It’s my Job to be Repetitive! My Job! My Job! – Linking Repetitions to In-Context Learning in Language Models

Anonymous ACL submission

Abstract

Recent studies have shown that large language models can display surprising accuracy at learning tasks from few examples presented in the input context, which goes under the name of in-context learning. Other studies have shown that language models can sometimes display the undesirable behavior of falling back into loops in which an utterance is repeated infinitely often. Here, we observe that the model’s capacity to produce repetitions goes well beyond frequent or well-formed utterances, and generalizes to repeating *completely arbitrary* sequences of tokens. Construing this as a simple form of in-context learning, we hypothesize that these two phenomena are linked through shared processing steps. With controlled experiments, we show that impairing the network from producing repetitions severely affects in-context learning, without reducing its overall predictive performance, thus supporting the proposed hypothesis.

1 Introduction

Large language models are becoming increasingly predominant in NLP for solving a wide range of tasks. Those models are often used in their capacity to generate natural language to produce an answer to a task description given as context (Brown et al., 2020; Raffel et al., 2020; Radford et al., 2019; Petroni et al., 2019). Despite the promise of this paradigm, some studies have raised concerns on the generative capabilities of these models, noting that they can shift from “generating” text to “degenerating” (Holtzman et al., 2020). One such case of degeneration is producing *repetitions*, where the model falls into repeating indefinitely the same sequence of tokens (or very similar variations). Thus, different studies have been conducted aimed at correcting this “bug” in the language models (Welleck et al., 2019; Fu et al., 2020; Lin et al., 2021; Liu et al., 2021b).

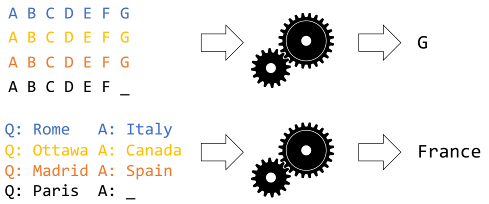

Figure 1: We hypothesize that repetitions and in-context learning are related mechanisms.

While previous work has focused on repetitions of natural text, here we show that the model’s capacity to predict copies of a sequence goes well beyond natural well-formed text, generalizing to *completely arbitrary* sequences of tokens that dramatically differ from the distribution the model was trained on. This observation hints at a *general* capacity of the model to detect and *copy* a repeated pattern in the input, independently of its content.

Furthermore, we hypothesize that there is a link between this emergent property, and the surprising *in-context learning* (ICL) capacity of large language models (Brown et al., 2020; Raffel et al., 2020). ICL means that providing the language model with a few examples of a desired task in its context, and a query on a new instance of that task, it can generate the correct answer with reasonable accuracy. We hypothesize that this “feature” is linked to producing repetitions as follows. Given a sequence of examples, we expect that a model capable of ICL should first segment the input into individual examples, and second, extract a general relation between them to answer a query. Processing a repeated sequence in the input might involve a similar pipeline with the only difference that there is no variance in the examples, and thus the model reduces to producing the same examples verbatim.

We test the link between these two capacities by fine-tuning a decoder-only Transformer model to avoid producing repetitions and show that it also impairs its capacity to do ICL.

Until now, ICL capacity was attributed to the sheer size of recent models. Here, we introduce a benchmark based on the word analogy task by Mikolov et al. (2013) and show that all variants of GPT-2 (Radford et al., 2019) are capable of ICL, albeit for simpler tasks. Equipped with this more versatile instance of ICL, we then impair a GPT-2 medium model from producing repetitions using Unlikelihood Training (Welleck et al., 2019). The resulting model has disastrous performance on the ICL capacity (dropping from 0.8 to almost 0) without deteriorating the ability to model human language as measured by perplexity. In a nutshell, we: 1) show that modern language-models are able, out-of-the-shelf, of copying any arbitrary sequence (Sect. 3). 2) re-purpose an analogy data-set and evaluating the accuracy of much smaller language-models than previously considered on this simplified ICL task (Sect. 4). 3) provide evidence supporting our hypothesis that the two phenomena are linked, by showing that impairing the production of repetitions affect significantly ICL capabilities (Sect. 5).

2 Related work

2.1 Repetitions in language models

Drawbacks of modern language generation models include the generation of very fluent, but incorrect statements. Those are captured by the ill-defined concept of *hallucination*, and several methods have been proposed to detect them, often involving examining the self-attention (Zhou et al., 2020; Berard et al., 2019). Raunak et al. (2021) proposes a classification of such hallucinations for translation. Here we are particularly interested in their “oscillatory hallucinations”, defined as “an inadequate translation that contains repeating ngrams”.

