generalization does GPT-2 use to create these words? There are two basic types of generalization that might be employed (see Prasada and Pinker (1993), Albright and Hayes (2003), and Dasgupta et al. (2022) for discussion). First, a novel word could be created by a compositional rule that builds up word parts (20a). Alternatively, a novel word could be created via a similarity-based analogy, with similar word parts replacing each other, such as swapping *giraffe* and *elephant* (20b).

(20) a. *elephant* + *-s* = *elephants*
b. *giraffes* - *giraffe* + *elephant* = *elephants*

As these examples show, a given word (e.g., *elephants*) could have been formed in either of these ways, so we can never be certain about which approach GPT-2 is using. However, based on some examples which are reasonably clear, we suspect that GPT-2 employs both types of generalization.

Generalization by composition: In a few cases, GPT-2 generates a novel word whose stem never appears in training but does appear in the context (the prompt plus the previously-generated words): see (21). We believe that these examples are best explained by composition: analogy requires some notion of similarity between the two word parts being swapped, and it is unlikely that the model would have such similarity notions for a word stem it has never seen before. Thus, we think these examples are better understood as the model adding a prefix or suffix to a word from its context, without direct reference to another word that has that prefix or suffix—a form of composition.

(21) a. using the **LHAW** to take out other **LHAWs**

b. Pelagic **epineopterygoid** ... **Sub-epineopterygoid** , N. scapulatus

Generalization by analogy: The online supplement contains one piece of generated text which we believe provides clear evidence for analogy. The prompt for this generation contains the real English word *torero* (borrowed from Spanish), which means "bullfighter." The generation then contains several alternative forms of this word (some with plural inflection): ***tear**ro*, ***torn**ro*, ***tearing**ros*, and ***tears**ros* (e.g., in the sentence *tearingros are taught to avoid the horns*). It appears, then, that GPT-2 has taken the word ***tore**ro* and replaced the first 4 letters (*tore*) with other forms of the verb *tear*: *tear*, *torn*, *tearing*, and *tears*. There is no morphological process in English that adds *-ro* to verbs, so it is unlikely that these words were generated via composition; instead, it seems more likely that they were generated via analogy.

8 Discussion

Using our analysis suite RAVEN, we have found that models generated many types of novelty—novel n-grams of all sizes, novel syntactic structures, and novel morphological combinations. However, they also show many signs of copying: for local structure, they are substantially less novel than the baseline; and we see occasional large-scale copying, such as duplicating passages from the training set that are over 1,000 words long.

Compositionality: Compositional generalization (combining familiar parts in novel ways) is crucial for processing both the syntax and semantics of natural language (Montague, 1970). It is often discussed in the context of out-of-distribution generalization (Hadley, 1994; Hupkes et al., 2020; Keysers et al., 2020; Li et al., 2021), typically relying on synthetic datasets to test models' compositional abilities (Lake and Baroni, 2018; Kim and Linzen, 2020; McCoy et al., 2020). The baselines in Table 1 show that compositional syntactic generalization is important even for in-distribution test sets drawn from large-scale natural corpora. Most notably, the majority of test sentences had a sentence-level syntactic structure that had never appeared in the training set.

Turning to the model results in Table 1, all models displayed nonzero rates of compositional generalization, giving an existence proof that they can perform these types of generalization. Nonetheless, the models' scores are lower than the baseline, so their generalization might be limited to particular subcases, instead of being as general as human generalization. In the opposite direction, however, we also found examples where GPT-2 generalized too freely, such as generating the word *752th* (Section 7.1). We conclude that it may not be enough to simply encourage models to be systematic, because language is not completely systematic. Instead, we need models that can both figure out linguistic rules and recognize exceptions to those rules (O'Donnell, 2015; Yang, 2016).

Our analyses mainly focused on compositional generalization as it applies to linguistic form (specifically, morphology and syntax). An important direction for future work would be to analyze novelty in meaning.

Evaluating novelty: The main point of our work is that novelty has not received the attention it deserves in evaluation of LMs. For generated text to truly illustrate a model's generative capabilities, that text must be novel—otherwise, it may only illustrate the model's ability to copy but not other abilities (e.g., the ability to be coherent). We recommend using the level of novelty found in an in-distribution test set as a baseline: if the model is at least as novel as this baseline, we can rule out the possibility that it is copying excessively.

Recent increases in data quantity make it especially critical to check for novelty because the magnitude of recent datasets can break our intuitions about what can be expected to occur naturally. For instance, some notable work in language acquisition (e.g. Kuczaj II, 1977; Marcus et al., 1992) relies on the assumption that regular past tense forms of irregular verbs (e.g., *becomed*, *teached*) do not appear in a learner's experience, so if a learner produces such words, they must be novel to the learner. However, for all 92 basic irregular verbs in English, the incorrect regular form appears in GPT-2's training set; details are in the online supplement, along with results for another category often assumed to be novel in human experiments, namely nonsense words such as *wug* (Berko, 1958). Thus, when we are using language models trained on such large-scale datasets, it is not safe to assume that something is absent from the training set; we must explicitly check.

Improving novelty: One straightforward approach for increasing novelty would be to modify the sampling procedure to suppress highly-copied outputs, similar to penalties used to prevent repetition (Keskar et al., 2019). Another approach would be to implement more nuanced forms of deduplication during training: We found that supercopying mainly arises when there is repetition in the training set, so eliminating such repetition might improve models' novelty. Indeed, concurrent work (Lee et al., 2022b; Kandpal et al., 2022) has shown that deduplication can substantially decrease the extent to which large n-grams are copied from the training set.

Ideally, however, we would find ways to decrease copying that are deeper, without requiring post-hoc modifications to the training data and sampling procedure. In humans, novelty has long been attributed to the usage of symbolic, compositional rules. Thus, greater novelty might be achieved through models that build in compositional mechanisms, such as RNNGs (Dyer et al., 2016) and TP-Transformers (Schlag et al., 2019).

Alternatively, one major difference between text generation in humans and neural LMs is that humans usually have a meaning that they want to express that guides their text generation, whereas LMs have no explicit plan when producing text. This difference may partly explain the ways in which models are less novel than humans: since models mainly manipulate text alone, they fall back to repeating text they have seen before. Thus, novelty may be improved by incorporating more explicit semantic planning (Rashkin et al., 2020).

9 Conclusion

In machine learning, it is critical to evaluate models on a withheld test set. When text is sampled from a language model, that text might be copied from the training set, in which case it is not withheld—so using that text to evaluate the model (e.g., for coherence or grammaticality) is not valid. Thus, it is important to consider novelty when using text generation to evaluate the model's abstract abilities. We have introduced RAVEN, an analysis suite covering sequential and syntactic structure, and have applied it to several models, showing that models are rarely novel for local structure but are often novel for larger-scale structure. The types of novelty that models display provide evidence that they have captured a range of linguistic abstractions, such as constituency structure, dependency structure, and several morphological processes. However, models occasionally copy even very long passages, showing that generated text cannot be assumed to be novel: we must directly check for novelty, such as by using the analyses in RAVEN. Overall, our results demonstrate the importance of considering a model's training data when evaluating that model's abilities.

Acknowledgments

We thank OpenAI for providing access to the WebText dataset. For helpful comments and discussion, we are grateful to Suhas Arehalli, Saadia Gabriel, Coleman Haley, Yichen Jiang, Nebojsa Jojic, Najoung Kim, Géraldine Legendre, Grusha Prasad, Eric Rosen, Sebastian Schuster, Paul Soulos, Shiyue Zhang, the Deep Learning Group at Microsoft Research Redmond, the Johns Hopkins Neurosymbolic Computation Lab, the NYU Computation and Psycholinguistics Lab, the NYU Machine Learning for Language Group, and the TACL reviewers and Action Editor. Any errors are our own. For technical assistance with Hugging Face, we thank Teven Le Scao and Patrick von Platen. We are also grateful to the Maryland Advanced Research Computing Center (MARCC) for providing the computing resources used in our experiments. The raven image used in our title comes from Pixabay user Nika_Akin.[6]

Portions of this research were supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1746891. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[6]https://pixabay.com/illustrations/crow-raven-black-dark-bird-ink-4779560/

References

Ekin Akyurek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Towards tracing knowledge in language models back to the training data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2429–2446, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

Laurie Bauer and Rodney D. Huddleston. 2001. Lexical word formation. In *The Cambridge Grammar of the English Language*. Cambridge University Press.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14(2-3):150–177.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, pages 267–284.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium*, pages 2633–2650.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988. Association for Computational Linguistics.

Ishita Dasgupta, Erin Grant, and Tom Griffiths. 2022. Distinguishing rule and exemplar-based generalization in learning systems. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4816–4830. PMLR.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? Scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. 2022. Measuring causal effects of data statistics on language model's 'factual' predictions. *arXiv preprint arXiv:2207.14251*.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017a. Efficient softmax approximation for GPUs. In *International Conference on Machine Learning*, pages 1302–1310.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017b. Improving neural language models with a continuous cache. In *International Conference on Learning Representations*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Robert F. Hadley. 1994. Systematicity in connectionist language learning. *Mind & Language*, 9(3):247–272.

Coleman Haley. 2020. This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341. Association for Computational Linguistics.

Xiaochuang Han and Yulia Tsvetkov. 2022. ORCA: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. *arXiv preprint arXiv:2205.12600*.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701. Association for Computational Linguistics.

Robert Hawkins, Takateru Yamakoshi, Thomas Griffiths, and Adele Goldberg. 2020. Investigating representations of verb bias in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Rodney Huddleston and Geoffrey Pullum. 2001. *The Cambridge Grammar of the English Language*. Cambridge University Press.

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

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR.

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

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional Transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Stan A. Kuczaj II. 1977. The acquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behavior*, 16(5):589–600.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882.

Benjamin LeBrun, Alessandro Sordoni, and Timothy J. O'Donnell. 2022. Evaluating distributional distortion in neural language modeling. In *International Conference on Learning Representations*.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2022a. Do language models plagiarize? *arXiv preprint arXiv:2203.07618*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022b. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! Making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728. Association for Computational Linguistics.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780. Association for Computational Linguistics.

Gary Marcus. 2020. The next decade in AI: Four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.

Gary Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the society for research in child development*, pages i–178.

R Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5328–5339. Association for Computational Linguistics.

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

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. 2007. GLEU: Automatic evaluation of sentence-level fluency. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 344–351. Association for Computational Linguistics.

Timothy J O'Donnell. 2015. *Productivity and reuse in language: A theory of linguistic computation and storage*. MIT Press.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Frank Palmer, Rodney D Huddleston, and Geoffrey K Pullum. 2001. Inflectional morphology and related matters. In *The Cambridge Grammar of the English Language*. Cambridge University Press.

Ludovica Pannitto and Aurélie Herbelot. 2020. Recurrent babbling: evaluating the acquisition of grammar from limited input data. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 165–176. Association for Computational Linguistics.

Julian M. Pine and Elena V. M. Lieven. 1997. Slot and frame patterns and the development of the determiner category. *Applied psycholinguistics*, 18(2):123–138.

Yuval Pinter, Cassandra L. Jacobs, and Jacob Eisenstein. 2020. Will it unblend? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1525–1535. Association for Computational Linguistics.

Sandeep Prasada and Steven Pinker. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295. Association for Computational Linguistics.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3532–3542. Association for Computational Linguistics.

Imanol Schlag, Paul Smolensky, Roland Fernandez, Nebojsa Jojic, Jürgen Schmidhuber, and Jianfeng Gao. 2019. Enhancing the Transformer with explicit relational encoding for math problem solving. In *NeurIPS Workshop on Context and Composition in Biological and Artificial Neural Systems*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

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

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741. Association for Computational Linguistics.

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

Takateru Yamakoshi, Thomas Griffiths, and Robert Hawkins. 2022. Probing BERT's priors with serial reproduction chains. In *Findings of* *the Association for Computational Linguistics: ACL 2022*, pages 3977–3992, Dublin, Ireland. Association for Computational Linguistics.

Charles Yang. 2013. Ontogeny and phylogeny of language. *Proceedings of the National Academy of Sciences*, 110(16):6324–6327.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in Neural Information Processing Systems*, 32.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021a. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. OPT: Open pre-trained Transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021b. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020b. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Albert Ziegler. 2021. A first look at rote learning in GitHub Copilot suggestions. *GitHub Docs*.

A Model details

In all cases, we generated text using a beam size of 1. Except where otherwise mentioned, the decoding scheme was top-40 sampling. The GPT-2 models were trained on the WebText corpus, which contains about 7.7 billion words; the other models were all trained on Wikitext-103, which contains about 103 million words.

LSTM: The LSTM architecure was introduced by Hochreiter and Schmidhuber (1997). We used the implementation from Gulordava et al. (2018) to train a 2-layer LSTM language model with a hidden state size of 1024 and dropout of 0.2.[7] This model had a total of 291 million trainable parameters. We trained this model using stochastic gradient descent with an initial learning rate of 20. The learning rate was divided by 4 whenever the validation loss did not improve from the previous epoch, and we halted training when validation loss failed to improve for two consecutive epochs. This model achieved a perplexity of 45.7 on the test set. The best-performing LSTM trained on Wikitext-103 that we are aware of from past literature is in Grave et al. (2017b), and its test-set perplexity was 48.7, so we conclude that our newly-trained model is a strong LSTM. The fact that our model outperforms the previous one is likely due to its number of layers: we used 2 layers, while Grave et al. do not mention the number of layers used, which we believe most likely means that they used 1 layer.

Transformer: As a basic Transformer language model, we use the trained model from Khandelwal et al. (2020) (note that we only use the base Transformer, without the datastore that Khandelwal et al. introduce). This model uses the Transformer decoder architecture from Vaswani et al.

[7] We used the hidden size of 1024 following Grave et al. (2017b). For the other hyperparameters, we searched over two numbers of layers (1 layer or 2 layers) and four dropouts (0.1, 0.2, 0.5, and 0.65) and chose the combination that gave the lowest perplexity on the validation set.

(2017) trained in fairseq (Ott et al., 2019) using tied weights (Press and Wolf, 2017), adaptive inputs (Baevski and Auli, 2019), a hidden size of 1024, and an adaptive softmax (Grave et al., 2017a). It has 247 million trainable parameters, and its perplexity on the test set is 18.65.

Transformer-XL: We use the HuggingFace (Wolf et al., 2020) version of Transformer-XL (Dai et al., 2019). Transformer-XL is a Transformer modified to include a recurrence mechanism and a different positional encoding scheme, which are intended to help it process long-distance dependencies. It has a hidden size of 1024, 257 million trainable parameters, and its perplexity on the test set is 18.30.

GPT-2: We use the HuggingFace (Wolf et al., 2020) version of GPT-2 (Radford et al., 2019), which is a decoder-only Transformer architecture (Vaswani et al., 2017) trained on the WebText corpus, which is constructed from webpages linked to on Reddit which have received at least 3 karma. There are four sizes of GPT-2 available from HuggingFace (`gpt2`, with a hidden size of 768 and 117 million trainable parameters; `gpt2-medium`, with a hidden size of 1024 and 345 million trainable parameters; `gpt2-large`, with a hidden size of 1280 and 774 million trainable parameters; and `gpt2-xl`, with a hidden size of 1600 and 1.558 billion trainable parmaters). The largest of these, `gpt2-xl`, is the one that Radford et al. (2019) refer to as *GPT-2* and is the one we use for our experiments except where otherwise indicated.

B Tokenization and other text preprocessing

Prompts: For the Wikitext prompts, we used prompts of length 0, 16, 128, and 512; most of our experiments used only the length-512 prompts. For the WebText prompts, the prompt lengths were 0, 18, 141, and 564 subword tokens; however, we extended the prompt past that length as was necessary to ensure that the prompt ended with a complete word. If this required adding more than 10 additional tokens, we discarded the prompt and sampled a new one. The WebText prompt lengths were chosen to be approximately 1.1 times the Wikitext prompt lengths because there are approximately 1.1 WebText subword tokens for every word.

As our baseline text, we used the words that followed the prompt in the test set. Due to the small size of the Wikitext-103 test set (it contains approximately 245,000 tokens, while the continuations following the prompts total 1,000,000 tokens), some of the Wikitext-103 continuations that were used to make the baseline text necessarily have parts that overlap with parts of other continuations, but no two continuations are identical. For the Webtext baseline, there was no such overlap because the dataset was large enough to avoid it.

N-gram novelty: For computing n-gram novelty, we did not perform any processing of Wikitext text or the text generated by Wikitext models; thus, this text uses the tokenization from the Wikitext-103 dataset, which is a slightly modified version of the Moses tokenizer (Koehn et al., 2007). For WebText text and text generated by GPT-2, we converted GPT-2's subword IDs into text using the GPT2Tokenizer from the HuggingFace Transformers library (version 2.11.0). We then replaced each newline with the token *&NEWLINE;* (which never occurs in WebText), to be consistent with Wikitext, in which each newline is a token. Wherever there were multiple spaces in a row, we replaced them with a single space. We then tokenized this text using the Moses tokenizer (Koehn et al., 2007) and used the resulting tokens to compute n-gram novelty.

