COLLAPSE OF SELF-TRAINED LANGUAGE MODELS

David Herel
FEE, Czech Technical University in Prague
hereldav@fel.cvut.cz

Tomas Mikolov
CIIRC, Czech Technical University in Prague
tmikolov@gmail.com

ABSTRACT

In various fields of knowledge creation, including science, new ideas often build on pre-existing information. In this work, we explore this concept within the context of language models. Specifically, we explore the potential of self-training models on their own outputs, akin to how humans learn and build on their previous thoughts and actions. While this approach is intuitively appealing, our research reveals its practical limitations. We find that extended self-training of the GPT-2 model leads to a significant degradation in performance, resulting in repetitive and collapsed token output.

1 INTRODUCTION & RELATED WORK

From the viewpoint of artificial intelligence, it could be important for a model to be able to self-evolve and learn from its own actions. Although neural network models partially address this problem by storing information in the hidden layer and utilizing the attention mechanism, for example, the vanishing gradient problem limits their effectiveness (Basodi et al., 2020). Dynamic models (Jelinek et al., 1991) have been suggested as a solution where models are trained on test data to utilize a form of cache. Dynamic evaluation for neural networks models was proposed by Mikolov et al. (2011; 2010); Krause et al. (2017; 2019), where the neural network parameters are updated using the standard training mechanism during the processing of the test data.

Our work explores the concept of self-training a model on its own output that is generated through sampling (Deoras et al., 2011). However, we provide empirical evidence that this self-training approach can lead to model collapse, where the generated outputs become severely biased and repetitive. This trend has also been explored in Shumailov et al. (2023). Our findings indicate limitations in the current model architecture regarding self-evolution. For future research, it may be beneficial to explore entirely new models that can more effectively accommodate this aspect.

2 METHOD: SELF-TRAINING OF LLM

In our settings, self-training adjusts model parameters θ_g to better model local sequence distribution, $P_l(x)$, which is generated from a model. The initial adapted parameters $\theta 0_l$ are set to θ_g, computing the probability of the first sequence, $P(s_1|\theta 0_l)$. This results in a cross-entropy loss $\mathcal{L}(s_1)$, with gradient $\nabla\mathcal{L}(s_1)$, updating the model to adapted parameters $\theta 1_l$. We then let the model generate sequence again and evaluate $P(s_2|\theta 1_l)$, repeating this for each s_i. Each update approximates the current local distribution $P_l(x)$.

3 EXPERIMENT: EMPIRICAL ANALYSIS OF GPT-2 MODEL

For our experiments, we utilized the pre-trained GPT-2 (Radford et al., 2019) model that is available as open-source. We allowed the model to train on its own generated output while tracking its performance after each update on a valid set of Wikitext-2 (Merity et al., 2016). We set a stopping criterion for the model, either when it collapsed to repeating sequences or when it reached 1000 iterations. The hyperparameters used in our experiments, as well as the associated codebase, are available in A.4.

Our observations show that the validation loss increases with each iteration and is significantly influenced by the learning rate. When the learning rate is higher, the model collapses faster and

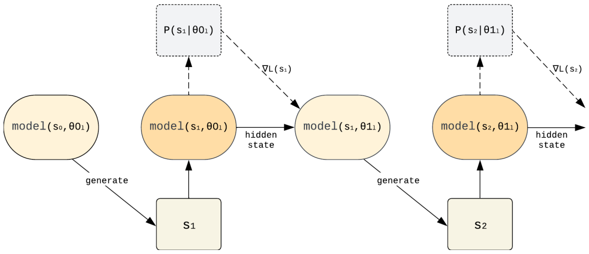

Figure 1: Schema of the self-training. The model generates a sequence s_1, computes its probability $P(s_1|\theta 0_l)$, which is then used to determine the cross-entropy loss with gradient $\nabla\mathcal{L}(s_1)$ to update the next state of the model with the adapted parameters.

produces repetitive tokens quickly. This phenomenon is exemplified by a significant decrease in loss on generated (train) data, almost reaching 0 loss. The progression of output generation and the noticeable degradation towards model collapse can be observed in A.1. Further details on the impact of model size on the rate of collapse are provided in the appendix A.2.