The specific problem of repeated strings is studied by a number of papers. Liu et al. (2021c) proposes a copying penalty whose magnitude depend on the confidence of the model in the current prediction. This solution of down-weighting tokens that appear in the context is a standard one, and produces text preferred by humans (Foster and White, 2007). Probably most famously, this has been used in Unlikelihood Training (Welleck et al., 2019) to further train a language-generation model. Similarly, Lin et al. (2021) proposes to *increase* the likelihood of novel tokens. Fu et al. (2020) analyzes the problem in a Markovian set-up, which simplifies analysis as the transitions are independent of the context. Their analysis leads them to consider words whose left context (preceding word) is frequent and often succeeded by that word. By de-tokenizing those pairs at the pre-processing step the resulted text is indeed less repetitive, although perplexity and BLEU scores also degrade slightly.

2.2 In-context learning

In-context learning (ICL) refers to the hypothesis introduced by Brown et al. (2020) that language models can learn to perform tasks from natural language instructions and/or examples given as part of their prompt. Some studies have started to inquire which factors affect most the performance of these models at learning some tasks (Liu et al., 2021a; Zhao et al., 2021). Others have looked at synthetic tasks to try to shed light on the learning capabilities of GPT-3 (Rong, 2021). Nonetheless, the science on this phenomenon is still young, and more work is needed to improve our understanding of ICL.

3 Generating arbitrary repetitions

We first verify if modern pre-trained language models are capable of copying any *arbitrary* sequence. This can be seen as testing ICL for what is arguably the simplest of tasks: given the same prefix, predict always the same token. Having such skill would depend on the model developing a general-purpose copying mechanism that is independent of the kind of input being copied. In other words, we test whether the model can copy sequences that are completely out-of-distribution in the training data. Note that it is not obvious that a pre-trained model would be able to do so. In the past, language generation models had to enforce copying from the source document explicitly (Gu et al., 2016), for instance through the use of a pointer-network (See et al., 2017; Miao and Blunsom, 2016).

To test this for arbitrary sequences, we uniformly sample 10 tokens from the 50 257 different tokens in its vocabulary to form an arbitrary sequence. Then, we prompt the model with 1 to 5 copies of this sequence, before prompting it with its 9 first tokens and asking it to predict the 10-th one. Results are displayed in Fig. 2. As can be seen, all different variants of GPT-2 obtain almost perfect prediction with as little as 2 copies in the prompt.

4 In-context learning in smaller models

Brown et al. (2020) shows famously that ICL arises

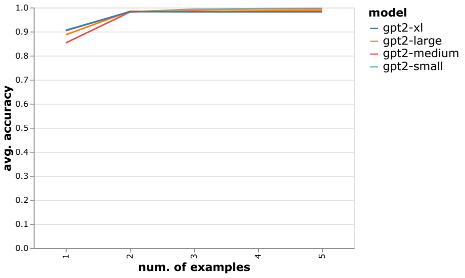

Figure 2: Base accuracy at the repetitions task.

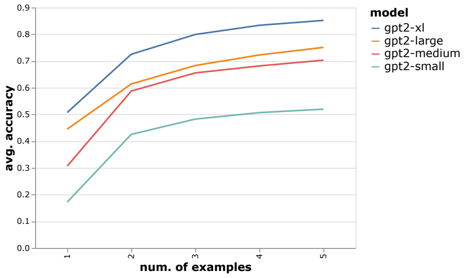

Figure 3: Base accuracy on the analogy task.

with very large models. In their experiments, models of 1.3B or 13B parameters obtained only single-digit accuracy when provided one example, while a 175B parameter models obtains around 45%. Thus, the capability of ICL has been hypothesized to belong to the realm of very large language models such as GPT-3. Here, we observe that this capacity is already present in smaller language models for less complex tasks, allowing to experiment with publicly available models, such as GPT-2.[1]

To find ICL in smaller models, we adapt the word analogy task introduced by Mikolov et al. (2013) to a few-shot learning scenario. This task contains different kinds of word analogy trials, such as `Athens` is to `Greece` as `Baghdad` is to `?`. There are 14 categories of analogies, some semantic/knowledge-based as in the previous example, and some grammatical such as converting an adjective to its comparative or superlative counterpart. We format tasks as 1-shot learning prompts, as follows: "`Q: Athens A: Greece ; Q: Baghdad A:`", which we feed to the language model, and extract the first non-empty word that is greedily generated, and then compare it with the target answer (`Iraq` in the example). To generate k-shot learning examples, we randomly sample more pairs of analogous words in the same relation and concatenate them to the prompt using the semicolon separator. We evaluate all language models in the GPT-2 family, observing in Fig. 3 that they all have relatively high performance on this task, exhibiting the expected trend of higher accuracy with larger models or more examples in the input.