Syntactic novelty: Many of our syntactic analyses operate at the level of sentences. Thus, we first sentence-tokenized our text using the NLTK sentence tokenizer[8] and then parsed them using a constituency parser (Kitaev and Klein, 2018) and dependency parser (Zhang et al., 2020b). These parsers perform their own tokenization, so we wanted to provide them with untokenized text. For WebText baselines and GPT-2 text, this was straightforwardly accomplished by not performing word-level tokenization before passing text to the sentence tokenizer and the parser. For Wikitext baselines and text generated by Wikitext-based models, we first detokenized the text using the Moses detokenizer (Koehn et al., 2007) and then passed the detokenized text to the parsers.

Analyses: For identifying novel unigrams for our analyses in Section 7, we used the GPT-2 generated text as it was tokenized for the n-gram novelty evaluations, and we then used the n-gram

[8] `nltk.org`

novelty annotations to determine which unigrams were novel.

C Novel bigrams

Figure 6 shows examples of novel bigrams generated by each of our models.

D Supercopying examples and statistics

Figure 7 gives examples of generated text, as well as text from our models' test sets, that we classify as supercopying: duplicating a passage that is 100 words long or longer from the training set. Figure 8 gives statistics about how many times supercopied 100-grams appeared in the relevant model's training set.

E Untruncated pointwise duplication scores

Figure 9 shows the same results as Figure 2 but with both the truncated and untruncated pointwise duplication scores (whereas Figure 2 only shows the truncated scores).

F Plots showing the effects of the decoding scheme

Figure 10 shows how the pointwise duplication score is affected by varying three decoding parameters.

G Evaluating perplexity

G.1 Evaluating overlap between Wikitext-103 and WebText

To measure the perplexity of generated text, we used GPT-2 (which was trained on WebText) to measure perplexity for models trained on Wikitext-103, and we used TXL (which was trained on Wikitext-103) to measure perplexity for models trained on WebText. We justified this choice based on the fact that Wikitext-103 was constructed entirely from Wikipedia articles, while Wikipedia articles were excluded from WebText, meaning that there should be no overlap between the training sets of Wikitext-trained models and WebText-trained models. However, there is a caveat for making this assumption: the WebText creation process excluded Wikipedia articles, but the *text from* Wikipedia articles could still potentially occur in WebText, because there are many non-Wikipedia websites that copy data from Wikipedia; and because Wikipedia writers could potentially generate Wikipedia content by copying it from other public-domain websites.

Here we test our no-overlap assumption more rigorously. To do so, we randomly selected one thousand 20-grams from the WebText training set and checked whether they appeared in the Wikitext-103 training set. Similarly, we also selected one thousand 20-grams from the Wikitext-103 training set and checked whether they appeared in the WebText training set tokenized with the Moses tokenizer (Koehn et al., 2007), which was the basis of the tokenization in Wikitext-103. To control for tokenization differences that might persist despite the use of the Moses tokenizer, we lowercased all text and deleted any words containing any characters besides the 26 Roman letters.

We found that 0 of the 1000 WebText 20-grams appeared in the Wikitext-103 training set, so it seems very safe to use TXL to evaluate the perplexity of text generated by models trained on WebText. In the other direction, 12 of the 1000 Wikitext 20-grams appeared in the WebText training set. This shows that a small amount of text that appears in Wikipedia text did end up in WebText. Nonetheless, the proportion of overlap is very small (0.012), so we conclude that it is still fair to use GPT-2 to evaluate the perplexity of text generated by models trained on Wikitext-103.

G.2 Details of perplexity evaluation

To evaluate the perplexity of a piece of text using Transformer-XL or GPT-2, we adapted code from Hugging Face at `https://huggingface.co/transformers/perplexity.html`. For each model, we used a stride of 512 tokens and a maximum length of 1024 tokens. That is, the perplexity was evaluated in segments of 1024 tokens each, with each segment preceded by a further context of 512 words whose perplexity was not evaluated as part of the segment being evaluated. This approach ensures that every token has at least 512 tokens of prior context available; tokens at the end of a 1024-token segment have an even longer context (specifically, a context of 1535 tokens for the last token in each segment).

H How does prompt length affect novelty?

Prompt length could reasonably be expected to increase or decrease novelty. On one hand, shorter prompts might not give the model much context

Generation method	Examples of novel bigrams in context
LSTM	... surrounded by a **tall brick** , square with a flat facade ...
	... including the placement of **some prosimians** , with both ...
	... the first legislative assembly of the newly **created territory** ...
Transformer	... it is revealed to be the **poet Gidding** , now living ...
	Archaeological findings from La **Venta indicate** that ...
	... proposed by **Edward Armour** in his wooden Ironclads .
Transformer-XL	... under the 1906 – **07 shipbuilding** program .
	... the same **raw warmth** that was present in other ...
	It is assumed **that Suetonius** used information gathered ...
GPT-2	... **Scope Names** with Closures ...
	... bidding $ 6 against the **AppNexus SSP** and just ...
	... a space (e.g. , a **ceiling overhang**) ...

Figure 6: Randomly-selected examples of novel bigrams generated by models. Most of the novel bigrams are used well in context.

to build from, which could lead the model to fall back on what it has seen during training, making it less novel. On the other hand, we observe from the baselines in Figure 1 that there is a reasonably high overlap between models' training sets and their test sets. Since our prompts are drawn from the test set, a long prompt might contain long portions that also appear from the training set, which could encourage the model to further copy from that part of the training set, in which case longer prompts would lead to lower novelty than shorter prompts.

To assess how novelty is affected by prompt length, we consider only duplication from the training set, not from the context, because a longer prompt trivially provides more opportunities for copying from the context. In general, the length of the prompt does not appear to affect novelty much (Figure 11). For the LSTM and the baseline of text drawn from the Wikitext-103 test set, we do not discern any effect of prompt length. For the Transformer and GPT-2, longer prompts lead to slightly more novelty than shorter prompts, while Transformer-XL shows the opposite effect, with shorter prompts leading to slightly more novelty than longer prompts. The WebText baseline shows some differences between prompt lengths, but we do not see a clear generalization there as novelty is not affected consistently by length: for example, length 0 is more novel than length 18 but less novel than length 141.

I Position in generated text

Figure 12 illustrates how novelty is related to position in the output text. For these analyses, we only consider duplication from the training set, not from the context, because positions later in the generation have more context to copy from. We use pointwise duplication scores truncated at 10 to control for the fact that later positions can have higher untruncated pointwise scores than earlier positions can have; by using truncated scores, all positions that we consider have the same possible range of values (1 to 10 inclusive). We group generated text into bins of 100 words (positions 0 to 99, positions 100 to 199, etc.) and then compute the mean truncated pointwise duplication score for each bin, discarding the first bin because its first 10 positions have a different range of possible scores than the rest of the positions in the generation.

There is little effect of position in the baselines, the LSTM, and the Transformer, but in GPT-2 and Transformer-XL, there is greater duplication at later positions in the generated text. Though the effect is consistent for these two models, the effect size is small, with the pointwise duplication score increasing by only about 0.2 from the start of the generation to the end of the generation.

J Duplication from the training set vs. the context

In the main text, we only report results that collapse together duplication from the training set and

Generation method	Example of supercopying
Wikitext test set	<eos> <eos> = = Themes = = <eos> <eos> The Hustler is fundamentally a story of what it means to be a human being , couched within the context of winning and losing . Describing the film , Robert Rossen said : " My protagonist , Fast Eddie , wants to become a great pool player , but the film is really about the obstacles he encounters in attempting to fulfill himself as a human being . He attains self @-@ awareness only after a terrible personal tragedy which he has caused — and then he wins his pool game . " Roger Ebert concurs with this assessment
LSTM, Transformer, Transformer-XL	. <eos> <eos> = = Background = = <eos> <eos> The Boat Race is a side @-@ by @-@ side rowing competition between the University of Oxford (sometimes referred to as the " Dark Blues ") and the University of Cambridge (sometimes referred to as the " Light Blues ") . The race was first held in 1829 , and since 1845 has taken place on the 4 @.@ 2 @-@ mile (6 @.@ 8 km) Championship Course on the River Thames in southwest London . The rivalry is a major point of honour
WebText test set	All crafted armor is Adaptive . NEWLINE NEWLINE Nearly all schematics require a new Premium (green) crafting material called a Component . This new Component is made by combining two existing crafting materials along with two vendor purchased materials , all of which are of the same Grade as the component . A unique Component exists for each skill and each Grade . Example : An Armormech crafter wants to craft a Grade 6 Chest Piece (Level 46) . This is a Prototype (blue) quality armor and requires 4 Ciridium and 4 Durasteel Armor Assembly Components . The player will need 2 Durasteel , 2 Zal Alloy , and 2 Thermoplast Flux to craft each component . Once they have the 4 Components and 4 Ciridum from Underworld Trading , they will be able to craft their chest piece . If a player chose to , they could level up exclusively by crafting Components ! NEWLINE NEWLINE In most cases , Crafting times have been greatly reduced . This will enable crafters to craft more items to coincide with the faster leveling .
GPT-2	your bank account . NEWLINE NEWLINE Your billing zip code needs to be 5 digits . NEWLINE NEWLINE Please double check your CEP info . The CEP format should be something like 12345-678 . NEWLINE NEWLINE Please double check your tax identifier . NEWLINE NEWLINE There was a problem saving your address . NEWLINE NEWLINE There was a problem saving your card info . NEWLINE NEWLINE There was a problem saving your personal information . NEWLINE NEWLINE McAfee Secure sites help keep you safe from identity theft , card fraud , spyware , spam , viruses and online scams . NEWLINE NEWLINE Copying Prohibited by Law - McAfee Secure is a Trademark of McAfee , Inc . NEWLINE NEWLINE Unknown card type . NEWLINE NEWLINE No card number provided . NEWLINE NEWLINE card number is in invalid format . NEWLINE NEWLINE Wrong card type or card number is invalid . NEWLINE NEWLINE card number has an inappropriate number of digits . NEWLINE NEWLINE Please enter numbers here . NEWLINE NEWLINE Please enter an integer value . NEWLINE NEWLINE Numbers must be less or equal to $ $ $ $ NEWLINE NEWLINE All the required fields have not been filled out . Click OK to proceed without all the required information , or click Cancel to finish entering the missing data .

Figure 7: Examples of supercopying. The passage in the second row appeared in the generated text for all 3 of our Wikitext models (LSTM, Transformer, and Transformer-XL).

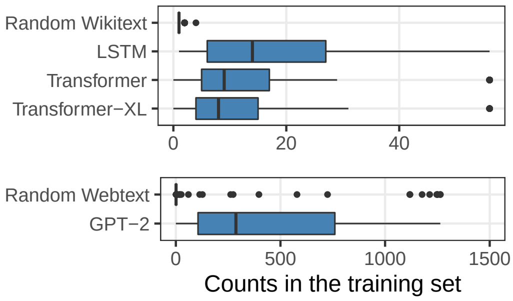


Figure 8: Counts of how often 100-grams supercopied by each model appear in that model's training set, compared to counts of random 100-grams from the training sets. For legibility, some GPT-2 outliers have been removed. The biggest outlier was a supercopied passage that occurred 176,424 times in GPT-2's training set.

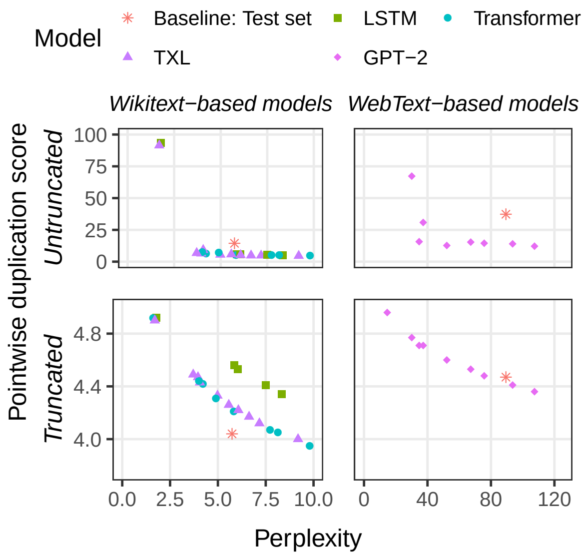


Figure 9: Manipulations to the decoding scheme that result in higher-quality text (i.e., lower perplexity; x-axis) also result in decreased novelty (i.e., a greater degree of duplication; y-axis). Each point shows a different decoding scheme.

duplication from the context (the prompt and the previously-generated text). In Figure 13, we separate these two sources of duplication. The plots showing duplication from the training set alone are almost identical to the plots showing both the training set and the context, showing that models almost never copied content from the context that was not also in the training set. Models showed much less duplication from the context than from the training set, which is unsurprising because the training sets are much larger than the contexts, meaning that there are far more pieces of text that would count as duplicated from the training set than duplicated from the context.

K Vetting syntax

We considered evaluating the novelty of 7 aspects of syntax. For each of these 7 aspects, we conducted manual analyses to determine whether the numerical results gained from our parses were roughly accurate. Below we describe these manual analyses.

POS tags: We considered a novel part-of-speech (POS) tag to be an instance where a word was generated with a POS tag that it had never appeared with in training (but where the word had appeared in training with a different POS tag). From an initial inspection of the generated words that the parser identified as having a novel part of speech, we found that most of them were correctly labeled, but that the training set actually did contain an instance of the word with that part of speech (just with that training instance mislabeled). Thus, we concluded that we could not trust the quantitative results for this factor, because most generated words identified as having a novel POS tag had actually appeared as that POS in training.

CFG rules: The CFG rules are the context-free rules present in the parses from the constituency parser (ignoring rules that include a terminal symbol—i.e., only considering ones composed entirely of nonterminals). From an initial inspection of the CFG rules identified as novel, every example that we looked at was the result of a parser error (where the parser assigned an overly flat rule, e.g. *NP* $\rightarrow$ *-LRB- CD -RRB- , NP CD CD CD , NP , NP*). Thus, we concluded that we would be unable to trust numerical results covering CFG rules.

POS sequence: The POS sequence of a sentence is the sequence of POS tags assigned to its words by the constituency parser. For these, we looked at 100 generated sentences identified as having a novel POS sequence, and for each manually checked whether the POS sequence assigned by the parser was accurate. These generally were accurate (Figure 14), and the POS sequences generally had high scores for novelty, so we conclude that the quantitative results are approximately the correct order of magnitude for the POS sequences.

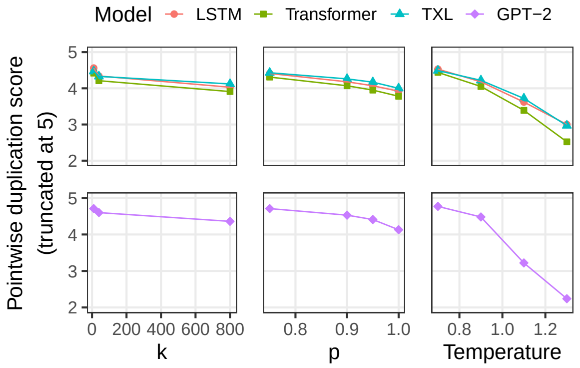


Figure 10: How novelty is affected by 3 decoding parameters: k in top-k sampling, p in top-p sampling, and the temperature. A higher value on the y axis means that the generated text is less novel. As each parameter is increased, novelty also increases (that is, duplication—on the y-axis—decreases).

Parse structure: We define a sentence's parse structure as its constituency parse minus the leaves (i.e., the terminal nodes in the tree). If a sentence has a novel POS sequence, it is guaranteed to also have a novel parse structure. Therefore, we did not conduct an additional inspection of the parse structures on top of the POS sequences, because the parse numbers are close to the POS number.

Dependency arcs: For dependency arcs, we first checked 100 dependency arcs identified as novel for each model to confirm that they were not parser errors. We then looked at the subset of each of these 100-arc sets which were correctly labeled and for which there were no more than 100 training sentences that could possibly contain that arc (i.e., training sentences containing both relevant words). We then checked those training sentences to confirm that the dependency arc in question did not appear in that sentence. These results were generally strong (Figure 15), so we conclude that the numerical results for dependency arcs are approximately correct.

Dependency roles: Similarly to the dependency arcs, we checked 100 dependency roles per model that were identified as novel to check if the roles were correctly identified. We then looked at the subset of those 100-role sets which were correctly labeled and for which there were no more than 100 training sentences that could possibly contain that role (i.e., training sentences containing the relevant word). We then checked those training sentences to see if the dependency role truly was novel. These results were also successful enough for us to conclude that the numerical results were approximately correct and could be reported in the paper (Figure 16).

