Predicting Task Forgetting in Large Language Models

Anat Kleiman [1] **Jonathan Frankle** [1] **Sham Kakade** [1] **Mansheej Paul** [2]

Abstract

In this paper, we offer a comprehensive evaluation of forgetting in large language models (LLMs) during sequential learning of finetuning tasks in a pretrained model. We empirically track the degradation of performance across diverse tasks and find that the validation perplexity can be predicted using a linear function, regardless of the specific task, model architecture, or task order. This knowledge sheds light on the dynamics of knowledge acquisition and retention, offering practical implications for managing and mitigating task forgetting in LLM-based systems.

1. Introduction

Common convention around sequential learning assumes that as a model is fine-tuned on new tasks, its performance on previously learned tasks degrades (Scialom et al., 2022). In some cases, this degradation can manifest as catastrophic forgetting, a rapid decline in performance, often measured by the accuracy of past tasks ((Lesort et al., 2019); (Delange et al., 2021); (Belouadah et al., 2021); (Hadsell et al., 2020)). Consequently, it has become increasingly accepted that the learning of new tasks may potentially replace previously acquired knowledge at an unpredictably high rate.

In this paper, we provide an extensive evaluation of task performance on large language models (LLMs) as they learn new tasks. Our objective is to precisely track the degradation of performance across a diverse range of tasks and model sizes. By doing so, we aim to shed light on the nature of forgetting within LLMs and offer empirical evidence about the rate of task forgetting.

Through our comprehensive analysis, we uncover a consistent pattern of performance degradation across various language tasks. We demonstrate that the degradation of the validation perplexity can be predicted, as it predominantly adheres to a linear function regardless of the specific task, model architecture, or the order in which tasks are presented. This finding has profound implications, as it suggests that the rate of degradation can be estimated with reasonable accuracy, thus providing valuable insights into the dynamics of knowledge acquisition and retention in LLMs. Furthermore, by investigating task performance degradation in LLMs, our research also offers practical implications for designing and optimizing LLM-based systems. Knowledge of the predictable degradation patterns allows for better management and mitigation of task forgetting, ultimately enhancing the robustness and long-term stability of LLMs in real-world applications.

In the subsequent sections of this paper, we present our experimental methodology, including the task datasets and models we use, and provide a detailed analysis of performance degradation we observe across multiple scenarios.

2. Related Works

There exist many works that analyze or solve an aspect of forgetting in continual and sequential learning. Given scenarios with sequences of disjoint training tasks, (Rebuffi et al., 2017) show that training classifiers from class-incremental data streams (like with stochastic gradient descent optimization) causes catastrophic forgetting. However, (Lesort et al., 2023) show how using continual learning on long sequences of tasks allows learning with stochastic gradient descent to result in knowledge retention and accumulation rather than pure forgetting. Thereby, challenging previous assumptions about catastrophic forgetting in computer vision scenarios. (Scialom et al., 2022) likewise explore using continual learning in LLMs to prevent catastrophic forgetting with similar successful results.

3. Methodology

The following section describes our implementation of finetuning LLMs and measuring their evaluation loss on various tasks, including information about the task datasets, our hyperparameters, and model architectures.

[1]Harvard University, Cambridge, MA [2]Stanford University, Stanford, CA. Correspondence to: Anat Kleiman <anatkleiman@g.harvard.edu>, Jonathan Frankle <jfrankle@g.harvard.edu>, Sham Kakade <sham@seas.harvard.edu>, Mansheej Paul <mansheej@stanford.edu>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

3.1. Methods

In the following section we describe our training scheme. We refer to the following definitions throughout this project:
Sequential Learning: Models are exposed and finetuned to one task at a time (Aljundi et al., 2019). After training of that task is over, we expose it to a new task while continuing to measure evaluation loss on previously learned tasks to measure rate of forgetting.
Multitask Learning: Models are exposed to data from all given tasks at a time (Caruana, 1997). In turn, models learn to optimize performance on all tasks at the same time resulting in a multitask model.
Forgetting: We measure forgetting in terms of loss on a given set. Specifically, an increase in loss on the same evaluation set indicates forgetting of that task for a given model.