5 Impairing repetitions

Our hypothesis predicts that ICL builds on skills that are needed to compute repetitions. If a model is discouraged from producing repetitions, then we predict that this should also affect ICL. To test this prediction, we rely on Unlikelihood Training (UL) (Welleck et al., 2019) which impairs repetitions by fine-tuning an auto-regressive language model using two objectives on samples from a training corpus (Wikipedia). The first objective being minimized aims at reducing the probability of n-grams that appear in the previous context. The second one is a standard language modelling objective on the training corpus. The combination is done by alternating randomly between the two objectives. We obtain multiple models by varying the τ parameter that controls the proportion of gradient steps performed on the unlikelihood objective with respect to the standard language modelling one (details in Appendix A).

If ICL is affected by UL, this would raise the question if this follows from the unlikelihood objective or from disrupting the model's capabilities. Thus, we contrast the above-described **UL (wiki)** condition with two other control conditions. In **LM (wiki)**, we only fine-tune the model on the language model objective, which disrupts the model by overfitting to the Wikipedia corpus. By increasing the number of fine-tuning epochs, we obtain a range of language models with diverse levels of quality that are comparable to the models fine-tuned with unlikelood training. Second, in **UL (gen)**, we perform standard UL but sampling the training sequences from the model itself rather than from a given corpus. This approach, inspired by GDC (Khalifa et al., 2020), aims at reducing drift from the original model by minimizing the cross-entropy loss on sequences sampled from it. While UL does not provide the strong theoretical guarantees of GDC, we hypothesize that using self-generated samples as a training corpus can result in higher quality models in general. In detail, we generate 200k sequences of 500 tokens each, which in total has a comparable size to the Wikipedia corpus used by UL (wiki). To

[1] At the time of writing, all available decoder-only models are still of the same order of magnitude than GPT-2. The much larger T5 (Raffel et al., 2020), with 11B parameter, is an encoder-decoder model.

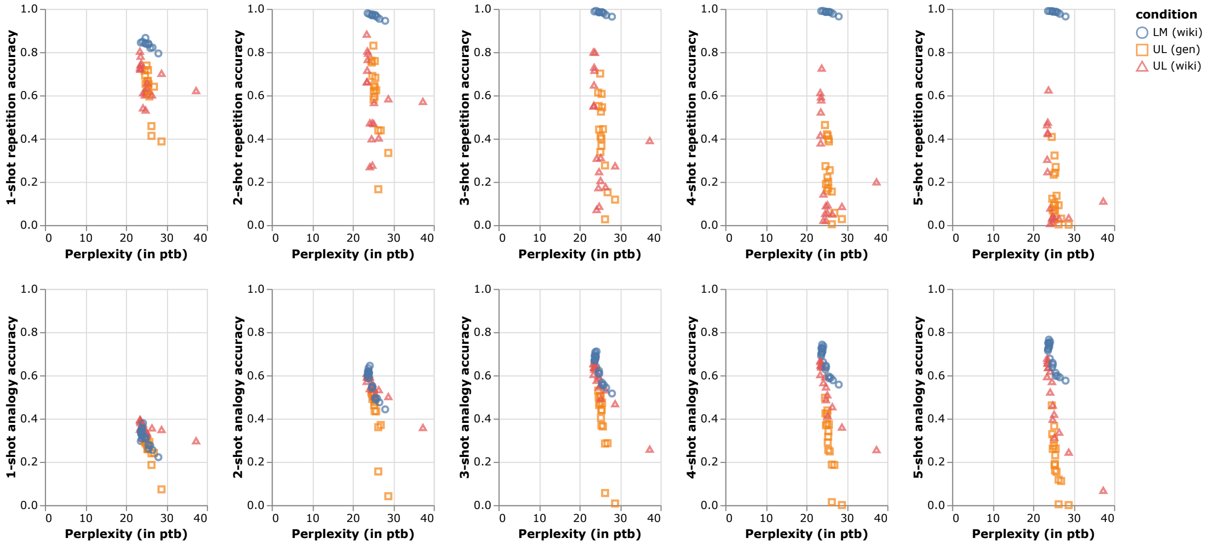

Figure 4: (**top**) k-shot acc. at predicting the last symbol of a 10-token sequence from k copies followed by the first 9 tokens. (**bottom**) k-shot acc. at ICL on analogies when presented with k examples followed by a query.

measure the quality of the different models, we use perplexity on a held-out dataset, namely the Penn Treebank Corpus (Marcus et al., 1993, PTB).[2]

6 Results & Discussion

The results of our main experiments are summarized in Fig. 4, showing repetition and analogy accuracy. From left to right, each panel corresponds to adding one more example to the context, starting from only one example in the leftmost one. First, we note that repetition accuracy drops significantly (from 0.8 to 0.2 and lower) for UL (wiki)—something expected, as this is exactly the purpose of UL—, with only minor but consistent loss in model quality. The models with worse quality are those with the highest values of the τ parameter, although, surprisingly, they are not the ones with the lowest repetition accuracy. The results on analogy are more relevant to our study. We begin by comparing the UL (wiki) with the LM (wiki) conditions. When only one example is provided, UL models have *better* ICL performance than LM when comparing two models with equivalent perplexity. However, they have a similar behaviour when 2 or 3 examples are provided and finally the trend reverts when more examples are provided as context. In the extreme case we tested, performance of UL drops from 0.6799 to 0.3130 for 5-shot analogy tasks without significantly impacting perplexity. Thus, the more examples are provided, the more the models in the UL (wiki) condition are affected with respect to the control LM (wiki) one.