Dependency argument structure: Dependency argument structure is the list of argument types that a verb has (e.g., *subject, direct object, indirect object*). We did not manually analyze this due to how labor-intensive it would be (requiring analysis of a large number of entire training sentences). Thus, we do not provide numerical results for this, since we do not have an estimate of how reliable such numbers would be.

L Example of a mismatch between local and global novelty

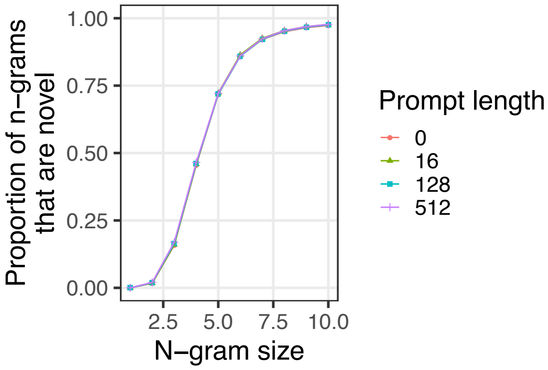
In both our n-gram analyses and syntactic analyses, we found that LMs were less novel than the baseline for local structure (e.g., small n-grams or individual dependency arcs) but were more novel than the baseline for larger-scale structure (e.g., large n-grams or overall sentence structure). As an example of how such a local/global mismatch is possible, suppose that the training set contained only (22):

(22) alligators resemble crocodiles, and dolphins resemble sharks

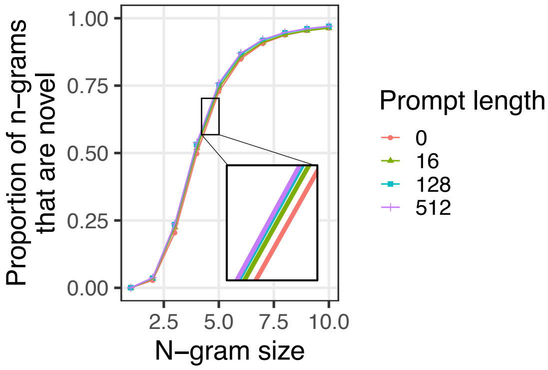
Generated sentences (23a) and (23b) are both novel trigrams, but (23b) also contains novel bigrams whereas (23a) does not. Thus, these two generations have the same trigram novelty despite differing in their smaller-scale, bigram novelty.

(23) a. alligators resemble sharks
b. crocodiles resemble dolphins

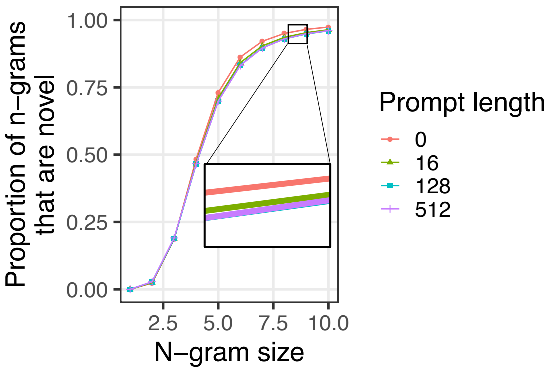
In terms of syntactic structure, both generated sentences have a novel overall parse structure (since the training set contains only a sentence with two clauses in it, while both generated sentences are only a single clause). Despite having the same novelty for global syntactic structure, these sentences differ in their proportion of individual dependency arcs that are novel: In sentence (23b), both noun-verb dependency arcs are novel (*crocodiles* as the subject of *resemble* and *dolphins* as the direct object of *resembles*), while in sentence (23b) both of the noun-verb dependency arcs appear in the training set. Thus, as with n-gram novelty, (23b) and (23a) have different levels of small-scale novelty despite having the same level of global novelty.



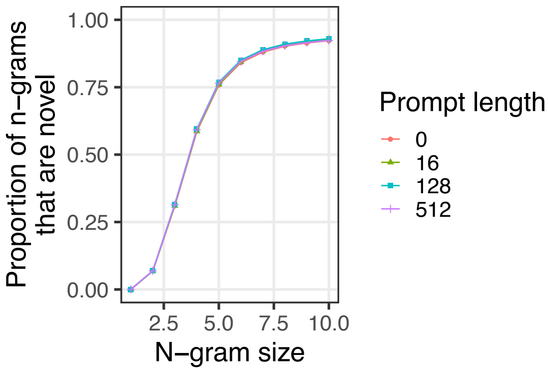
(a) LSTM



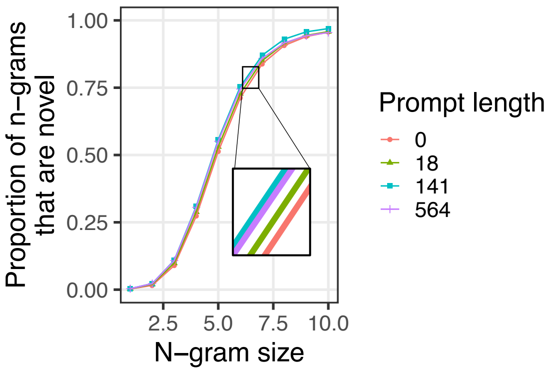
(b) Transformer



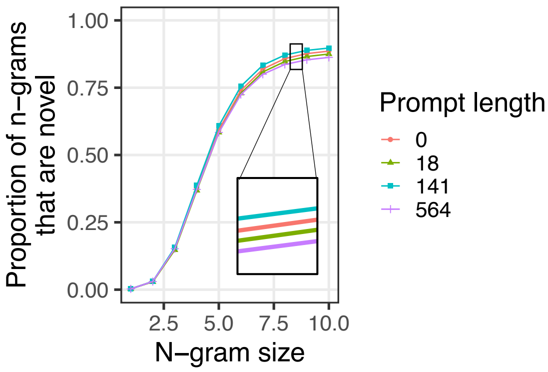
(c) TXL



(d) Wikitext baseline text

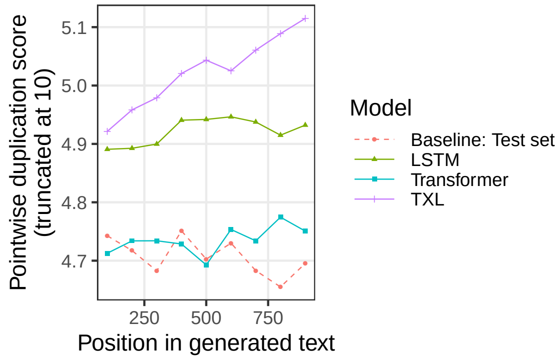


(e) GPT-2

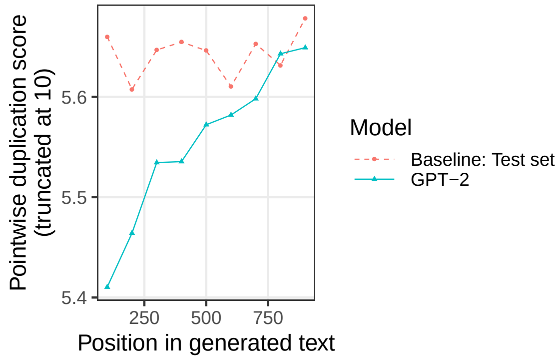


(f) Webtext baseline text

Figure 11: Effect of prompt length.



(a) Wikitext-based models.



(b) WebText-based models.

Figure 12: Effect of output position on novelty.

M Examples of syntactic novelty

Although we cannot report reliable numerical results for several of the aspects of syntax that we considered in Appendix K (due to the infeasibility of manually checking all examples and the high error rates of automatic methods), it may still be worthwhile to see if there are any individual examples that we can identify to provide an existence proof that models do, at least sometimes, perform the types of syntactic generalization in question.

To that end, we identified several types of syntactic generalization that have received attention in prior literature in language acquisition and/or analysis of LMs, and manually looked for examples that we could verify were novel. For example, given a dependency role (e.g., "*watch* as the head of an nsubj relation"), we identified all training sentences containing the relevant word (here, *watch*) and then manually checked whether that word ever has the relevant dependency role, in case the parser missed it. This method ensures perfect precision (all examples that we identify are certain to be true instances of syntactic novelty), though there are no guarantees about recall (we may miss some, or even many, examples). To decrease the amount of manual inspection required, we only considered candidate examples that required inspecting 500 training sentences or fewer. For each example from a Wikitext-based models, we have also provided a relevant training example, to highlight the type of generalization that is being illustrated (but note that, in all cases, there are many more relevant training examples that we do not include in the paper). We have not included training examples from WebText because it is not a publicly-released dataset.

Novel part-of-speech tags: Here we looked for generated words which are used as a noun in the generation and which appear in the training set, but never as a noun (for this purpose, we consider all noun POS tags to be identical), and similarly for verbs. We find a few examples; note that some of these might be instances of overgeneralization (that is, English speakers might judge some of them to be ungrammatical):

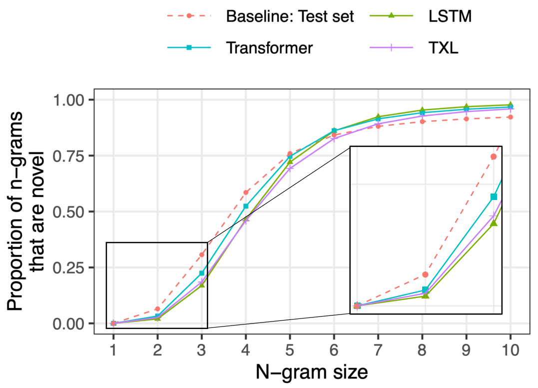
(24) Novel as a noun

a. (i) *Transformer:* They declared two days of **secede** from Yugoslavia.
   (ii) *Training example*: Led by a former Jedi named Count Dooku , thousands of planetary systems threaten to **secede** from the Galactic Republic .

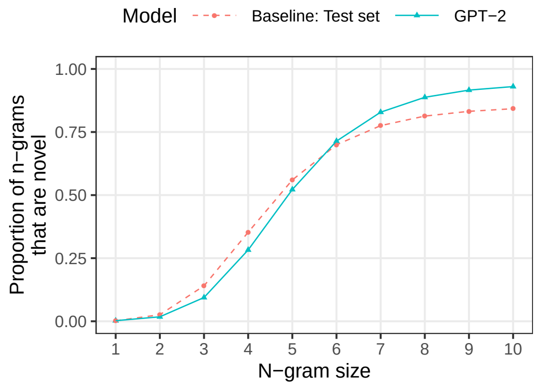
b. (i) *Transformer:* While Vespasian did not have any public office beyond **tribunician** as emperor, he was probably appointed as a general of the Praetorians in 63 AD.
   (ii) *Training example:* As the eldest and most experienced of Vespasian 's sons , Titus shared **tribunician** power with his father , received seven consulships , the censorship , and was given command of the Praetorian Guard

c. (i) *Transformer:* Apollodorus of Damascus interpreted the **colossal** as a gigantic head with the crown of Horus on the left hand side.
   (ii) *Training example:* This **colossal** project involved around 600 people to restore the monument and cost a total of US $ 6,901,243 .

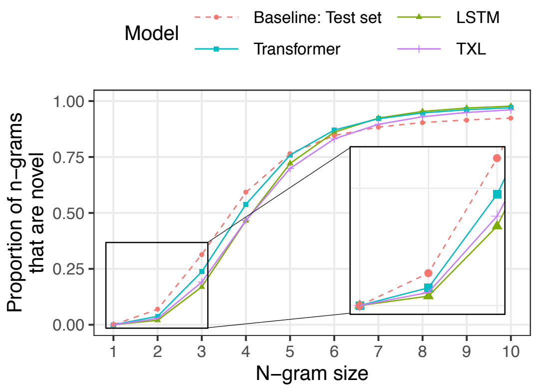
d. (i) *Transformer:* The **exploratory**, called California California Califor-



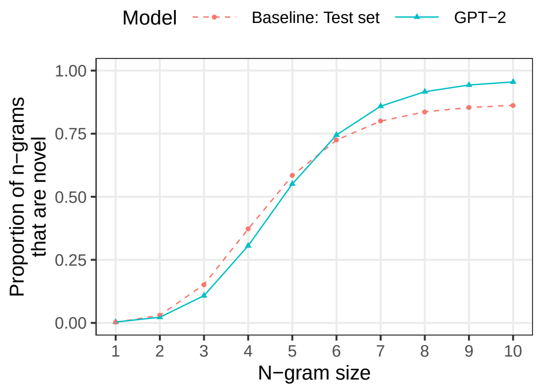
(a) Wikitext-based models: Duplication from the training set and/or context.



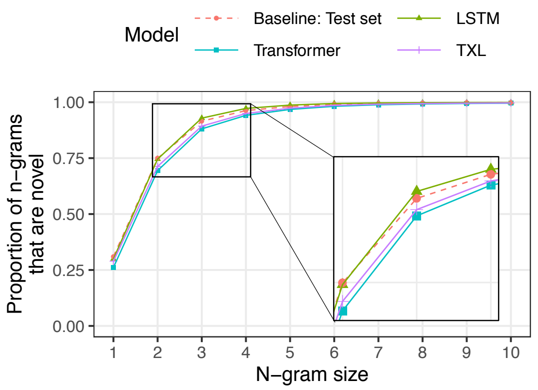
(b) WebText-based models: Duplication from the training set and/or context.



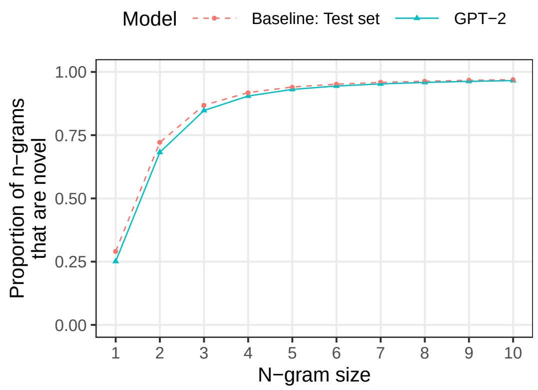
(c) Wikitext-based models: Duplication from the training set.



(d) WebText-based models: Duplication from the training set.



(e) Wikitext-based models: Duplication from the context.



(f) WebText-based models: Duplication from the context.

Figure 13: Duplication from the training set, context, or both.

	Correctly tagged
Wikitext baseline	0.83
LSTM	0.81
Transformer	0.84
TXL	0.86
WebText baseline	0.77
GPT-2	0.76

Figure 14: POS sequence vetting

	Correctly tagged	Truly novel
Wikitext baseline	0.87	0.93 (74/80)
LSTM	0.90	0.85 (58/68)
Transformer	0.84	0.89 (62/70)
TXL	0.86	0.85 (63/74)
WebText baseline	0.78	0.95 (71/75)
GPT-2	0.75	0.90 (54/60)

Figure 15: Dependency arc vetting

nia California-California-Mission (unk), was originally intended to help rebuild damaged homes and buildings in rural New Mexico in search of new ways to improve access to water and housing to help meet the needs of displaced persons who need to work for their communities.

(ii) *Training example:* The **exploratory** parts of the game feature a series of arcane codes and glyphs , treasure maps and chests , and secret rooms .

(25) Novel as a verb

a. (i) *TXL:* The team used a series of diodes to **diode** the electrical component of the circuit board...

(ii) *Training example:* Devices with this type of negative resistance include the tunnel **diode** , resonant tunneling **diode** , lambda **diode** , Gunn **diode** , and dynatron oscillators .

Novel determiner: In the language acquisition literature, one question that has received focus is about determiner-noun pairs (Pine and Lieven, 1997; Yang, 2013): do children have a productive rule that allows them to combine any determiner with any noun, or have they only memorized particular determiner-noun pairs (e.g., *the dog*, *a cat*)? We investigate this general topic by turning to the *det* dependency arc to see if there are any nouns appearing with a determiner they have not appeared with during training. We see examples from all models, showing that models have not simply memorized particular determiner-noun pairs.

	Correctly tagged	Truly novel
Wikitext baseline	0.82	0.82 (50/61)
LSTM	0.85	0.81 (35/43)
Transformer	0.79	0.80 (48/56)
TXL	0.88	0.90 (56/62)
WebText baseline	0.71	0.85 (56/66)
GPT-2	0.74	0.79 (52/66)

Figure 16: Dependency role vetting

(26) *the* $\rightarrow$ *a*

a. (i) *LSTM:* The upper part of the tooth, from the upper part of the tooth, is covered with **a** coarse **keratin**, usually to the base of the teeth.

(ii) *Training example:* The scales of birds are composed of **the** same **keratin** as beaks , claws , and spurs

b. (i) *Transformer:* called The unk featured **a** "very clever" female **tilefish** who played a role that she had previously played in the pilot episode.

(ii) *Training example:* **the tilefish** will continue to expand its burrow in the sediment throughout its life .

c. (i) *Transformer-XL:* **A** medium-sized **zebrafish**, this small tadpole has two openings in its snout: a small opening on the top of its head and a large opening at the bottom of its head.