3.2. Data

We use 6 language generation tasks to train and evaluate our models on. Specifically, these are Wikiauto (Jiang et al., 2020), Eli5 (Fan et al., 2019), Gigaword (Scialom et al., 2022), Covidfact (Scialom et al., 2022), COVID-QA (Möller et al., 2020), and Twitter Top 20 (Bin Tareaf, 2017). Additional information about each task can be found in Appendix Section 5. Given each task's unique source and output, we are able to test the ability of our models to satisfy: length and style requirements, as well as topical knowledge.

3.3. Hyperparameters

We predominantly use the T0 validated hyperparameters described in (Scialom et al., 2022) to set up and finetune our models. The details of these can be found in Appendix Section 5. One key difference we incorporate is to use a constant learning rate instead of a learning rate schedule to confirm that jumps in evaluation loss between tasks are not due to a reset in learning rate.

3.4. Model

We finetune two language models, both in the T0 family (Sanh et al., 2021): T0_3b (3B parameters) and T0_pp (11B parameters). These models are based off T5 (Raffel et al., 2020), which is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised, supervised, and self-supervised tasks and for which each task is converted into a text-to-text format. T5 is trained using teacher forcing, where for training, there is an input sequence and a corresponding target sequence. As such, T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task. We believe this makes it an especially promising model to finetune different language tasks on.

3.5. Training

We train the following models and compare and contrast evaluation loss across all of our tasks.

3.5.1. Baselines

We incorporate 2 types of baselines to compare our sequential learner models to:
No Exposure: For each task we analyze, we train a model using multitask learning on all tasks except that task. This serves as our "no exposure" baseline and acts as a substitute for pure forgetting of that task.
Multitask Model: We train a model using multitask learning on all of the task data to act as a model that fully remembers how to perform every task.
By incorporating a multitask model as our ideal learner, we can compare a sequential learner with a model that remembers how to perform a task as well as possible. Furthermore, by incorporating a model with no exposure to a task, we can introduce a proxy for fully forgetting a particular task.

3.5.2. Orderings

We train sequential learner models using different orders of the tasks in order to compare how learning certain tasks first affects learning/forgetting of future tasks, as well as to control for such differences to determine the rate of forgetting for each task (See Appendix Section 5).

4. Results

4.1. Rate of Forgetting

We train our sequential learner on the tasks described in 3.2 by introducing each task separately. As our model trains on a new task, we continue to evaluate loss on previous tasks to measure changes in performance and potential forgetting. Across all tasks, forgetting occurs with loss increasing as the number of tasks increases (See Figure 1 and Appendix Section 5 for all other tasks). Despite this, the rate of forgetting is relatively low. Instead of large sporadic instances of complete forgetting, we find that the model does slightly worse on a given task over time as it trains on a new task. Specifically, for Figure 1, the slope is approximately 0.00071.
This holds when we compare our baselines to the sequential learner. As shown in Figure 1, as training continues, the sequential learner performs worse than the multitask model with exposure to the given task. However, the sequential learner continues to retain information about Wikiauto as new tasks are introduced, because it performs at a lower loss than our multitask baseline with no exposure to Wikiauto. As such, we find that while forgetting occurs, it does so slowly over time and exposure to new tasks.

