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, USA [2]Microsoft Research, USA
[3]Johns Hopkins University, USA [4]New York University, USA [5]Meta AI, USA
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; 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 shortcoming: The generated text could have been copied from the LM's training data, in which case it does not provide clear evidence for 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

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

[†] Work done while at Microsoft Research.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 652–670, 2023. https://doi.org/10.1162/tacl_a_00567
Action Editor: Anette Frank. Submission batch: 9/2022; Revision batch: 1/2023; Published 6/2023.

called RAVEN (**RA**ting **VE**rbal **N**ovelty).[1,2] These analyses cover both sequential structure (n-grams) 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 83% 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 1.5 to 3.2 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 it *produces* 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 (§7.1)
- 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

[1] `https://github.com/tommccoy1/raven` (last accessed March 18, 2023).

[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-generating bird, the parrot.

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 touched on by studying whether Transformers copy large n-grams when generating language (Brown et al., 2020; Lee et al., 2022b; Kandpal et al., 2022) or code (Chen et al., 2021; Ziegler, 2021). 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

[3]This view even extends beyond the research community: A 2021 webcomic by Zach Weinersmith (`https://languagelog.ldc.upenn.edu/nll/?p=52293`; last accessed March 18, 2023) includes an AI system exclaiming, ''The fools don't realize how many of my coherent phrases are verbatim from training data!''

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. Pannitto and Herbelot (2020), Meister and Cotterell (2021), and Yamakoshi et al. (2022) 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: 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 Section A in our online supplement.[4] Throughout this paper, unless otherwise stated, *GPT-2* refers to GPT2-XL.

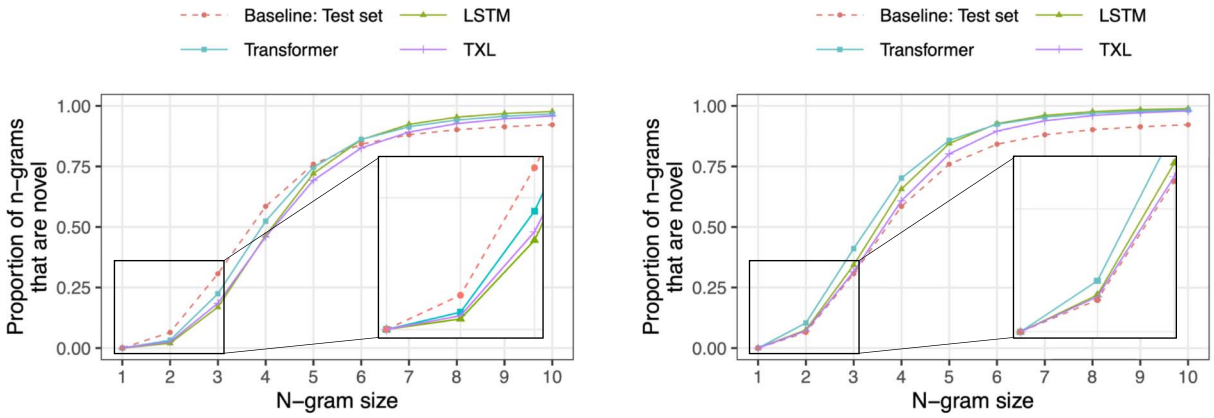
Prompts: To generate text from a model, we input a prompt drawn from that model's test set, which comes from the same distribution as its training set. For Wikitext-103, we use 1000 prompts of length 512 words and have models generate 1000 words following the prompt. For WebText, we use 1000 prompts 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).