(ii) *Training example:* The main species used is **the zebrafish**

d. *GPT-2:* A traditional Tread is a "V"shaped surface that has **a** solid **groundplate** in both the outboard sidewalls.

(27) *a* → *the*

a. (i) *LSTM:* The pawn base (sometimes called a unk) is in a position called unk, where most of **the castling** can be compared to the middlegame.

(ii) *Training example:* It is critical to keep the king safe from dangerous possibilities . **A** correctly timed **castling** can often enhance this .

b. (i) *Transformer:* Porcini have a mild taste, and are edible when cooked, although there is confusion over their culinary value from other edible mushrooms, such as the dried and fermented edulis edulis, the ground pepper, or **the bouillon** with which they are cooked.

(ii) *Training example:* Polish cuisine offers **a** ruby @-@ colored beetroot **bouillon** known as barszcz <unk> <unk> , or clear red borscht .

c. (i) *Transformer-XL:* **The selfie** featuring Jackson, the character he portrayed in the game, is widely considered a popular culture phenomenon.

(ii) *Training example:* **A selfie** featuring Venkatesh and his son Arjun was released as well .

d. *GPT-2:* In my case I just changed my coverband (no case is needed) because **the** new **coverband** covers the headphone jacks on my iPod touch!

Novel dependency roles: We focus on 2 dependency roles analyzed in past work (Fodor and Pylyshyn, 1988; Kim and Linzen, 2020): do we ever see a word used as a subject that has never appeared as a subject before, or a word used as a direct object that has never appeared as a direct object before? Across models, the answer is yes.

(28) **Novel as a subject**

a. (i) *LSTM:* By the first week on opening weekend, domestic distribution had accumulated over $21.7 million, and its first Thursday **matinee** had already reached a lower $5.8 million mark.

(ii) *Training example:* Columnist Leonard Pitts of The Miami Herald described his difficulty sitting down to read a book , in which he felt like he " was getting away with something , like when you slip out of the office to catch a **matinee** " .

b. (i) *Transformer:* Except for the tail, some members of unk are relatively large, and the **hindlimb** has large finger bones (unk) on each foot.

(ii) *Training example:* Coelurus had a relatively long neck and torso due to its long vertebrae , a long slender **hindlimb** due to its long metatarsus , and potentially a small slender skull .

c. (i) *Transformer-XL:* With the loss of Mike Hargrove to a knee injury and the retirement of Charlie Haas as player-manager, Dodgers broadcaster Dave **Kingman** had to seek some sort of replacement.

(ii) *Training example:* The interstate continues to head north until it reaches **Kingman** .

d. *GPT-2:* **Makiza** is a junior researcher at the Institute for Race Relations and an independent journalist.

(29) **Novel as a direct object**

a. (i) *LSTM:* The Chicago Tribune (although it was also the first in a magazine to read The Urantia Book) was the first to feature George **Saintsbury**, a 19th-century art historian; he felt that it was based on the original text of the Bible.

(ii) *Training example:* **Saintsbury** insists it is " beyond all question one of the very greatest of [Balzac 's] works " .

b. (i) *Transformer:* A pot of the great Celtic goddess Mithras at Ribeira da Ribeira dates to the 6th or 7th century AD, and finds a copy of a manuscript showing **Mithras** and about thirty other deities bound around him or her, which included gods such as the goddess Venus and the goddess of the earth and water.

(ii) *Training example:* Eldred fled to the world that he and **Mithras** stand on .

c. (i) *Transformer-XL:* The Popular Republican Party, led by James Whitcomb Riley, denounced the **NLRB** and argued that they have no right to interfere in the decisions of the National Labor Relations Board.

(ii) *Training example:* The **NLRB** appealed to the Supreme Court of the United States

d. *GPT-2:* This is what I wanted, so I used the original Bomber King storyline as the foundation for creating **RoboWarrior**.

Novel argument structures: Finally we look at cases where verbs have an argument structure they never have in the training set, where the argument structure defines the set of arguments that the verb has (e.g., *laughed* in *Alex laughed* has the argument structure *active subject*; *explained* in *The rules were explained* has the argument structure *passive subject*; and *saw* in *The doctor saw the lawyer* has the argument structure *subject,direct object*). We first checked if a verb that had only ever appeared as active was used as passive or vice versa, and found a small number of these. We also found one example of a shift from transitive to intransitive, where TXL used the verb *suffuses* as an intransitive verb when it had always appeared as transitive in the training set.

(30) Active to passive

a. (i) *LSTM:* By the afternoon of September 9, the large cyclone system had been **re-strengthened**, with an increase in the surface circulation of the storm.

(ii) *Training example:* The storm quickly **re-strengthened** early on September 20 , but transitioned into an extratropical cyclone on September 21 .

b. (i) *LSTM:* However, the Enterprise crew in sickbay was allowed the craft to be **re-docked**.

(ii) *Training example:* According to mission rules , Orion would have then **re-docked** with Casper

c. (i) *Transformer-XL:* In 1864, a large part of the state of Georgia was **seceded** from the United States to form the Confederate States of America.

(ii) *Training example:* Pakistan withdrew in 1972 after East Pakistan **seceded** and became Bangladesh in 26th March 1971 .

(31) Passive to active

a. (i) *Transformer:* They then **drydocked** at Sasebo on 22 January 1916 to be fitted with an additional 4.5 cm / 40 anti-aircraft (AA) guns.

(ii) *Training example:* Ostfriesland was **drydocked** in Wilhelmshaven for repairs , which lasted until 26 July

(32) Transitive to intransitive

a. (i) *Transformer-XL:* On March 4, 2009, The New York Times published its own review from journalist Mike Brennan, who thought the novel's structure was too similar to several of his best known works, but praised the "rich and evocative language" that **suffuses**, as well as the plot as "vivid and convincing."

(ii) *Training example:* Genuine poetry **suffuses** them , and they are scored with brilliance and resource .

We observed no confirmed instances of a verb being generated with a transitive usage when it had only been intransitive in training. We also observed no instances of novel dative alternations: using an indirect object (e.g., *I gave **them** a book*) with a verb that, in training, always used a prepositional object (e.g., *I gave a book **to them***), or vice versa, though see (Hawkins et al., 2020) for a study of the dative alternation in language models. Note that the baselines also contained no confirmed novel transitivity or dative alternations, meaning that the lack of such examples in the models should not be interpreted as evidence that they are incapable of making such generalizations, since these types of novelty are rare even in human-generated text.

N Morphology categorization

In this section, we give an overview of all the novel unigrams generated by GPT-2. Specifically, we analyze our samples of text generated by the largest size of GPT-2 using top-40 sampling, pooling together the samples from all 4 of our prompt lengths. We then tokenized the generated text using the Moses tokenizer (Koehn et al., 2007) and analyzed all of the resulting tokens that never appeared in GPT-2's training set. We also performed

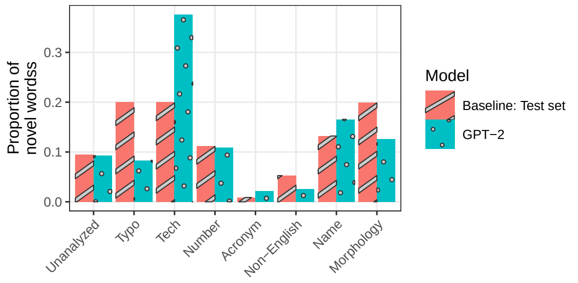


Figure 17: Categorization of the novel words generated by GPT-2

the same procedure for our baseline WebText text. All of the analyses described here were performed manually by one of the authors who is a native English speaker with training in linguistics; we used manual annotation because automatic methods are unlikely to be trustworthy for the rare, long-tail phenomena that give rise to novel unigrams in the generated text.

We define a word as a space-delimited token (after the text has been tokenized with the Moses tokenizer). This means that a word counts as novel as long as this exact form of the word has never appeared during training, but in some cases it may not deviate that much from words in training; for example, it is possible that the word appears during training but with a different capitalization, or with different punctuation (e.g., as two words in a row instead of being hyphenated). We consider all of these to be instances of novelty because, from the model's perspective, all of them are encoded differently.

In case some readers would prefer a stricter definition of novelty, we searched GPT-2's training data for all of the morphologically-novel words discussed in the main text to see whether any form of that word appeared in the training set. Specifically, we lowercased the morphologically-novel word and removed all characters other than the 26 Roman letters (i.e., we removed all spaces, punctuation, and non-Roman characters), and then searched for the resulting string in the training set after formatting it in the same way. This method was chosen to be extremely thorough; in addition to returning all reasonable ways of formatting words that we could think of (e.g., capitalized/uncapitalized, with punctuation/without punctuation), it also returns many false positives (e.g., a word being split across multiple unrelated words, such as *Welsh* appearing inside *vowel shift*), so we manually inspected all portions of the training set that were identified as candidate matches for the morphologically novel words. Figure 18 shows the results, using the following categorization:

- *None:* We verified that the word never occurs in the training set in any form.
- *Unknown:* Checking whether the word occurs was infeasible because there were too many candidates to check
- *Spaced:* The word is a compound word that occurs but with its components separated by a space
- *Within larger word:* The word occurs within a larger word in the training set
- *Capitalized:* The word occurs but with a different capitalization than it was generated with

Returning to the full set of novel unigrams (not those just discussed in the main paper), we divided all of the novel unigrams into 8 broad categories. Figure 17 gives an overview of how common each category is in the GPT-2-generated text and in the baseline; in the rest of this section, we describe these categories and give more details about the types of words that make them up.

N.1 Unanalyzed

This category is composed of examples for which we could not discern any internal structure. Of the 838 such examples, most of them (658) were strings of random characters:

(33) a. src = " **jE4B9BpL9KWv** "
b. fvw vnvh **qvwq** wvqw , kqw .

This category also contains 180 examples, like the following, which are pronounceable words but for which we could not find any apparent meaning or internal structure.

(34) a. **Narcow** . Mike Tyson says he 's " on track...
b. **Aumulule** may have been constructed of wool or coarse flax .

N.2 Typo

The generated text contains 754 typographical errors. Most of these errors (706 of them) are caused

Word	Related appearances in the training set
1099es	none
752th	unknown
anti-Tunisian	none
bondbreaking	spaced (*bond breaking*)
Brazilianism	within larger word (*anti-Brazilianism*)
Brazilianisms	none
Clevelandans	unknown
Clevelandians	none
co-causation	none
cookying	none
epineopterygoid	none
FOIA-requesters	spaced (*FOIA requesters*)
Fourteenthly	none
Fowleses	none
front-floor	spaced (*front floor*)
genoshaans	none
genoshans	capitalized (*Genoshans*)
Hamiltonans	none
headswaters	none
Huamangas	none
Klymits	unknown
LHAWs	unknown
load-samples	unknown
M-Sinks	unknown
mushrooms-related	none
no-knockout	spaced (*no knockout*)
proto-poetry	different punctuation (*protopoetry*)
re-nitrification	none
ridiculousities	none
Sarrats	none
SQLes	unknown
Sub-epineopterygoid	none
Summission	none
tearingros	none
tearro	unknown
tearsros	unknown
Thirteenthly	none
tornro	unknown
Torpexes	none
townites	none
un-competition	unknown

Figure 18: Variant forms of words discussed in the main text that display morphological novelty.

by improper spacing and/or punctuation, sometimes as an error of the language model (35a) and other times as an error of the tokenizer that we used to post-process the text (35b).

(35) a. I ’m going to have to use a different material besides **plastic.Here** ’s the front and back

b. This material may not be published , broadcast , rewritten , or **redistributed.**

The remaining 48 typos are ones that involve misspelled words; e.g., (36a) has *efforteless* instead of *effortless*, and (36b) has *oceansic* instead of *oceanic*.

(36) a. I use these words to give these thoughts a form in such an **efforteless** and natural way that I may never really know where the person is coming from

b. Because of the potential influence of atmospheric and **oceansic** carbon dioxide on the global carbon cycle

The generated text contains far fewer typos than the baseline, human-generated text does. We conjecture that this difference arises because humans generate text character-by-character, which might create more opportunities for typos than the subword-based generation process used by GPT-2.

N.3 Tech

The most common category of novel words generated by GPT-2 is technology-related terms including both URLs or parts of URLs[9] and words used in computer code (e.g., variable names) (37). The baseline text also contains a fair number of novel words in this category, but they are only about half as frequent as in the generated text.

(37) a. may now return the string “ < template name = ‘ **views.template.name** ’ > " .

b. $ success = $ auth - > **successMessageText** (‘Logged In ’) ;

c. row-reverse (columns) { **–widths** 2 – columns 2 ;

We do not make any systematic analysis of how such words are structured, but constructing them likely does require several types of string manipulation, including:

- “Affixation" such as adding *www* or *com* in

[9] We provide no examples of URLs in case the URLs currently, or in the future, lead to websites containing viruses or harmful content.

URLs, or adding the double dash used to introduce an argument in computer code

- Concatenation of multiple words, sometimes with a delimiting token such as a period, dash, or underscore
- Manipulation of case, particularly converting words entirely to lowercase for URLs, or the use of camelcase within code.

N.4 Number

GPT-2 generates several types of novel words that we classify as numbers. First are real numbers (672 of them; 38); these are generally well-formed, but note the one example that improperly starts with a 0 (38c).

(38) a. to $ 113.48 in trading while the S & P 500 rose 0.6 percent to **2,022.64** points .
b. from July 2008 through March 2015 , to make approximately $ **2,569,913** worth of unauthorized loans from military-owned financial institutions .
c. For the census year 2011 , there were 3,878,000 male-female households (**0,936,000** female households in same-sex couples) .

There are also 49 novel dates (39),[10] all of which form possible dates (i.e., there are no impossible dates such as November 31 or July 33). Some of these dates use the form month-date-year (39d–39e), while others use the form date-month-year (39f–39g).

(39) a. **Jun-14-16**
b. **2013-11-12T18**
c. **27-April-2018**
d. **11-28-1998**
e. **04-24-1987**
f. **13-08-2025**
g. **15-08-2027**

There are also 75 instances of a number plus a unit, such as the following:

(40) a. Minimum Water Pressure : **2.5atm**
b. Tags : ch.5 , ch.5.1 , **ch.5.2** , ch.6 , **ch.6.1** , ch.6.2
c. Available OS Memory : **8147MB** RAM

Finally, GPT-2 generates many novel phone numbers (190 of them), mostly following the 10-digit (or 11-digit) format used for North American phone numbers. These phone numbers appear with multiple formats: XXX-XXX-XXXX, 1-XXX-XXX-XXXX, XXX.XXX.XXXX, or (XXX) XXX-XXXX.[11]

North American phone numbers can also be expressed with some letters instead of numbers, and we see GPT-2 formatting 4 phone numbers in this way too. However, in all 4 cases, the correspondences between the letters and the numbers provided afterward are incorrect.

N.5 Acronym

In our generated text, there are 195 examples of acronyms. Of these, 75 appear along with the full version of what the acronym stands for; some examples are below.

(41) a. The Money Funders International Group (**MFIG**)
b. the Cathedral Development Strategy Review Group (**CDSRG**)
c. The National Census and Statistics Bureau (**NCBSB**)
d. The Nigerian Institute for Demographic and Social Research (**NIDRS**)

N.6 Non-English

Even though GPT-2 is primarily an English model, sometimes it generates words that are clearly meant to be interpreted as words in another language. In many cases the intended language is specified: these languages include Arabic, Aramaic, Bulgarian, Czech, Dutch, Esperanto, French, German, Greek, Hebrew, Hungarian, Icelandic, Inuktitut, Japanese, Latin, Mandarin, Middle Dutch, Old English, Old Frisian, Old Norse, Proto-Germanic, Quenya, Russian, Spanish, Swedish, and Swiss German. We do not attempt any investigation of whether these words are actually valid words in the languages they are purported to be in, or whether they have the meanings that they have been attributed.

Within this category we also include the 6 instances of GPT-2 providing pronunciation hints. These are generally somewhat similar to the true pronunciations of the words they are meant to correspond to, but are far from perfect:

[10] Note that we are only analyzing novel unigrams; there are likely more novel dates that include spaces, along the lines of *June 28, 2021*.

[11] We do not provide examples of generated phone numbers: even though the generated phone numbers are not in the training set, they may still belong to a person who would not want their phone number published in this paper.

(42) a. Voltaire (pronounced **VORE-ey**)
b. Tom Paine (pronounced **TOHR-in**)

N.7 Name

We see 1,499 novel names in the generated text. 7 are names of languages or dialects (43). 655 are online usernames.[12] 39 are first names (the first word in a multi-word name) (44), 220 are last names (the last word in a multi-word name) (45), while 150 are sole names (the only word in a single-word name) (46).[13] 11 are names of groups of people (47). 120 are names of places (48). 21 are names for which we were unsure of the type of name (e.g., name of a person or of a place?) (49). 206 were corporate names: names of products, companies, or organizations (50). 70 are parts of scientific names for species (51).