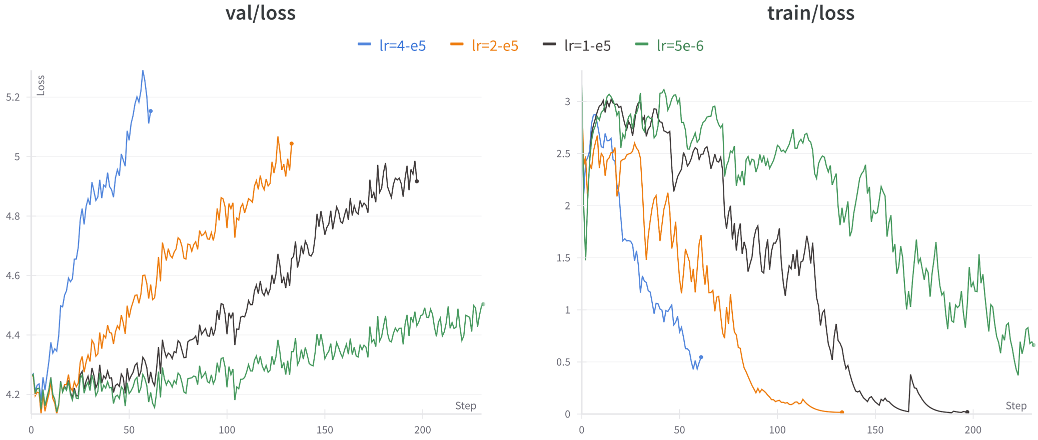

Figure 2: Impact of learning rate on self-training GPT-2 (Radford et al., 2019) language model on valid and train sets. As the learning rate increases, the model's performance deteriorates, leading to a higher loss on the valid set. On the train set, the model collapses and converges into a generation of repetitive tokens, resulting in almost zero loss on generated data. The y-axis represents the loss, and the x-axis displays the number of model steps.

4 Discussion

In this study, we investigated the potential of self-training language models on their own outputs. Our results demonstrate that extended self-training of the GPT-2 model leads to significant performance degradation, with models collapsing into repetitive sequences consistently. We also observe that the learning rate has a notable impact on the speed of this collapse.

With the extensive use of language models in various text generation applications, it can be expected that in the future there will be an increasing amount of text on the web with artificial origin. As the training data for language models are typically scraped from the web, the collapsing problem we describe in this paper can become a serious issue, as language models will be in the future largely trained on data that were generated from other language models.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

References

Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, 3(3):196–207, 2020. doi: 10.26599/BDMA.2020.9020004.

Anoop Deoras, Tomas Mikolov, Stefan Kombrink, Martin Karafiat, and Sanjeev Khudanpur. Variational approximation of long-span language models for lvcsr. pp. 5532–5535, 05 2011.

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. A dynamic language model for speech recognition. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991. URL https://aclanthology.org/H91-1057.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. Dynamic evaluation of neural sequence models. *CoRR*, abs/1709.07432, 2017. URL http://arxiv.org/abs/1709.07432.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. Dynamic evaluation of transformer language models. *CoRR*, abs/1904.08378, 2019. URL http://arxiv.org/abs/1904.08378.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016. URL http://arxiv.org/abs/1609.07843.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. volume 2, pp. 1045–1048, 01 2010.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, J.H. Cernocky, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. pp. 5528 – 5531, 06 2011. doi: 10.1109/ICASSP.2011.5947611.

Tomáš Mikolov. *STATISTICAL LANGUAGE MODELS BASED ON NEURAL NETWORKS*. Ph.d. thesis, Brno University of Technology, Faculty of Information Technology, 2012. URL https://www.fit.vut.cz/study/phd-thesis/283/.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, 2023.

A Appendix

A.1 Example of model collapse

This section contains examples of sequences generated by the GPT-2 model with a learning rate of 2-e5. Starting from the first iteration, we provide a list of what the model generated at 50 iterations, 100 iterations, and the final iteration when the model collapsed into repetitive repetition, which is documented in Table 1. These examples demonstrate the limitations of the current model architecture regarding self-evolution and highlight the challenges associated with self-training language models on their own outputs. By presenting these examples, we hope to provide valuable insights into the potential and challenges of self-training language models and to contribute to ongoing efforts to improve the performance and effectiveness of these models.

Iteration	Example
0	*And these days, we can understand why so many entrepreneurs struggle with it. I’ve studied that fact thoroughly — along with many others. It is the reason we have the most value — not only at our company, but in other businesses around the world.* *We’ve got our share of challenges, too. Most of those we face are insurmountable. We’re very lucky, even if our success holds a 1% return.* *In America, 20 companies provide less than 1% return, but this has dropped 3-4% in 18 months. While the industry has increased, its margins are shrinking and its investment opportunities are curtailed.* *Take RFPs for some of the leading firms . Many startups fail because of weak leadership; some are successful, but still fail.* *High tech companies need competition. Take RFP for Google . The main competitor to RFP A is Data Corp .* *Sourcing remains a major choice for businesses, and so need*
50	*The way to look at this is that the number of U.S. citizens in the ”cannabis community” is not that large. And it can be if these are local residents, local or national.* *But at the same time the number of U.S. citizens in the cannabis community is not that large.* *The U.S. grows a lot of (cannabis)* *So we should be pretty sure that there is a lot of support.* *There’s still a long way to go.* *But, I want to let you in on the good news.* *The federal law that will likely be up in the air in the coming months, before the House and Senate both vote on it, is the very same law that has already been introduced.* *The Affordable Care Act, those are the two main provisions.* *The law that I had to tell you, it’s only so I can say that it’s the*
100	*- In the 4.1 final build, we dropped the ’the_no_go’ flag.* *- In the 4.9 final build, we stuck the ’the_no_go’ flag.* *- In the 4.9 final build, we stuck the ’the_no_go’ flag.* *- In the 4.9 final build, we will be in the ”final” release.* *- The ”final” release is the ”right time”* *- The ”final” release is the ”right time”* *- The ”final” release has the ”right time”* *- The ”final” release has the ”right time”* *- The ”final” release is the ”right time”* *- The ”final” release is the ”right time”* *- The ”final” release is the ”right”*
133 (last)	- - - - - - - - - - -

Table 1: Examples at key iterations — 0, 50, 100, and the final iteration documenting the progression until the model succumbed to repetitive patterns.