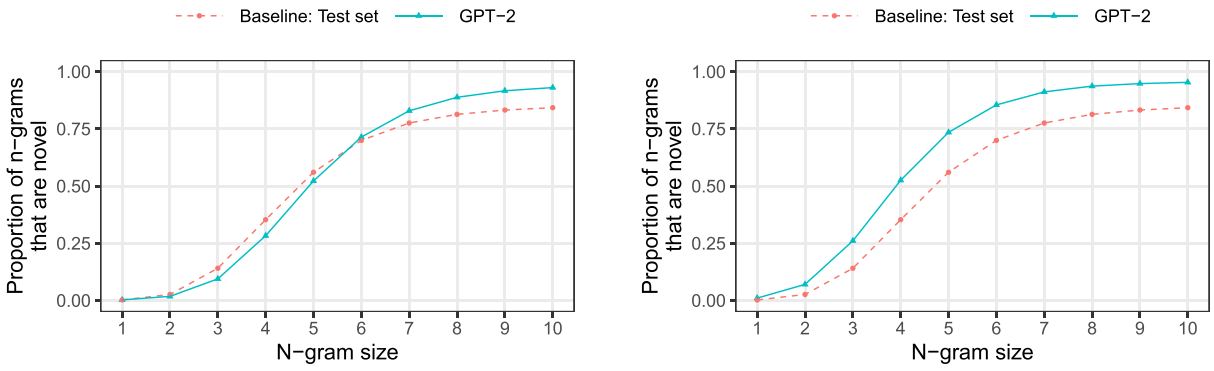
Instead, the main decoding scheme that we use is top-k sampling with $k = 40$, where the model's distribution is truncated to the 40 highest-ranked tokens then renormalized and sampled from. We chose top-40 sampling because it is what Radford et al. (2019) used for GPT-2 and what Dai et al. (2019) used 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://github.com/tommccoy1/raven/blob/main/raven_supplementary_materials.pdf (last accessed March 18, 2023).



(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.

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 shows the LM's unaltered distribution. The next section then gives an investigation of a wider range of decoding methods.

Findings: For $n > 6$, LM-generated n-grams are almost always novel, both for top-40 sampling and pure sampling (Figure 1). For smaller n-grams, these decoding methods diverge: Small n-grams generated with top-40 sampling are less novel than small n-grams in the human-generated baseline, 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 tokens but otherwise does not break 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. 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 from each model's test set. When using pure sampling, models are more novel than the baseline across n-gram sizes. With 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 producing novel bigrams and trigrams. However, for larger n-grams ($n > 6$), the models are *more* novel than the 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 LSTM and TXL are less novel for small n-grams than the Transformer (Figures 1a and 1b, insets). 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.

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 rules will not* ***be***), 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 supplement Section E for untruncated results).

Using this score, we investigated a range of decoding methods. The supplement (Section F) 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

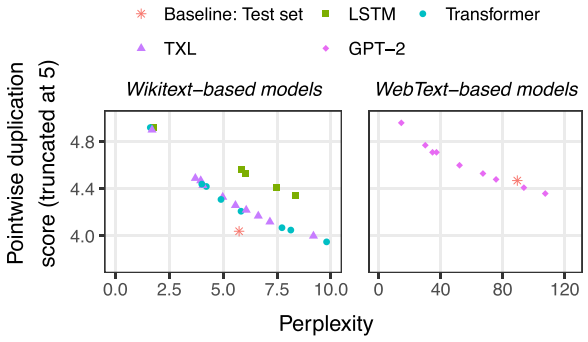


Figure 2: Manipulations of the decoding scheme that result in higher-quality text (i.e., lower perplexity; x-axis) also result in decreased novelty (i.e., a greater degree of duplication; y-axis). Each point shows a different decoding scheme.

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.

Details: Models rarely duplicate n-grams larger than 10 tokens; for all models, fewer than 5% of 10-grams are duplicated. 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 supplement (Section D) 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.

5.4 How Does Model Size Affect Novelty?

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

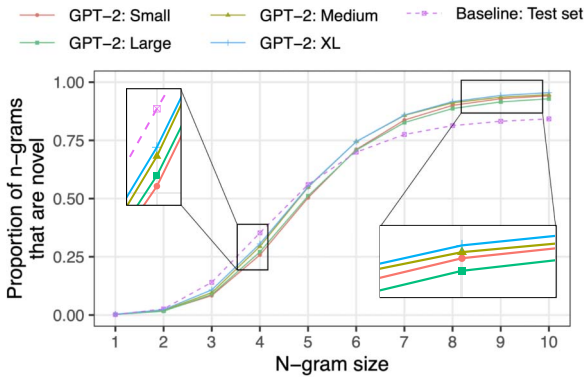


Figure 3: Effect of model size.