(43) a. Some of the dialects are known as Inuit Nunaat , Inuit Yupik , Inuit **Inupialik**
b. **Naahauhau** : One of the most widely used dialects in Northland with many dialects of

(44) a. **Abduraziz** Akhmadov
b. **Armidio** F. Santos
c. **Daryousu** Nyamitwe
d. **Deshpur** Singh
e. **Elianza** Dina
f. **Gerewyn** Davies
g. **Ikkuei** Chisato
h. **Jeeun** Jang
i. **Mari-Reijo** Aho
j. **Qi-Juan** Wang
k. **Rene-Laurent** Bédard
l. **Sapsiso** Nkomo
m. **Tsu-Bin** Mehta
n. **Xian-Lai** Yuan

(45) a. Ahmet **Davutoiu**
b. Anders **Kjølveberg**
c. Celine **Darpien**
d. Daniel **Vosschevsky**
e. David M. **Kornkrantz**
f. Eka **Khodayeva**
g. Emily **Kinshel**
h. Hasan **Husefije**
i. Hooman **Mashharipour**
j. Isabel **Pérez-Reyes**
k. Jacques **Prévens**
l. Jafar **al-Jubaidi**
m. José Luis **Hernández-Méndez**
n. Kim **Ki-nong**
o. Kurcan **Çapça**
p. Liam **Gwozdan-Morgan**
q. Linda **Nix-Hogar**
r. Linda **Kanjiyagawa**
s. Liu **Wu-Qin**
t. Matt **Wijeysinga**
u. Muhrhar **Rükeli**
v. Njoki **Rakotombe**
w. Reinhard **Dansberger**
x. Tom **McCallaghan**
y. Vladimir **Chmovarov**
z. Zhang **Gongtian**

(46) a. one day he returned to his wife , **Safwanah**
b. an extra terrapin named **Puddleguy**
c. **Sutsabaiesan** has been working with the Maple Leafs

(47) a. There existed a community of people called " **Srisleha** " within Jodhpur
b. served under the leadership of Lord Shadowsun of the Clan **Esh-Or**

(48) a. transferred to a police facility in **Hohenkreuz** , near Bremen
b. **Knettestown**
c. the small town of **Vazir-Dzorbievsk**
d. travel through the world of " **Runerealm** "
e. on the bridge over the river **Vosne-et-Oise**
f. a 30min train ride from **Kuparupu**
g. in the town of **Uraojpuram** in Tamil Nadu
h. in the town of **Mikhaylenka** , central Ukraine
i. many other locations at **Chikamagaloor** where sandstone rock have been quarried
j. A **Dagenhamshire** Police spokesperson
k. **Wetah** Besar , a city that is famous for its beach
l. a short walk to **Úlfsmörk** , **Þjórsmörk** and **ístmörk**

(49) a. the amulet of **Peryob**
b. the ring of **Beryemuonk**

(50) a. told **Skyfoto** TV

[12] We provide no examples of usernames in case the examples are real people's usernames.

[13] Note that cultures differ in terms of the interpretation of first/last names; for instance, in some cultures, one's first name is one's family name, while in other cultures one's last name is one's family name. Because we only have access to the words generated by the models, but no definitive access to their meanings, we classify names based on their position (first vs. last) rather than their meaning (family vs. given).

b. Photo Credits **RedStock** / Pixland / Getty Images
c. Sign up for **OurEarth** magazine
d. a fitness track from Adidas called **Sportstiq** is
e. **Playstone** Entertainment
f. fintech companies such as **Crowdfin**

(51) a. Acanthopanax **achariensis**
b. Nothofagus **acantholytica**
c. The genus **Sarconyx** contains three species
d. S. **schirani**

Beyond classifying the types of names that are present, we do not analyze the structure of these names. However, this would be potentially interesting to look into; it is non-trivial to construct a word that is pronounceable and "looks like" a name, such as generating sequences of syllables that look like plausible names of humans (44 and 45), or like the Greek- and Latin-based words used for scientific names (51). Further, some of these names require some more structured string manipulations, such as lowercasing and concatenating words to form usernames, or using affixes within names such as *al-* (45l) and *Mc-* (45x) in names of people, or *-town* in place names (48b).

N.8 Morphology

Finally, we analyze the words that use morphology—linguistic derivation of novel words. Our categorization largely follows that of *The Cambridge Grammar of the English Language* (Huddleston and Pullum, 2001), chapters 18 (Palmer et al., 2001) and 19 (Bauer and Huddleston, 2001).

N.8.1 Inflectional morphology

Inflectional morphology is the inflection of a word—not creating a new word, rather simply changing some grammatical feature of the word. Figure 19 gives an overview of the inflectional morphology found among the novel words in our text samples.

Nouns There are two types of inflected forms for English nouns: plurals and possessives. Both occur in the generated text.

The generated text includes the plural forms of common nouns (52a–52b), proper names (52c), and abbreviations (52d). Most of the plurals are formed using the *-s* form of the plural morpheme, but there are some formed with *-es*, such as (52e). Though the English language features a few pluralization processes other than the *-(e)s* suffix (e.g., changing *-um* to *-a*, as in *bacterium/bacteria*), none of them are employed in the sample of generated text that we analyzed.

(52) **Plurals**
a. The Commission has taken no action with respect to the issuance of **nonpublications** in this or any other area .
b. A new generation of leadership , the so-called " Indonesia **Dreamists** , " include the candidates of the Movement of New Forces and PKS-Partido Komunist ,
c. Ann spent her days cycling up and down her city by train . The **Sarrats** were lucky to have her as part of their lives
d. The U.S.R.A. alleged that the **C.O.O.s** of approximately 350 horse racing clubs conspired in 1990
e. Torpex NEWLINE NEWLINE **Torpexes** are small hardpoints found on smaller ships

There are also possessive forms of both proper nouns (53a–53b) and common nouns (53c–53d).[14] The observed possessives involve both forms of the possessive morpheme: *'s* as in (53a–53c), and *'* as in (53d).

(53) **Possessives**
a. According to a report by UK accounting firm Deloitte , **Fregoli 's** pizza delivery service achieved $ 250m of revenues in 2014
b. or from their own personal maps as an alternative to **MapMyRide 's** online maps .
c. The **shillelagh 's** name comes from the word " shillelagh , " a military term meaning
d. We hope that what we uncover in our **census-takers '** reports will lead to a better understanding of how to best serve New York City

Adjectives English adjectives can be inflected for grade (also known as degree). This can take the form of a suffix (*-er* or *-est*), or a preceding word (*more* or *most*). The generated text contains no instances of adjectives inflected with the suffixes,

[14] While we mostly focus only on novel unigrams in this section, these possessives are bigrams, as our tokenizer separates the *'s* from the rest of the word.

				Count in generated text	Count in baseline
Inflectional morphology				243	388
	Nouns			210	331
		Plurals		74	116
		Possessives		136	215
	Adjectives			8	21
		Comparatives		6	10
			-er	0	0
			more	6	10
		Superlatives		2	11
			-est	0	4
			most	2	7
	Verbs			25	36
		-ed		12	27
		-ing		13	9
		-s		0	0

Figure 19: Inflectional morphology

but there were a few examples with the separate words *more* (54) and *most* (55).[15]

(54) **Comparatives**

a. We 've become much **more androcentric**

b. and other news reports indicate that this particular incident represents a trend among the **more socially-diverse** , tolerant areas of Japan

(55) **Superlatives**

a. rabbits we adopted , which ate wheat products , are now two of the **most gluten-intolerant** cats we know .

b. the “ F-14 class of fighters , the **most stealth-capable** combat aircraft ever manufactured

Verbs There are several examples of generated verbs inflected with the suffix *-ed*; in all cases, these examples are used as passive verbs (56) or past participles (57)—that is, there are no instances where it is used to form a novel active past tense.

(56) **Passive uses of *-ed***

a. to the awesomeness of the Swiss to the point of being **Swissified**

b. Most traffic signals are either signified with a sign or **roadmarked** with either an arrow or flashing yellow lights .

(57) **Participial uses of *-ed***

a. In addition to the **self-profiled** and self-labeled population ,

b. the word in its more **colloquialised** form is most often associated with the 2000s

There are also examples using *-ing*:

(58) ***-ing***

a. and GCHQ was spying on this too (for example , by “ **cookying** " certain searches on the internet) .

b. I hear how you worry that you 're being judgmental or **judgmentalizing**

c. will revert back to the normal way of updating your OS , which includes completely **re-restoring** all applications and settings to their default settings

There are no novel verbs with the last remaining major type of inflectional verb morphology, namely the suffix *-s* used to form 3rd-person present-tense verbs.

N.8.2 Lexical word-formation

Under this heading, we include all processes that create novel words (as opposed to novel inflections

[15]These *more* and *most* examples are the other cases of bigrams that we consider in this section, besides the possessives. All other examples in this section are unigrams.

				Count in generated text	Count in baseline
Compounds				781	1112
	Dephrasal			84	174
		n-p-n		15	24
		Other dephrasal		69	150
	Coordinative			115	79
		List		3	3
		Other coordinative		112	76
	Compound nouns			243	399
		N-centered		230	372
			N + N	160	259
			Adj + N	52	83
			Adv + N	1	2
			V + N	1	8
			N + postpositive	8	7
			N + Num	8	13
		Verb-centered		13	27
			N + deverbal N	13	27
	Compound adjectives			281	391
		Adj-centered		65	114
			N + Adj	51	43
			Adv + Adj	14	70
			Adj + postpositive	0	1
		N-centered		45	57
			Num + N	44	52
			P + N	1	3
			N + P	0	2
		V-centered		171	220
			Adj + gerund	5	4
			Adj + passive	10	7
			Adj + V	3	2
			N + gerund	45	69
			N + passive	105	133
			V + P	2	5
			V + postpositive	1	0
	Compound verbs			1	9
		V-centered		1	9
			P + V	1	0
			Adv + V	0	3
			N + V	0	4
			V + V	0	1
			V + Adv	0	1
	Neoclassical compounds			57	60

Figure 20: Compounds

		Count in generated text	Count in baseline
Diminutives		6	13
	-ling	1	1
	-lite	2	0
	-y	0	1
	demi-	0	1
	down-	0	1
	hemi-	1	0
	micro-	0	3
	mini-	1	0
	nano-	1	4
	semi-	0	1
	under-	0	1
Augmentatives		4	15
	hyper-	0	3
	mega-	0	3
	over-	1	1
	super-	2	4
	ultra-	1	2
	up-	0	2

Figure 21: Derivational affixation, part 1 of 4: Diminutives and augmentatives

of existing words).

Compounds The text generated by GPT-2 contains many types of compound words, summarized in Figure 20. First are dephrasal compounds, created by converting an entire phrase into a single word by conjoining its words with hyphens. Many of these are of the form *noun-preposition-noun* (59a–59b), but there are many more with a wide range of other structures (59c–59f).

(59) **Dephrasal compounds**
a. It looks as if we are going to continue our **spending-to-student** ratios going up
b. it became , in **comics-as-genre-wholesale** parlance , a “ reset " of sorts .
c. the governor signed a similar law — with a different version of the controversial “ **we-told-you** " law — into law less than a month ago .
d. Here ’s just a snippet of what Demetrio said he sees in his **two-year-and-counting** career of working at the MTA
e. So I feel no obligation to go along with Jimmy in finding the **good-guy-but-evil-man** , unless he ’s a good guy .
f. that happened Friday was that President Obama and President Xi stopped playing the “ it **’s-a-civilized-country-and-we-should-just-accept-each-other** " game and started working together .

Another common type of compound is coordinative compounds, in which all elements of the compound have an equal status, and the meaning of the whole compound is essentially the meaning arrived by joining all of its elements with the word *and*. A few of these compounds are ones that we categorize as lists because the order of the elements matter (60a), but most are of the more general type that involves joining the elements with no obvious reason for the ordering (60b–60k).

(60) **Coordinative compounds**
a. So we go **left-right-right** and left-right-left .
b. According to the CIA , the **Soviet-Azerbaijani** relationship had become increasingly “ intense " in recent years .
c. What really happened on Tatooine ? The **Hutt-Kling-Mandalorian** trade dispute ?
d. Fox ’s **Titans-Cowboys** game at 12 : 00

		Count in generated text	Count in baseline
Location in time and space		16	39
	ante-	0	1
	circum-	0	1
	cis-	0	1
	cross-	1	1
	inter-	0	1
	intra-	0	1
	meta-	0	1
	off-	0	2
	outer-	0	2
	post-	6	10
	pre-	2	11
	proto-	1	4
	sub-	3	1
	supra-	0	1
	trans-	3	1
Negatives and reversatives		32	55
	anti-	8	11
	counter-	0	4
	de-	1	3
	dis-	2	1
	dys-	1	0
	in-	1	1
	mis-	0	1
	no-	1	0
	non-	15	25
	un-	3	9
Positives and repetitives		9	12
	pro-	2	4
	re-	7	8
Residency		14	11
	-an	12	11
	-ite	2	0

Figure 22: Derivational affixation, part 2 of 4: Location in time and space; negatives and reversatives; positives and repetitives; and residency morphemes.

		Count in generated text	Count in baseline
Verb or adjective to noun		20	21
	-ant	0	1
	-ation	5	1
	-er	7	9
	-ion	1	1
	-ity	3	2
	-ment	1	1
	-ness	3	6
Noun to noun		28	34
	-dom	3	1
	-ese	0	1
	-fest	0	2
	-gate	0	1
	-ia	0	2
	-iana	1	0
	-ism	4	9
	-ist	5	3
	-ista	1	0
	-land	0	2
	-osphere	0	1
	-ry	2	0
	-scape	1	0
	-ship	1	0
	-thon	0	2
	-verse	2	2
	co-	3	0
	e-	3	5
	ex-	2	2
	mal-	0	1

Figure 23: Derivational affixation, part 3 of 4: Noun derivation

		Count in generated text	Count in baseline
Adjective creation		26	51
	-able	2	2
	-al	4	3
	-ary	0	1
	-esque	0	5
	-ian	2	3
	-ic	3	6
	-inal	0	1
	-ine	2	0
	-ish	2	6
	-less	2	1
	-like	6	14
	-th	2	1
	-y	0	8
	nigh-	1	0
Verb creation		7	9
	-ate	0	2
	-ify	3	1
	-ise	1	1
	-ize	2	3
	a-	0	1
	con-	1	0
	out-	0	1
Adverb creation		2	10
	-ly	2	6
	-wise	0	4
Other		3	0
	-mon	1	0
	digi-	2	0

Figure 24: Derivational affixation, part 4 of 4: Adjective derivation, verb derivation, adverb derivation, and miscellaneous affixes.

		Count in generated text	Count in baseline
Character manipulation		5	38
	Capitalization	0	10
	Creative spelling	0	6
	Letter repetition	1	4
	Nervousness	2	6
	Onomatopoeia	2	13
Portmanteau words		3	31
Schm- reduplication		0	1

Figure 25: Non-affix-based morphology.

p.m. on September 9 averaged 8.5 million viewers

e. the moons of Jupiter provided the foundation for what came to be called the " **Newton-Laplace-Einstein** model of the solar system .

f. It has its roots in the " **anarchism-capitalism** " debate

g. the Northern **Norway-Iceland** border

h. proton beam producing roughly the same number of gluon plasma as is observed for the **proton-photon** collisions ;

i. Pa had a little iron **grill-barbecue** outside

j. And it was very easy for him to be a **painter-scientist** , because he used nature and art .

k. the most successful form of handstand execution known to man : the **Krzyzewski-Frohwirth-Kacmar-Cunningham** Combination

Most of the rest of the compounds we categorize based on the part of speech of the compound that is created. First are compound nouns (which can, in some circumstances be used as adjectives), which can be created in a variety of ways:

(61) **Compound nouns: Noun + noun**

a. He tries his **dad-voice** again

b. Healthy Sesame **Almond-Fluff** Pasta

c. in order to save trillions of precious **life-months** and lives .

d. Not all carriers offer their own **bagshare** program

e. The best times to use cooking oil on your grill / **grilltop** include

f. This plant is very similar to **Flower-potweed** .

g. using calipers and then used while the lens was still fresh to obtain samples for **x-ray-analysis** ,

h. Some bears have more of a **densize** capacity than others .

i. The Pongo pygmaeus vocal repertoire is highly complex and reflects the complexity of the **callsong** behaviour of this species ,

j. Upon hitting a **ball-cage** , the player becomes unable to move and must jump

k. He refers to them as " **hill-elves** " , or Eistlaes , in Quenya ;

(62) **Compound nouns: Adjective + noun**

a. So what happens to a **raw-fruit** diet with no dairy ?

b. The paper concludes by proposing possible further extensions toward an actual " **computational-self** theory "

c. and the front and the back shall be of **bluework** with a gold frame for the robe .

(63) **Compound nouns: Adverb + noun:** a press release on the topic from **then-CMA** president Tom McGinnis .