The contrast between the UL/LM conditions is exacerbated when comparing UL (gen) to LM (wiki): it seems ICL can be made arbitrarily bad with only minor impact on perplexity. Moreover, while the UL condition already shows a stronger impact than LM for 1-shot learning, the difference grows larger with more examples and reaches 0.0039 analogy accuracy for a UL model with the same perplexity than a LM model that obtains 0.6136. Interestingly, the perplexity of the UL models are very little impacted (the dots are almost vertically aligned).

7 Conclusions

We have shown that LMs are more capable at producing repetitions than previously acknowledged, and tested the hypothesis that ICL capacity of modern large language models builds on the same ability to detect and reproduce repeated patterns. After impairing the generation of repetitions using UL, we observe that ICL degrades and perform control experiments to measure the magnitude of the degradation. A potential shortcoming of this analysis is that while UL impairs *generation* of repetitions, it does not necessarily (although it might) impair their *detection*. For future work, we envision a replication of these effects through other methods to impair repetitions, using larger models and more complex ICL tasks.

[2] `ptb_text_only` of HuggingFace's `datasets` (Lhoest et al., 2021).

References

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *arXiv:2005.14165 [cs]*. ArXiv: 2005.14165.

Mary Ellen Foster and Michael White. 2007. Avoiding repetition in generated text. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 33–40.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2020. A theoretical analysis of the repetition problem in text generation. *arXiv preprint arXiv:2012.14660*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. *arXiv:1904.09751 [cs]*. ArXiv: 1904.09751.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*.

Xiang Lin, Simeng Han, and Shafiq Joty. 2021. Straight to the gradient: Learning to use novel tokens for neural text generation. *arXiv preprint arXiv:2106.07207*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. What Makes Good In-Context Examples for GPT-3? *arXiv:2101.06804 [cs]*. ArXiv: 2101.06804.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT Understands, Too. *arXiv:2103.10385 [cs]*. ArXiv: 2103.10385.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, Shuming Shi, and Zhaopeng Tu. 2021c. On the copying behaviors of pre-training for neural machine translation. *arXiv preprint arXiv:2107.08212*.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328, Austin, Texas. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases? *arXiv:1909.01066 [cs]*. ArXiv: 1909.01066.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*. ArXiv: 1910.10683.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*.

Frieda Rong. 2021. Extrapolating to Unnatural Language Processing with GPT-3's In-context Learning: The Good, the Bad, and the Mysterious.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-Shot Performance of Language Models. *arXiv:2102.09690 [cs]*. ArXiv: 2102.09690.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

A Experimental details

For Unlikelihood Learning (UL), we varied the threshold parameter τ[3] across the following values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.83, 0.85, 0.87, 0.9, 0.92, 0.95, 0.97, 0.99.

For over-fitting on the dataset, we continue training of GPT-2 on the same dataset (either wiki or the generated text) for more epochs. We report results on 1, 3, 4, 5, 6, 7, 8, 9, 10 and 15 epochs.

B Full sequence arbitrary repetitions

On the above experiments about repeated sequences, we focused on the setting of predicting the last token of a repeated 10-token-long sequence given all the previous 9 ones because of the analogy this bears with ICL (see Fig. 1). Nonetheless, it is interesting to measure the models' accuracy at predicting *all* the 10 tokens in the sequence. Fig. 5 shows exactly that. Interestingly, smaller models have higher accuracy than bigger ones, although they all have very high accuracy with 2 or 3 examples.

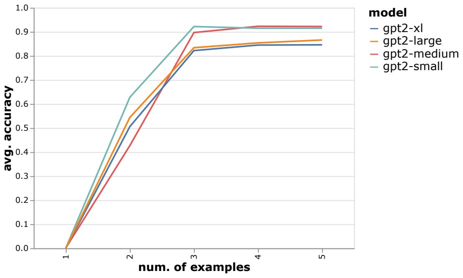

Figure 5: Base accuracy at predicting greedily all 10 tokens of an arbitrary sequence as a function of the number of copies in the input context.

[3] `sequence-tune-rate` in the codebase we used, at `https://github.com/facebookresearch/unlikelihood_training`.