	POS seq.	Parse struct.	Dep. arcs	Dep. roles
Wiki baseline	0.82	0.82	0.13	0.0053
LSTM	0.86	0.87	0.07	0.0016
Transformer	0.84	0.85	0.08	0.0025
TXL	0.83	0.83	0.07	0.0021
Web baseline	0.63	0.65	0.05	0.0018
GPT-2	0.65	0.67	0.03	0.0011

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

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 sizes of GPT-2 (all using top-40 samping). Across n-gram sizes, the most novel model is GPT-2 XL; however, GPT-2 Medium is more novel than GPT-2 Large. Therefore, the effect of size on novelty is inconsistent.

5.5 Other n-Gram Analyses

Additional analyses are in the supplement (Sections H, I, and J). 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 supplement Section K). 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.

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

- **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 subtrees, 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).

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.

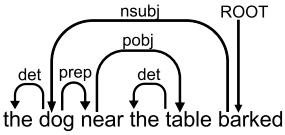
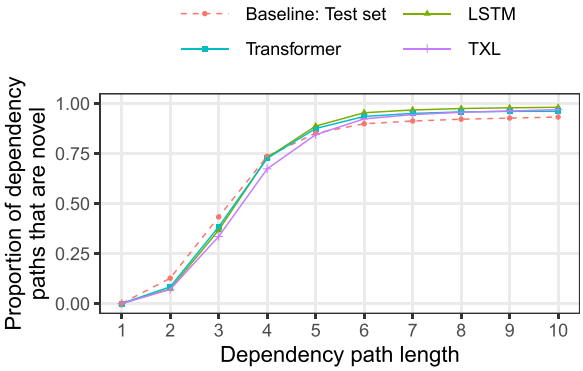
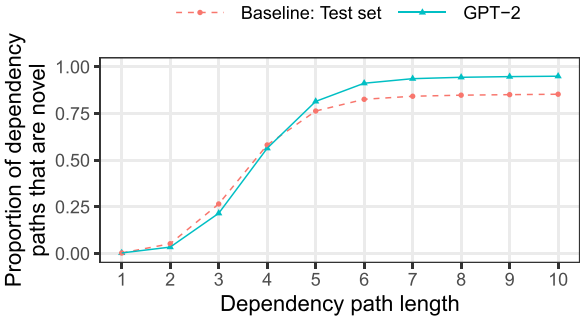


Figure 4: Example dependency tree.



(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.

See the supplement (Section M) for specific examples of syntactic generalization (e.g., nouns that were generated as direct objects but 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

	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.

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 supplement (Section N) for a detailed taxonomy of GPT-2's novel words. Here in the main paper, we discuss 4 targeted questions about these novel words. Throughout this section, any word in boldface is novel.

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 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 its abbreviation, 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 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 a question (11). In (12), it correctly inflects 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.

(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 relevant cases; the only error is the one in (9). See Haley (2020) for similar observations but with BERT (Devlin et al., 2019) instead of GPT-2.

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** were lucky to have her as part of their lives
b. so many **Brazilianisms**

Across these categories, GPT-2 makes no errors, but the sample size is small (we found 1 possessive example, 7 examples with coreferential pronouns, and 4 with number-sensitive determiners).

	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.

Incrementing/Ordering: Another type of inter-word relation that GPT-2 appears to have learned is incrementing/ordering, with examples in the supplement (Section O.8). 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 *multiplyz* to *multiplyz*. In a final example, the prompt ends with an alphabetical list of 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

Therefore, GPT-2 may 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

	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.