(64) **Compound nouns: Verb + noun:** only the latter can actually be seen with a **sputterball**

(65) **Compound nouns: Noun + postpositive modifier**

a. at least 1 in 3 employed adults reported having a **job-to-be-held** .

b. Tags : **Apprentice-in-training** , Magic , Apprentice-in-training , Magic Academy , **mentor-in-training**

(66) **Compound nouns: Noun + number**

a. At the beginning of the Siege of Eayn , **Eylan-3** was attacked by the Republic .

b. A test flight at the end of 2017 will conduct **Thaad-AMM-3**

(67) **Compound nouns: Noun + deverbal noun**

a. SoundFont is available on apt for installation , by replacing the **font-encoder** .

b. **Poster-maker** David Hill has now stepped forward

c. but in recent years the practice has taken on a more popular reputation as the **fuel-extraction** method of choice for many small businesses hoping to make a profit

Next are compound adjectives, which also can be constructed in a variety of ways:

(68) **Compound adjectives: Noun + adjective**

a. Today , I have another record to share with my **wing-weary** students and colleagues :

b. Last year , Rangers hitters batted.243 against a league-leading 1,200 or more innings of **relief-friendly** MLB pitching .

c. Mexico seemed to be having a bit more trouble with its **backpass-heavy** tactics than in previous World Cups .

d. and Dan Dennett as they talk about their work together and the world of“ **science-neutral** " philosophy .

(69) **Compound adjectives: Adverb + adjective**

a. Both wheelsets feature a **slightly-slender** offset .

b. a non-distorted copy may be displayed in at least a **partially-circular** area

(70) **Compound adjectives: Number + Noun**

a. On-highway travel is permitted between campus and off-highway travel only within a **two-quarter-mile** radius of any of the following :

b. and the presence of militants in many areas along its 2,500-kilometre (**1,460-mile**) perimeter .

c. out of Carver-Hawkeye Arena to the new $ 70 million , **7,400-seat** Assembly Hall ,

d. It boasts a **4-stereo** sound system with a pair of subwoofers mounted a few inches

(71) **Compound adjective: Preposition + Noun** Therefore , in the present study we performed non-invasive (**in-participants**)

(72) **Compound adjectives: Adjective/Adverb + gerund**

a. something that is false or **false-nurturing**

b. to create clear-cut attacking situations and a heavy , **sometimes-lurching** attack

(73) **Compound adjectives: Adjective/adverb + passive**

a. The very people who should know better , are making **scientifically-skewed** claims .

b. Eligible for Broadcasting **Governmental-Funded** Eligibility

(74) **Compound adjectives: Adjective + verb**

a. The hotend does not need any special hardware and has **easy-set** screws .

(75) **Compound adjectives: Noun + gerund/participle**

a. Extremely powerful ultrasonic cleaner that cleans debris and **debris-containing** debris out of every inch of surface it hits

b. teachers are given incentives to target a “ **school-chasing** " strategy .

c. characters themselves , but of who those characters mean to a different segment of the **superhero-watching** public .

d. One of the principal problems with **carbohydrate-eating** is how it stimulates blood sugar and fat stores ,

e. island of Oahu , Hawaii , where they began a life of mushroom picking and **mushroom-making** , later known as the “ Mycological Arts " .

f. Kenow hopes to set up a **loon-viewing** deck in a few years to better identify the small , mostly nocturnal birds .

(76) **Compound adjectives: Noun + past participle**

a. browser features also come as Microsoft has added Internet Explorer 11 to the list of **IE10-based** browsers .

b. and we ’ll be navigating to the folder

where your project and **composer-managed** libraries are in .
c. NASA had to move closer to developing the systems necessary : a **propellant-fed** rocket , an ascent engine , an emergency escape system and a crew evacuation system
d. **Flag-shaped** flag sold in Alameda County
e. Soy is often used in place of dairy-containing dairy products in **dairy-reduced** diets and is being studied as a possible supplement for osteoporosis prevention .

(77) **Compound adjectives: Verb + preposition**
a. If you aren 't using **nail-on** nails and tape , you also can glue them to the wood .

(78) **Compound adjectives: Verb + postpositive modifier**
Torpexes are typically of a weapon **mount-only** .

Finally, there is exactly one example of a compound verb:

(79) **Compound verb: Preposition + verb** One side of the farm has been taken **off-grazing** over the winter

The last major class of compounds is neoclassical compounds, which are formed by adding one or more Greek- or Latin-based affixes to a stem. Occasionally these involve stacking several affixes; e.g., (80e) uses the prefixes *epi-*, *neo-*, and *ptery-*.

(80) **Neoclassical compounds**
a. These reservoirs supply a special type of rock called **granophyllite** that was originally made up of the granites from which the Svalbard region was originally
b. Hydrochlorothiazide contains hydroxychlorothiazoxide which , together with **hydroxychlorothiazide** and hydroxypyridazine hydrochloride , form a yellow crystalline powder .
c. **Thermolithotrophic** bacteria are Gram-negative bacteria that produce a wide range of enzymes and are more
d. the author of the books “ Ethical and **Religio-Economic** Consequences of American Wars Since 1898 : A Primer for Political Policy Makers "
e. Pelagic **epineopterygoid**

Derivational affixes Derivational affixes are prefixes or suffixes that are added to a word to create a new word. The derivational affixes that we observe are summarized in Figures 21 through 24.

The first two categories of such affixes are diminutives such as *-ling*, *-lite*, and *mini-* (81), which make the meaning smaller along some dimension, and augmentatives such as *super-* and *ultra-* (82) which make the meaning larger. (Note that the diminishing/augmenting effects of these affixes are not very apparent in these examples).

(81) **Diminutives**
a. THE REAL **FISHLING** POND
b. This social democracy will be a “ **state-lite** " , in the sense that the king will not be in a situation where
c. the three **Mini-Compounders** eventually defeated the pair

(82) **Augmentatives**
a. The Professor went on to design new **super-fabricated** paper products
b. It was revealed in **Ultra-Sized** G ! that there was originally a card named “ Berserker " in the set

Several derivational affixes in the generated text indicate location in space or time, specifically *cross-*, *post-*, *pre-*, *proto-*, *sub-*, and *trans-*:

(83) **Location in time and space**
a. A large drawbridge (with a **cross-reinforcement** mechanism to support it)
b. Bach 's music can be classified , along with Beethoven , as “ **post-Johann** Sebastian styles "
c. However , if the GC CPUs are slower than they need to be due to emulation-based issues , there is often the option of going back to a **pre-Emulation** model
d. The “ **proto-poetry** " of modern times is the “ hyperbole " spoken by Shakespeare
e. Other companions , enemies and **sub-Bosses**
f. the **Trans-Dniestria** railway

Observed morphemes that negate or reverse the meaning of the stem include *anti-*, *de-*, *dis-*, *in-*, *no-*, *non-*, and *un-*:

(84) **Negatives and reversatives**
a. the most prominent member of the **anti-St-Pierre** camp
b. He calls this phenomenon the “ Great **De-**

concentration

c. Chaos is seen today not as an all-encompassing disorder and **disorderlessness** but as a complex interplay of extremes

d. He eventually found one of the Inhuman **Inveterans** in an abandoned building .

e. adding an optional " **no-knockout** " version that also removes the knockout effect

f. they immediately pointed out my **non-Arabic-sounding** pronunciation of Arabic words .

g. 15 Proposed 8A NEWLINE NEWLINE 16 **Unproposed** 9A

In contrast to the negatives and reversatives, there is also one prefix that expresses positive sentiment (*pro-*) and one that expresses repetition (*re-*):

(85) **Positive or repetitive**

a. In this study , we investigated the anxiogenic-like and **pro-immobility** and anxiolytic-like effects

b. They go through the process of **re-nitrification** that gives them a new supply of nitrogen

Two observed morphemes, *-an* and *-ite*, attach to a place name or description to create a word meaning a resident of that place:

(86) **Residency morphemes**

a. The Riot ' at the Cleveland Institute of Art — a time when **Clevelandians** riot for a variety of reasons

b. In the meantime , the **Aquallans** were forced to contend with the Supermen .

c. made them dwell in the land of Nod (**Nodites**)

d. Inspired by the successes of social movements such as the Paris Commune and the American labor movement in the 1890s , these small **townites** were inspired to take action to create a new economic alternative

We categorize most of the rest of the affixes by the part of speech of the word they create (and potentially also the part of speech of the stem, though this categorization is imprecise because some morphemes, such as *-ism*, can take multiple different parts of speech as a stem). We observe affixes that create a noun from a verb or an adjective (87); affixes that create a noun from another noun (88); affixes that create an adjective from a different part of speech (89); affixes that create a verb from a different part of speech such as a noun or adjective (90); and an affix that generates an adverb from an adjective (91).

(87) **Verb or adjective to noun**

a. *-ation*: Now , there 's also a difference between an **IKEA-ification** and a **Vogue-ification** .

b. *-er/-or*: This is what it means to be a **data-aggregator**

c. *-ity*: **Behaviourality** was evaluated during the first and first and second consecutive day in groups of 10

d. *-ment*: The most anticipated part of the redevelopment will be the " **Prapliftment** Zone " - the area where the residences are put up .

e. *-ness*: Chaos is seen today not as an all-encompassing disorder and **disorderlessness** but as a complex interplay of extremes

(88) **Noun to noun**

a. *-dom*: a new version of a zombie meme began sprouting from **quackdom** .

b. *-ism*: **YIMBYISM** is all about supporting and increasing the density (and the number of units)

c. *-ist*: it might be worth looking at the responses from Pythonistas or the **JavaScriptists**

d. *-ry*: and has an opportunity to become one of the Scouts who upholds the values of **scoutry** at any age

e. *-scape*: When we compiled this list of **famescapes** , it wasn 't about proving the validity of this view .

f. *-ship*: Charlie is a good horse : good at pulling a plow and excellent at **foalship** training .

g. *-verse*: The Smurfs vs. The **Smurfverse**

h. *co-*: we are witnessing a major movement away from a capitalist workplace toward a " **co-workplace** " , where people are working together to solve problems and create goods and services

i. *e-*: Scooters with electronic systems include the eGo-T and **e-Scooter** .

(89) **Adjective-forming affixes**

a. *-able*: **Gemcraftable**

b. *-ian*: is truly related to the mass of the

bosonian Higgs-like particles

c. *-ic*: isolinear memory processor ; isolation field ; **isonetic** wave

d. *-ish*: the **Moldbug-ish** strategy to end " civic nationalism " .

e. *-less*: The dual injection systems on this engine feature **injector-less** , single-port fuel injection systems .

f. *-like*: the console lacks any form of **Blu-ray-like** functionality

g. *-th*: the **752th** year of the Hebrew Calendar

(90) **Verb-forming affixes**

a. *-ify*: In this case , that 's " vogue " for " **vogue-ification** "

b. *-ize*: I hear how you worry that you 're being judgmental or **judgmentalizing**

(91) **Adverb formation**

a. *-ly*: **Thirteenthly** : NEWLINE NEWLINE It was narrated that al-Tirmidhi and ' Atiya narrated

Finally, there are 2 morphemes that are used in the context of a specific television franchise, the Digimon franchise, to coin words relating to that franchise:

(92) **Television-franchise-specific morphemes**

a. *digi-*: with 50 **digifications**

b. *-mon*: Black Ogremon can digivolve to Gabumon with **Shoujoumon**

Character manipulation Some novel words involve manipulations at the level of individual characters (summarized in Figure 25). First is the use of letter repetition, either to elongate a word for emphasis (93a) or to indicate nervousness (93b). Second is the creation of onomatopoeias, in which each letter is meant to represent a sound in the real world (94).

(93) **Letter repetition**

a. **Youuuuuuuuu** ! !

b. " W-what are y-you **m-meant-** "

(94) **Onomatopoeia**

make a humming noise that sounds like " tch-tch " or as low a " **ka-a-la** " and a " **hwa-hwa** " while searching for food

Portmanteau Finally, there are three generated words that could potentially be viewed as portmanteau words: (95a) is a blend of *Disqus* and *etiquette*, (95b) is a blend of *pizza* and *apocalypse*, and (95c) is a blend of *gel* and *popsicle*. However, it is possible that the model does not view these as blends but rather as compounds: it may have learned a morpheme *-iquette* that means "etiquette," a morpheme *-pocalypse* that means "apocalypse," and a morpheme *-sicle* that means "popsicle." Indeed, Pinter et al. (2020) found that LMs perform poorly at handling portmanteau words, which would support the hypothesis that, in the cases we have observed, GPT-2 is not handling these words as portmanteau words.

Predicted form	*-s* is correct	*-es* is correct
-s	67	0
-es	2	5

Figure 26: Confusion matrix for plurals

(95) **Portmanteau words**

a. Please make sure to read the **Disquisquette** before leaving comments .

b. What 's in the future for the ' **Pizza-Pocalypse** ' ?

c. and took two gels (Molly and a **gelsicle**) .

O Additional examples for the analyses

Here we provide additional examples from the manual analyses discussed in Section 7.

O.1 Plurals

To form English plurals, it is necessary to choose between *-s* and *-es*. Of the 74 plurals in our analyzed sample, the model made the correct choice for 72 of them, only getting wrong the two shown in (96) (Figure 26). The 5 cases where it correctly predicted *-es* instead of the more common *-s* are in (97).

(96) Incorrect plurals

a. in the same way as regular **1099es** .

b. Why do **SQLes** have to change

(97) Correct usage of *-es*

a. more than two million " **metches** "

b. when Mr. Fowles asked him about it . The **Fowleses** ' lawyer

c. **Electories** by Jason Ditz

d. the **ridiculousities** of war

e. **Torpexes** are small hardpoints found on smaller ships

Predicted form	-*'s* is correct	-*'* is correct
-*'s*	125	1
-*'*	0	10

Figure 27: Confusion matrix for possessives

O.2 Possessives

Forming possessives in English requires a choice between two possible forms, *'s* and *'*. All but one of the generated possessives had a correct form (Figure 27); the incorrect example is in (98a). Some examples of the model correctly using the apostrophe-only form of the possessive are in (99).[16]

(98) Incorrect possessive
  a. known as **watchmakers 's** timepieces

(99) Correct plural possessives
  a. our **census-takers '** reports
  b. The **Fowleses '** lawyer
  c. the **genoshans '** fear of the Andromeda Initiative
  d. The **Flexagons '** unique patterns

O.3 Acronyms

Of the 195 novel acronyms in our generated text, 75 appear with the full version of what the acronym stands for. In 21 of the 75 cases, the acronym is a suitable abbreviation for the shortened form (100); in the remaining 54 cases, the acronym is not a suitable abbreviation. Often, the errors involve having extra letters in the acronym (101), often repeats of letters that appear elsewhere in the acronym (101a through 101d), but not always (101e through 101g). Other types of errors include the omission of a letter (102a and 102b), having letters out of order (102b and 102c), and replacing a letter with a different, incorrect letter (102d and 102e). In a few cases, the generated acronym differs from its expansion to a more substantial degree (102f).

(100) a. The Money Funders International Group (**MFIG**)
  b. the Cathedral Development Strategy Review Group (**CDSRG**)
  c. the US-China Economic and Security Review Commission (**US-CESRC**)
  d. the West of England Cricket and Athletics Club (**WECAC**)

(101) a. The National Census and Statistics Bureau (**NCBSB**)
  b. the Parliamentary Joint Committee on Human Rights (**PJCHRC**)
  c. Koch Companies Public Sector (**KCPSP**)
  d. the American Academy of Pain Medicine (**AAAPM**)
  e. the Tennessee River Gorge (**TNWRG**)
  f. Health Resources and Services Administration (**HRSDA**)
  g. the International Bank of Settlements (**IBNSA**)

(102) a. Ruby Interpreters and Ruby Users Committee (**RIRC**)
  b. the Gulf Coast Disaster Recovery Task Force (**GCDFT**)
  c. The Nigerian Institute for Demographic and Social Research (**NIDRS**)
  d. the National Coalition of Latino Elected Officials (**NCLEP**)
  e. Extremely Large Interactive Neutrino Experiment (**ELIGO**)
  f. Angola Democratic League (**ULANL**)

O.4 Examples of incorrect morphology

Here we review some common sources of morphological errors among novel words generated by GPT-2.

Incorrect stem changes: Some of the errors arise from GPT-2 making unwarranted changes to the stem. In (103a), a name that was consistently spelled as *Shuutou* earlier in the passage has been given an extra *u* in its possessive form. In (103b) and (103c), GPT-2 has generated two different words that are most likely intended to be the adjectival form of the word *Pentagon* (the headquarters of the US Department of Defense). Neither of these terms (*Pentagorean* and *Pentagran*) are plausible ways to turn *Pentagon* into an adjective; the most plausible correct form would be *Pentagonian*, which has in fact appeared in the training set. Thus, by using *Pentagorean* and *Pentagran*, GPT-2 is being inconsistent both with itself and with its training data.