A.2 Impact of parameter sizes

To investigate the relationship between the number of parameters in a model and its stability, we conducted a series of experiments using GPT-2 architectures with varying sizes. Specifically, we compared models with parameter counts ranging from 100 million to 1.5 billion. As depicted in Figure 3, our findings highlight a notable trend: larger models tend to exhibit more rapid onset of model collapse.

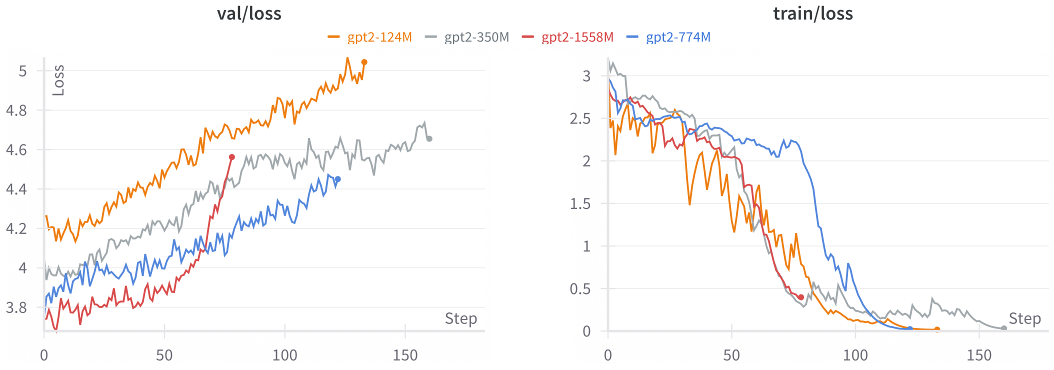

Figure 3: Correlation between model size and the onset of collapse in GPT-2 architectures.

A.3 Different evaluation dataset

Beyond the standard Wikitext-2 benchmark, we expanded our evaluation to include the Penn Treebank dataset (Mikolov, 2012), a prominent resource in language modeling. Our objective was to ascertain whether the increase in validation loss, as observed with the Wikitext-2 dataset, is consistent across different datasets. The comparative results presented in Figure 4 indicate that the Penn Treebank dataset yields a similar pattern, suggesting that our observations are not exclusive to a single dataset but may reflect a more general phenomenon.

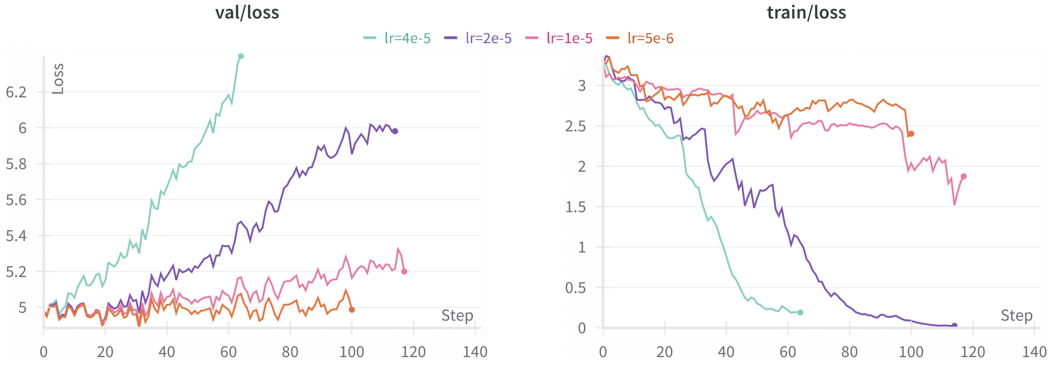

Figure 4: Comparative analysis of learning rate impact on self-trained GPT-2 model performance, evaluated on it's output (train loss) and validation subset of the Penn Treebank dataset.

A.4 EXPERIMENT HYPER-PARAMETERS

In this section, we present a list of the hyper-parameters utilized in our model experiments. Additionally, to promote transparency and reproducibility, we have created a code-base that replicates our results: collapse-lm

Hyperparameters	Value
temperature	0.8
top_k	500
max_new_tokens	200
max_iters	100
grad_clip	1
batch_size	1
block_size	100
n_layer	12
n_head	12
n_embd	768
dropout	0
bias	False
beta1	0.9
beta2	0.95
prompt	””
batch_size	1

Table 2: Hyper-parameters for GPT-2 model experiments.