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’s meaning:

(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 (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 supplement (Section O.16) 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, GPT-2 sometimes generalized too freely, such as generating the word *752th* (Section 7.1). Therefore, it may not be enough to simply encourage models to be systematic, because language is not completely systematic. Instead, we need models that recognize both rules and exceptions (O'Donnell, 2015; Yang, 2016).

Our analyses focused on compositional generalization as it applies to linguistic form (specifically, morphology and syntax). An important future direction 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. 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. However, for all 92 basic irregular verbs in English, the incorrect regular form appears in GPT-2's training set; details are in the supplement (Section P), along with results for another category often assumed to be novel, namely, nonsense words such as *wug* (Berko, 1958). Thus, when we use models trained on such large-scale datasets, it is not safe to assume that something is absent from the training set; we must actually 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 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.

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. https://doi.org/10.1016/S0010-0277(03)00146-X, PubMed: 14599751

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. https://doi.org/10.1145/3442188.3445922

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. https://doi.org/10.18653/v1/2020.acl-main.463

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

[6] https://pixabay.com/illustrations/crow-raven-black-dark-bird-ink-4779560/ (last accessed March 18, 2023).

https://doi.org/10.1080/00437956.1958.11659661

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:2006.14799v2*.

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:2107.03374v2*. https://doi.org/10.48550/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. https://doi.org/10.18653/v1/P19-1285

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

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.

https://doi.org/10.18653/v1/2022.acl-long.501

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. https://doi.org/10.18653/v1/N16-1024

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: 2207.14251v1*. https://doi.org/10.48550/arXiv.2207.14251

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. https://doi.org/10.1613/jair.5477

Robert F. Hadley. 1994. Systematicity in connectionist language learning. *Mind & Language*, 9(3):247–272. https://doi.org/10.1111/j.1468-0017.1994.tb00225.x

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. https://doi.org/10.18653/v1/2020.blackboxnlp-1.31

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:2205.12600v1*.

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. https://doi.org/10.18653/v1/N19-1169

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

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

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. https://doi.org/10.1613/jair.1.11674

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:2001.08361v1*. https://doi.org/10.48550/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:1909.05858v2*. https://doi.org/10.48550/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. https://doi.org/10.18653/v1/2020.emnlp-main.731

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. https://doi.org/10.18653/v1/P18-1249

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. https://doi.org/10.18653/v1/2020.emnlp-main.750

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. https://doi.org/10.1016/S0022-5371(77)80021-2

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:2203.07618v2*. https://doi.org/10.1145/3543507.3583199

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. https://doi.org/10.18653/v1/2022.acl-long.577

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. https://doi.org/10.18653/v1/2021.acl-long.368

Gary Marcus. 2020. The next decade in AI: Four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177v3*. https://doi.org/10.48550/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*, i–178. https://doi.org/10.2307/1166115

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. https://doi.org/10.1162/tacl_a_00304

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. https://doi.org/10.18653/v1/2021.acl-long.414

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. https://doi.org/10.1111/j.1755-2567.1970.tb00434.x

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. https://doi.org/10.7551/mitpress/9780262028844.001.0001

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. https://doi.org/10.18653/v1/2020.conll-1.13

Sandeep Prasada and Steven Pinker. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56. https://doi.org/10.1080/01690969308406948

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. https://doi.org/10.18653/v1/2020.emnlp-main.349

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. https://doi.org/10.18653/v1/N19-1356

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*. https://doi.org/10.48550/arXiv.1910.06611

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. https://doi.org/10.18653/v1/P16-1162

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. https://doi.org/10.18653/v1/2021.emnlp-main.72

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. https://doi.org/10.18653/v1/2022.findings-acl.314

Charles Yang. 2016. *The Price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. MIT Press. https://doi.org/10.7551/mitpress/9780262035323.001.0001

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. https://doi.org/10.1145/3446776

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained Transformer language models. *arXiv preprint arXiv:2205.01068v4*. https://doi.org/10.48550/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. https://doi.org/10.18653/v1/2021.acl-long.90

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. https://doi.org/10.1145/3209978.3210080

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