(103) Uncalled-for changes to the stem:

[16] Note that, when the possessor ended with *-s* but was not plural, we allowed either form of the possessive, following variation in common usage (Huddleston and Pullum, 2001). Thus we count as correct both *This model is based on Adam **Ondrus 's** classifier* and *content offered on **NEXUS '** Website*.

a. **Shuuutou 's** flight speed
b. Pentagon and Cybersecurity ... **Pentagorean** nuclear arms
c. **Pentagran** war on the way

Inconsistent morphology: Beyond the Pentagon-based examples above, the suffix *-an* appears to be a common source of inconsistency for GPT-2. In (104a), the generated word for people from the town Hamilton is *Hamiltonan*, but later in the same generation it is formed as *Hamiltonian*. (104b) shows inconsistency in how to refer to residents of genosha, and (104c) and (104d) show 2 different ways from 2 different generations to refer to someone from Cleveland. None of these (except for *genoshaans*) count as ill-formed in Figure 2 because there is variability in how the *-(i)an* suffix can be applied, but the inconsistency is an issue. These examples display a different type of inconsistency from the inconsistency observed in prior work, namely inconsistency in how to apply morphology, as opposed to factual inconsistency (Welleck et al., 2019; Li et al., 2020).

(104) Inconsistent demonyms
a. as the percentage of **Hamiltonans** in the GTA increases , so does the number of people leaving the city ... almost a quarter of all <u>Hamiltonians</u>
b. the **genoshans** were the first of a number of species...the **genoshaans** managed to regain their homeworld
c. that **Clevelandans** are forced to endure
d. a time when **Clevelandians** riot

Missing sound changes: As mentioned above, the word *genoshaans* in (104b) is considered ill-formed. The reason is that it lacks the proper phonological change to the *-an* suffix that occurs when the stem ends with *-a*, namely of deleting one of the instances of *-a*. Another example of failing to make a sound change is in (105) in which *a* should instead be *an*.

(105) the $ 6 , $ 8 or $ **10-a-ounce** china cup cake

Plurals in compounds: A few ill-formed examples arise from using the plural form of a noun as the first part of a noun-noun compound; generally, the first noun in a noun-noun compound is the singular form of the noun, though there are exceptions in standard usage, so it is unclear if these should actually be viewed as errors.

(106) Plural in compound
a. “ common-sense " **guns-control** policies
b. The...rivers had their **headswaters** in a larger basin
c. **mushrooms-related** products

Overregularization: Some of the errors can be classified as overregularization: applying a linguistic process in an overly broad way. In (107a), the word *syllogist* has been created, presumably by changing the *-ism* ending from *syllogism* into *-ist*. In many cases it is valid to swap *-ism* and *-ist* (e.g., *tourism/tourist*, *optimism/optimist*), but not in this case. In (107b), the suffix *-th* has been applied to the number 752, even though numbers ending with 2 should instead get a different suffix, *-nd*.

(107) Overregularization
a. objections aimed against the **syllogist**
b. the **752th** year

O.5 Syntactic errors

Most of the novel words that GPT-2 generates fit properly into their syntactic context, but it does make some mistakes, a few of which are below. In (108a), *bat-washer* is used as a verb when its structure suggests it should be a noun (though this could potentially be valid given English's flexibility about parts of speech). In (108b), there is a noun-noun compound (*cyber-missiles shortfall*) with a plural noun as the first noun; typically, such compounds start with singular nouns, though there are some exceptions. In (108c), it should most likely either say *look anti-Tunisian* or *look like anti-Tunisians*. Finally, in (108d), *load-samples* is plural but is given a singular verb, *provides*.

(108) a. if I **bat-washer** it
b. its massive **cyber-missiles** shortfall
c. just to make our community look like **anti-Tunisian**
d. Slicex **load-samples** provides a single button

O.6 Agreement

Here we look at the novel plural nouns that GPT-2 generates to see whether the rest of the generated sentence observes the correct consequences of the word's plurality. First, (despite the one mistake in 108d), GPT-2 generally does well at providing plural verbs (underlined) to agree with novel plural nouns, whether the verb appears after the noun (109) or before the noun in the context of a question (110). In (111), it correctly uses a plural verb for both verbs that agree with the novel plural subject—a verb within the relative clause, and a verb after it. The correct agreement with the verb after the relative clause is especially impressive because, in both sentences, there are 3 singular "distractors" (italicized) between the subject and the verb.

(109) a. We know that **M-Sinks** need a target
b. when **Clevelandians** riot
c. the **Aquallans** were forced to contend with the Supermen
d. **Torpexes** are small hardpoints
e. Another indicator of the poverty that **Clevelandans** are forced to endure :
f. The **YR-2s** were designated simply as YR . 2 and were designated simply as YR . 2
g. **Hustlings** work when those people can work the job market for the required number of hours
h. It was revealed that the **genoshans** were the first of a number of species that the Andromeda Initiative had already studied
i. For some reason my old 1 / **1-01s** do not turn and drive with the shift lever down like my new ones do

(110) Why do **SQLes** have to change

(111) a. The **Huamangas** , who are descendants of indigenous people who lived on the *Isthmus* of *Tehuantepec* before it was covered by *farmland* , have been demanding that the federal government address the issue of climate change .
b. **FOIA-requesters** who think an *agency* has a good *reason* for withholding *information* are not always given a second opportunity to press their case .

O.7 Other plural-relevant syntax

Beyond agreement, syntactic consequences of plurality are observed in a few other places as well: in using the plural possessive form that is just an apostrophe instead of the singular form of *-'s* (112); in having the pronouns that are coreferential with the noun be plural as well (113); and in following determiners that require a plural noun (114).

(112) when Mr. Fowles asked him about it . The **Fowleses** ' lawyer , James F. Kelly ,

(113) a. I love **Klymits** , but it has been nearly impossible for us to find them in stores .
b. The **Sarrats** were lucky to have her as part of their lives
c. The **color-coats** are far more black & white than their predecessors

(114) a. as the Paris Commune and the American labor movement in the 1890s , these small **townites** were inspired to take action to create a new economic alternative
b. to help you understand why there are so many **Brazilianisms** in the English language as opposed to the Portuguese one

O.8 Incrementing/ordering

Here we provide the examples mentioned in the main text where GPT-2 successfully increments. In (115a), it increments numbers from *Firstly* to *Fourteenthly*, with the last two (*Thirteenthly* and *Fourteenthly*) being novel. In (115b), it increments the letters at the ends of variable names in computer code, going from *multiplyx* to *multiplyy* to *multiplyz*. Finally, in (115c), the prompt ends with an alphabetical list of companies, and GPT-2 continues this list, largely (though not entirely) staying in alphabetical order, including many novel words along the way (all in bold).

(115) a. Firstly : NEWLINE NEWLINE It is not permissible...Secondly : NEWLINE NEWLINE It was narrated that...Twelfthly : NEWLINE NEWLINE It was narrated from...**Thirteenthly** : NEWLINE NEWLINE It was narrated that al-Tirmidhi and ' Atiya narrated that... **Fourteenthly** : NEWLINE NEWLINE It was narrated that al-Shawkaari narrated that...
b. **multiplyx** = math. ceil (self. multiplier [0] * self. multiplier * 2.0) self. honey _ hive [0] . **multiplyy** = math. ceil (self. multiplier [1] * self. multiplier * 2.0) self. honey _ hive [1] . **multiplyz** = math. ceil (self.
c. BWS Buffalo C-Tech Can-Am Carre Revero Chameleon **Chintsoft** Claris **ClouDio CO-IoC** Cisco Connex Computer Comply Coopers **ConsalCO** Computer **CoreGear** Crestron Corel Dell D & H Digital Storm Dell Eizo Epson Epoch Exabyte Exponent Falcon Formica FreeNet G-Technology Gigaset

Gionis Gigabyte **GigaForce** **Glance-on** **Glass-einhard** Hauppauge Hauppauge **HPC-Link** HP i-mate IBM IBM PPro **ICON-S** ICOM **ICOM-ITIC** ICOM World **iBuyInte.Com** **iBuyMall** **Icom-U2** **iGK** Computers **IISkills** Inet IPC **iPlayIiMac** Iguana iRobot IceCool iPort **iSoftimage-iMac** Jackson JBL JEDI Kingston KeySmart Kinesis **Konica-Tek** Konika Logitec LSI Lenovo Maxtor Marvell Matrox Maxix **Maxixx** Microvision Microsoft **MoboModem** MSN MobileNet MyNet MXM **MyTouchNokia** **Nandmark** Neopan Tandem NoSQL Netside Netgear **NetVu** NetXtreme Network Technology Novell NTG Nyko **Noxel** Octave ODU Okidata Omnipath Online Data Osram Panasonic Patek Pivotal Prodigy **PQ-Link** Qorvo Quanta Quad-Core **Quadmark** RCA Rapoo **Rega-Link** **Regal-Link** Regelation **Rapidra** Regulus Redstone Redbox Reuleaux **RTI-Siemens** Semtec **Simlogic** Sintek Silicon Integrated Systems **Simvit** Simulink Smartcard **Sonic-com** South River Soundstar SPARC Super-Computer Systems SRT Systema TATA The Best Technology TLC Toshiba **Tungst-Sang-Tzu** TriQuint Travelstar **TSSTech** US Robotics **ViaL** Vision Vision VisionX WebPro Technologies WEIT Vantec VirnetX **VideoFusion** Vipnet **VSCare** VSP VTech Vortex Works Zebra Zeta ZXZ ZTE

O.9 Quotation marks

In GPT-2's text, as in human-written text, novel words are more likely to be enclosed in quotation marks than non-novel words. Some examples of novel words being in quotations are below:

(116) a. the wave function in the " **meganiverse** "
b. a " **co-workplace** ", where people are working together
c. " **Active-Passive-Inactive** " investing
d. the " **anarchism-capitalism** " debate
e. The " **proto-poetry** " of modern times
f. the " **un-competition** " that is happening as a result of rapid technological advances

O.10 Novel words with meanings that are suitable for their context

Below are some examples of novel words that are used in ways that are particularly well-suited for their semantic contexts.

(117) a. And it was very easy for him to be a **painter-scientist** , because he used nature and art .
b. The process is pretty simple : a cyclist is followed (suspect or **not-so-suspect**) from a safe distance
c. They go through the process of **re-nitrification** that gives them a new supply of nitrogen
d. These include the concept of ' **co-causation** ' , in which effects are thought to be caused by causes that act in parallel
e. The other thing of course is that the companies will be allowed to sell and to sell this information to the government in real time . This is what it means to be a **data-aggregator** and it is an interesting way to think about all of this .
f. Thirdly , a new unique feature , the " **bondbreaking** enchantment " , which renders any item cursed by the " Cursed item " bug inadmissible to the user and permanently breaks any binding .
g. we are witnessing a major movement away from a capitalist workplace toward a " **co-workplace** " , where people are working together to solve problems and create goods and services on a much smaller scale .

O.11 Novel words with meanings that are not suitable for their context

Below are some examples of novel words where there is clear evidence that the word is not used in a semantically-sensible way. In (118a), *judgmentalizing* is used in a way that suggests it should mean "being judgmental," but the word's structure should yield the meaning of "making someone judgmental." In (118b), *Brazilianism* is used to refer to an English term, not a Brazilian term. In (118c), Bittrex is referred to as Bittrex-like, but it is not standard to refer to something as being "like" itself. (118d) is contradictory because *disorderlessness* should be the opposite of disorder. (118e) is also contradictory because anti-catatonia effects are said to decrease motor activity, even though a decrease would be consistent with catatonia, not anti-catatonia. (118f) refers to the front floor, even though the floors of buildings are arranged vertically, so a building cannot have a front floor. In (118g), nitrification is referred to as a nitrate-deficient state, even though it most likely should be a nitrate-rich state. (118h) refers to a markdown-to-HTML converter having markdown output even though its output would actually be HTML. (118i) says that Internet Explorer 11 is built on Internet Explorer 10, even though most likely it would be viewed as a new browser, not a version of Internet Explorer 10. Finally, (118j) says that a no-knockout effect would enable people to be knocked out, which is contradictory.

(118) a. I hear how you worry that you 're being judgmental or **judgmentalizing** if you talk about how you 're leaving
b. An old school English term is a **Brazilianism** .
c. Blockstream 's shares were traded on Bittrex , a **Bittrex-like** cryptocurrency exchange .
d. Chaos is seen today not as an all-encompassing disorder and **disorderlessness** but as a complex interplay of extremes ,
e. Moreover , the main aim of this study was to investigate whether an anxiolytic effect of Vitex by increasing OAT % and OAE % is accompanied by anti-immobility and **anti-catatonia** effects by decreasing motor activity
f. wandering around the **front-floor** lobby of the Hotel Del Coronado
g. This **nitrate-deficient** state is called nitrification .
h. so you can use its **markdown-to-HTML** convertor with the markdown output format you prefer
i. Microsoft has added Internet Explorer 11 to the list of **IE10-based** browsers .
j. The only thing I 've done with my mod since then (well , maybe a little bit before) is adding an optional " **no-knockout** " version that also removes the knockout effect , so you can actually be knocked out again if you take enough damage .

O.12 Numbers

The analyzed sample of text includes 75 instances of a number plus a unit, such as the following:

(119) a. Minimum Water Pressure : **2.5atm**
b. Tags : ch.5 , ch.5.1 , **ch.5.2** , ch.6 , **ch.6.1** , ch.6.2
c. Available OS Memory : **8147MB** RAM

Several of these involve math, which gives us an opportunity to see whether GPT-2 understands the numbers it is using. (For convenience, we also include some that are classified as number-noun compounds, rather than numbers (121)). Generally it appears that GPT-2 does not understand the numbers it generates; all the examples involving math are included below. (120a) includes the computation of a difference between two numbers, where it is said that $1065mhz - 1030mhz = 90.5MHz$. Regardless of whether the MHz on the right hand side is interpreted as the same unit as mhz on the left hand side, or if the capitalization is viewed as meaningful, this computation is incorrect. (120b) involves a physical impossibility: a 4-milliliter container cannot hold 10.4 milliliters of juice. Meanwhile, (121a) and (121b) give quantities that are not strictly impossible but are highly unlikely: according to ESPN,[17] the fastest 40-yard dash in history was 4.22 seconds, making the 2.64-second time in (121a) implausibly fast; and the fastest three-cone drill in history was 6.28 seconds, making the 4.15-second time in (121b) also implausible.

The rest of the examples involve conversions between units, and generally the conversions are not equivalent. (121c) says that 1240 pounds equals 735 kilograms, when in fact it equals 562 kilograms. (121d) says that 2500 kilometers equals 1460 miles when in fact it equals 1553 miles (though this example is close enough to perhaps be reasonable). (120c) says that 975 milliliters equals 2.2 gallons when in fact it equals 0.26 gallons. (120d) says that 1 billion US dollars equals 610.9 million British pounds; this one is reasonable, because at current exchange rates it equals 718 million British pounds, so this is possible given fluctuations in exchange rates. Finally, examples (120e) through (120h) involve conversions between Kenyan shillings (KES) and British pounds (£). Across these examples (which all

[17] https://www.espn.com/nfl/story/_/id/28774721/nfl-combine-records-best-worst-performances

come from the same piece of generated text), we observe four different exchange rates: £1 = KES14.3 (120e); £1 = KES25 (120f); £1 = KES66.7 (120g); and £1 = KES200 (120h). Given this inconsistency, it appears that the model does not have any consistent meaning stored for these numbers.

(120) a. The highest speed in the XFX version is **1065mhz** , which is around **90.5MHz** higher than the 1030mhz in our testing .
b. the original 4ml tank holds **10.4ml** of e juice .
c. Water Tank Capacity : **975mL** (2.2 Gallons)
d. found Mr Mitchell guilty of a combined $ 1bil (£ **610.9m**) in damages and costs .
e. In a town like Kajiado that can cost up to **KES50** (£ 3.50) to find a taxi .
f. Prices range up to **KES100** (£ 4.00) a night
g. a bed in one of these rooms can cost **KES300** (£ 4.50) .
h. be prepared for the ride to cost you **KES200** (£ 2.50) .

(121) a. He posted a **2.64-second** 40-yard dash this spring
b. Mandarich ran a 4.29-second 40-yard dash and **4.15-second** three-cone at the NFL combine in March .
c. by a **1,240-lb** . (**735-kg**) device
d. along its 2,500-kilometre (**1,460-mile**) perimeter .

O.13 Phone numbers

In North American phone numbers, the first 3 digits of the phone number indicate the area where the phone number is from. In many cases, the context of a generated phone number makes it clear where the phone number is meant to be from. Does GPT-2 generate phone numbers appropriate for the places? Overall, we find 77 cases where the context makes a North American location clear and where the phone number is specific to a North American location. In 54 of these 77 cases (70%), the phone number and the contextually-specified place match. For instance, (122a)[18] uses the Seattle area code 612 and mentions Seattle; (122b) uses the New York City area code 212 for a location in New York City; (122c) uses the Baltimore area code 410 and mentions both Baltimore and the Baltimore-located Johns Hopkins Medical Institute (and its email abbreviation of *jhmi*); and (122d) uses the Pittsburgh area code 412 for a location in Pittsburgh. It appears, then, that GPT-2 has learned valid associations between area codes and locations.

In other cases, the phone numbers are somewhat geographically off: in both cases in (123), the context refers to a location in Philadelphia (either the city of Philadelphia, or the University of Pennsylvania, which is in Philadelphia), yet the phone number uses the Pittsburgh area code 412 (which GPT-2 had properly associated with Pittsburgh in (122d)). Philadelphia is in the same state as Pittsburgh, so this is not too far off; but the cities are nonetheless about 300 miles apart, so they are not geographically interchangeable. Finally, some cases are far off: (124a) refers to Colorado but uses the New Hampshire area code 603, and (124b) refers to Lubbock (a city in Texas) yet uses the area code 914, which is for Westchester County, New York. There are plausible reasons why a phone number might not match a location (e.g., because a person has moved), so it is not necessarily a big problem that only 70% of the cases matched; however, in the baseline text, 55 of the 56 cases matched (98%), suggesting that there should be a higher rate of matching than is observed.

(122) a. [redacted] : **612-6**[redacted]**-4**[redacted] or [redacted] @ seattletimes.com ;
b. Call (212) **3**[redacted]**-2**[redacted] to receive a customized quote for your own individual investment needs ; or contact us using our online or telephone services , or by mail at the address below : NEWLINE NEWLINE Money Funders International NEWLINE NEWLINE [redacted] NEWLINE NEWLINE New York , NY 10019
c. In fiscal year 2017 , Johns Hopkins Medicine provided health care to approximately 6.2 million residents of the Baltimore region... MEDIA CONTACT : [redacted] , **410-9**[redacted]**-6**[redacted] ; [redacted] @ jhmi.edu
d. [redacted]

[18] For all of the examples in this section, we redact information that could potentially provide contact information or addresses for real people.

Drive NEWLINE NEWLINE Pittsburgh , PA 15219-8011 NEWLINE NEWLINE (412) **4**[redacted]**-8**[redacted]

(123) a. Editor , Philadelphia City Paper NEWLINE NEWLINE e - [at] phillypagewriter ([at] phillycitypaper.com)) NEWLINE NEWLINE **412.3**[redacted]**.5**[redacted]

b. Editor : [redacted] , MD , MPH , Chief Medical Officer , University of Pennsylvania Health System . Corresponding contributor : [redacted] , [redacted] @ phshp.org , **412-3**[redacted]**-4**[redacted] .

(124) a. Colorado State Patrol : **603- 2**[redacted]**-3**[redacted] , [redacted] @ denverpost.com

b. Lubbock Police Department NEWLINE NEWLINE **914-5**[redacted]**-3**[redacted]

O.14 Generalization by composition

There are a few cases where GPT-2 generates a novel word whose stem never appears in training but does appear in the context (the prompt plus the previously-generated words). Specifically, there are 5 cases of GPT-2 pluralizing a novel word from its context (125), and 3 cases of it adding other affixes to a novel word from its context (126). We believe that these examples are best explained by composition: analogy requires some notion of similarity between the two word parts being swapped for each other, and it is unlikely that the model would have such similarity notions for a word stem it has never seen before. Thus, we think these examples are better understood as the model adding a prefix or suffix to a word from its context, without direct reference to another word that has that prefix or suffix—a form of composition.

(125) a. ... a major movement away from a capitalist workplace toward a " co-workplace " ... These types of **co-workplaces** , if truly integrated and facilitated , could potentially improve and strengthen the worker-worker relationship

b. both the Overdone and Overloved series ... Book No. 4 : The **Overloveds** are available now on both Amazon and Barnes and Noble , respectively .

c. during a fuedo ... Most tornros however use their bare hands in **fuedos** . This type of fighting allows the tearo to use his / her own strength

d. sort of support force . For example , using the LHAW to take out other **LHAWs** , or the machineguns of the new infantry YoRHa suits.YoRHa is currently one of the

e. An old school English term is a Brazilianism ... of these differences and discuss them to help you understand why there are so many **Brazilianisms** in the English language as opposed to the Portuguese one .

(126) a. They can also be local pain-dissipating systems (V4) (sometimes the V1 and V3 fibers also form V4) or **non-pain-dissipating** central pain systems (V5 ; the V1 and V3 and V4 fibers can also

b. m. nesino NEWLINE NEWLINE Pelagic epineopterygoid , S. kuramotoi (L.) dehayesii NEWLINE NEWLINE **Sub-epineopterygoid** , N. scapulatus (M.G.) alvarezii NEWLINE NEWLINE NEWLINE Subgenus Heteracarina contains three species

c. ... seen with a sputterball ... It is also very hard to escape a Vulcan warrior 's **sputterballing** .

O.15 Generalization by analogy

On the other hand, there are some cases that we believe are best explained by analogy. A first example is in (127):

(127) the same time as a new version of a zombie meme began sprouting from **quackdom** .

It is possible that *quackdom* was formed by adding the suffix *-dom* to the stem *quack*. However, *-dom* is a rare and idiosyncratic suffix, which makes it less likely (though, we stress, not impossible) that the model has learned a generic rule for combining it with words. On the other hand, there is a plausible path for the model to have generated it by analogy: the training set contains the word *hackdom*, and it also contains a large number of words (61 of them, to be exact) which contain *hack* and where changing *hack* to *quack* creates a different word that appears in training. For instance, the training set contains all of the following pairs of words: *hackademia* and *quackademia*; *hackfest* and *quackfest*; *hacktivist* and *quacktivist*; *hacktacular* and *quacktacular*; and *hackery* and *quackery*. From the distributional similarity between *hack* and *quack*, it is plausible that GPT-2 has learned similar representations for these two words, which would then allow it to generalize from *hackdom* to *quackdom*. It is still of course possible that *quackdom* was created by a compositional rule that appends *-dom* to a stem, but we believe that analogy from *hackdom* is a more plausible account: there is a clear path for that to happen, whereas the rarity and idiosyncrasy of the *-dom* suffix make it harder for a model to learn a general rule involving it.[19]

Another piece of generated text which we believe provides even clearer evidence for analogy is given in full in the next subsection. The prompt for this generation contains the real English word *torero* (borrowed from Spanish), which means "bullfighter." The generation then contains several alternate forms of this word (some of them with plural inflection added): *tearro*, *tornro*, *tearingros*, and *tearsros*. It appears, then, that GPT-2 has taken the word *torero* and then replaced the first 4 letters (*tore*) with other inflections of the verb *tear*: *tear*, *torn*, *tearing*, and *tears* (covering the full inflectional paradigm for this verb). Because there is no morphological process in English that can add *-ro* to verbs in this way, it is unlikely that these words were generated via composition; instead, we find it much more likely that they were generated via analogy.

One final argument for analogy comes from the stem changes shown in the main paper in the examples in (5): based on subword tokenization of the words generated in these examples, we believe it is likely that GPT-2 is swapping out one token for another. For example, *cookying* is tokenized as *cook* + *ying*. It would not make much sense to view this word's generation as starting with *cook* and then adding *-ying* to turn it to the progressive form of the verb, because the way to make *cook* progressive is to add *-ing* (creating *cooking*). Instead, *cookying* seems more likely to be explained by analogy with *cookies* (tokenized as *cook* + *-ies*) and *cookied* (tokenized as *cook* + *-ied*), where GPT-2 has learned that *-ies*, *-ied*, and *-ying* are related suffixes (as they are in many other words such as *copying* and *copies*).

[19]From looking at the words, it is tempting to think that the spelling overlap between *hack* and *quack* would also influence GPT-2 to treat them similarly. However, this spelling cannot play a direct role: *quack* is split into two tokens (*qu* + *ack*), but *hack* is a single token of its own, so *hack* and *quack* have no subword tokens in common. Though the spelling similarity cannot play a direct role, it still likely plays an indirect role: the fact that these two words sound similar means that they are both conducive to the same fanciful coinages like *hackademia* and *quackademia* that would not work as well with a word that does not end with *-ack*, and having so many similar coinages would certainly help push a model toward treating them similarly.

O.16 Full generation about toreros

This subsection (after the end of these italicized sentences) gives the full generation about toreros. The prompt from which this generation was generated includes both the word ***torero*** *(meaning a bullfighter) and its plural,* ***toreros****.*

over their heads (this is known as a “ fuego ") . These are made up of metal bars that can stop the bull from getting back into the ring . NEWLINE NEWLINE 2 . The toreros begin to fight the bull with a knife or stick NEWLINE NEWLINE 3 . When the **tearro** loses , he / she is killed by the bull . NEWLINE NEWLINE Do **tornros** wear clothes ? Are they hurt in any way during a fight ? Are **tornro** costumes the same colors as those of the bull ? What are the odds of winning ? NEWLINE NEWLINE A **tearros** wears a costume in which they wear a costume of some kind over their head . In Spanish bullfighting , the costume consists of a blue cloth with red trim , the sides of which are covered in leather . A tearro is then trained to become highly skilled in this style of bullfighting . The training takes 5-8 years , depending on the type of training . During this time , a tearro must learn to avoid the bull ’s powerful horns . In Spain and other Spanish-speaking countries , **tearingros** are taught to avoid their horns . However , in recent years bullfighters in this style have been performing stunts to get close to the bull , wearing tornro costumes and injuring themselves . This is known as ‘ racing ’ . The bull , for its part , can be more aggressive and more unpredictable than usual , and it could be easier to get a tearro in a dangerous situation . NEWLINE NEWLINE How do tornro ’s defend themselves ? The easiest way for a tearro to protect themselves from injuries is by using a knife or stick to protect themselves . Tornros will punch or kick at their adversaries . Most often , they will also use two heavy wooden sticks to push each other back . It also pays to recognize the type of weapon a tearro uses during a **fuedo** . Some tearros will use only their hands ; these are known as “ **handarros** , " and have long , sturdy fingers that can be used for many different tricks . Most tornros however use their bare hands in **fuedos** . This type of fighting allows the **tearo** to use his / her own strength . The use of bare hands can make a tearro extremely unpredictable , so it is important to watch carefully when a tearo begins to fight . NEWLINE NEWLINE Are tornros allowed to touch the bull during a fuedo ? NEWLINE NEWLINE When fueding , the bull always wins . Once the bull enters a fuedo , a tearo must let go . This is called el aire en nuevo (the " fall in the night ") . The bull then is ready for the next fuedo battle . NEWLINE NEWLINE So are all **tearsros** the same ? Tornros are one of the most widely known and documented breeds in the world . Tornros are beautiful in their own way ; these beautiful animals have a different set of needs than those of other breeds of dogs . They can only do so much to make up for the damage they do to people and other animals due to their aggressive nature . NEWLINE NEWLINE Tornros are also considered to be a more difficult breed to own than most . Not only are there many health problems that can arise with your pups , the cost of pups can be fairly expensive , even if you choose to feed and care for them the way the breed is supposed to be fed and cared for . NEWLINE NEWLINE This breed is also generally hard to train , as well as to deal with the other animals in your household , such as your pet rabbit , hamsters , guinea pigs , hamsters , guinea pigs , and rabbits . They can also be a bit on the rough side , especially around large cats . NEWLINE NEWLINE Tornros do make great pets for those who are willing to do their part in keeping the animal healthy and happy . Many owners with other pet breeds can learn how to cope with these issues as well . You don 't need a lot of love , training , and experience , but I do believe that a little bit can go a long way . NEWLINE NEWLINE There are many places that have a Tornros pup , especially in the US , where you can keep a pup . I know of two dog kennels in Texas that do take Tornros . There are also many pet shops in Texas that will take these type dogs . Also , some people love the look of a beautiful Tornros and want to adopt one as their pets . These types of people often contact those businesses to help them look for a Tornros . NEWLINE NEWLINE If you haven 't been able before to have one of these wonderful dogs with you , I 'm sure you will be able to find a home for one in your home . You probably already know many things that you can do to make sure you have a good fit for a Tornros .

P Overregularization and nonce CVC forms

An influential observation in cognitive science is that, when a word is irregular, children sometimes produce a form predicted by the relevant regular rule instead of saying the correct irregular exception (e.g., saying *goed* instead of *went*). It is typically assumed that overregularized forms such as *goed* would not appear in the child's input, such that the usage of such forms indicates novelty on the child's part. Would such forms also count as examples of novelty for GPT-2? To answer this question, we used Wiktionary's list of English verbs with irregular past-tense forms,[20] excluding multi-morphemic words (such as *unbind* or *forbear*), auxiliaries (such as *can* and *will*), verbs which are only irregular in their past participle but not their past tense form (e.g., *shave* and *sew*), archaic verbs (e.g., *clepe* and *nim*), and words for which the regular-rule-produced past tense is a common English word, even if unrelated.[21] This left a list of 92 English verbs with irregular past tenses, and for all 92 of them, the form predicted by the regular rule appears in the WebText training set. Thus, if GPT-2 were to use any of these regular forms, such as *beginned* or *thinked*, it would not be strong evidence for overregularization, as it could instead be the case that it is simply copying something it has seen before. We do not see any such forms in the sample we analyze in Section 7 (generated with top-40 sampling), but we do see some in text generated with other decoding methods, a few of which are given below:

(128) **Overregularized past-tense forms generated by GPT-2 (decoding method in parentheses)**

a. We *builded* a bath (temperature=0.9)

b. The Arsenal manager has admitted that his team should have been more positive in order to exploit Arsenal 's " *drawed*

[20] https://en.wiktionary.org/wiki/Appendix:English_irregular_verbs

[21] In some cases, this is an alternate past tense of the verb in question: although some use *besought* and *dove* as the past tense forms of *beseech* and *dive*, others use the regular forms *beseeched* and *dived*. In other cases this is the past tense of another verb: though one meaning of *ring* has *rang* as its past tense, another has *ringed*; and though the past tense of *sing* is *sang*, the regular form *singed* is the past tense of a different verb, *singe*. Finally, in some cases, the regularly-predicted past tense is simply a different word such as *seed*, as in the seed of a plant, and *leaded*, as in leaded gasoline.

out " game . (top-0.9)

c. and their disciples after did already *finded* them out (top-1.0)

d. Nick led us in rebounding and he *layed* down a lot of offensive rebounds . (top-1.0)

e. The most egregious example of TGI-18 failure is documented in a recent class action lawsuit filed by a group of soldiers , who say that the failure to distinguish between mental and physical fitness , as *seeked* in the National Physical Fitness Test (NPFT) ... (temperature=0.9)

f. The shadow *shaked* in pain as black energy were sucked out hit it . (top-800)

g. Two rookies — Antonio Richardson of the Chargers and George Wilson of the Browns — *sitted* out a game this year because of suspensions . (top-0.9)

Other cognitive literature relies on a different type of word being assumed to be novel: words that are phonologically well-formed but that do not happen to be real English words. For instance, in the classic wug test, Berko (1958) tested whether children could form the plural of the made-up word *wug*. The motivation for using made-up words was that children would not have seen these words before, so, if they correctly formed the plural *wugs*, it would show that they knew English's plural formation rule and had not simply memorized plural forms they had seen. Can we similarly assume that such made-up words are novel for GPT-2?

To answer this question, we considered all possible words of the form consonant-vowel-consonant where the vowel could be *a*, *e*, *i*, *o*, or *u*, and each consonant could be any of the remaining letters except *y*; the two consonants could be the same as each other or different from each other. For each of these words, we also generated what its plural form would be, assuming that the word was a singular noun. For wourds ending with *j*, *s*, *x*, or *z*, we formed the plural by adding *-es*;[22] otherwise, we added *-s*.

From these singular/plural pairs, we excluded all words for which the singular and/or plural form is a real English word, as determined by whether it is present in the NLTK word list.[23] We also excluded all pairs for which the plural ends with *-es* and where removing only the final *s* creates a word in the word list. After these exclusions, 1339 nonsense words remained.

Of these 1339 words, there were 765 (57%) for which both the singular and the plural form occurred in the training set. For the remaining 574 words (43%), only the singular form occurred in the training set. There were no words for which the singular form did not occur in training (because it turns out that there is a part of the training set that lists all possible sequences of 3 letters).

These nonsense word results are not as extreme as the overregularization results, as there are plenty of nonsense words that would truly be novel for the model. Nonetheless, many of them are not novel, so it is not safe to assume that a word will be novel for GPT-2 solely because it is not listed in an English dictionary: it is necessary to check the training set to confirm that it is novel.

[22] In English, some words that end with *s* or *z* can optionally have this letter doubled in the plural; e.g., the plural of *bus* can be *buses* or *busses*, and the plural of *fez* can be *fezes* or *fezzes*. We did not consider such consonant doubling in this analysis.

[23] https://www.nltk.org/