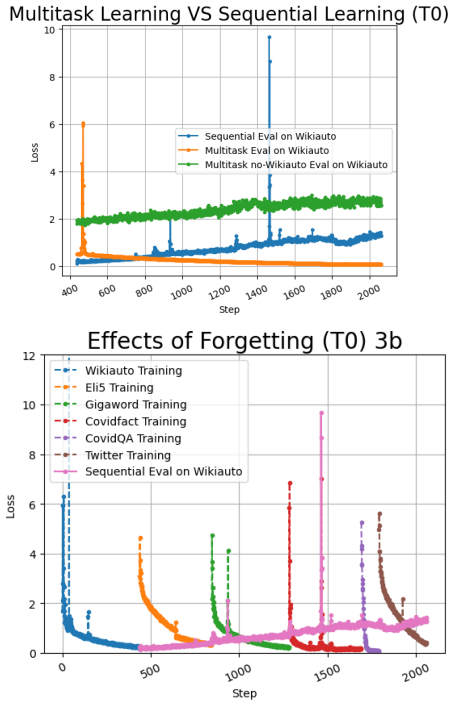

Figure 1. Evaluation on Wikiauto compared to baselines and other tasks during training

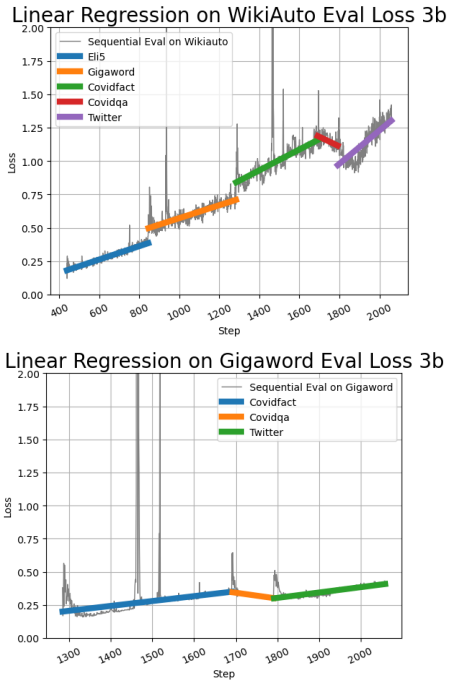

Figure 2. Linear regression on forgetting of Wikiauto and Gigaword for T5-3b

4.2. Linear Form of Forgetting

We also analyze the functional form of forgetting to determine if it can be predicted. Ultimately, we find that validation perplexity on the task being forgotten in our language model is made up of piece wise linear functions with task boundaries (See Figures 2 and Appendix Section 5). This is true across all tasks and shows that forgetting is predictable given its linear form. Furthermore, the rate of forgetting is training and evaluation task dependent. Interestingly, we also find that there are jumps between the piece wise linear functions. By using a constant learning rate, we are able to determine that these are not caused by a reset to the learning rate schedule.

4.3. Various Orderings of Tasks

To determine whether our finding is limited to a particular sequential ordering of tasks, as well as to better understand whether ordering affects the weights of a forgetting function, we train multiple sequential models on random orders of tasks (See Figures 3, Appendix Section 5). We find consistent results to 4.1, specifically that forgetting increases gradually for all tasks, despite learning order. Furthermore, for all orderings, piece wise linear functions can be constructed across all tasks (See Figures 4, Appendix Section 5). However, the weights of these task forgetting functions differ for each order. As such, while the linear function form exists, its exact values are also dependent on previous tasks.

4.4. Model Scale

Finally, we compare training sequential models on the 3b and 11b model to see the effect of model scale on rate of forgetting (See Figure 5 and Appendix Section 5). Our results are consistent across model sizes, and for both the 3b and 11b models, forgetting increases at a gradual rate, and this rate fits a linear functional form separated by task boundaries. Furthermore, this rate is also dependent on the order of training tasks, as well as the evaluation task. Interestingly, model scale also affects the weights of the forgetting function. In general, the rates of forgetting are *higher* for the larger models. This implies that the larger models forget tasks more quickly.

5. Conclusion

In this paper, we provide a comprehensive evaluation on task forgetting in LLMs during sequential learning. Our work reveals a consistent pattern of gradual performance degradation as new tasks are introduced with generally a slow rate of task forgetting. We also show that the degradation of the validation perplexity follows a predictable piecewise linear form, allowing for accurate modeling and estimation of forgetting, regardless of validation/training task, training task ordering, or model scale. However, the exact weights

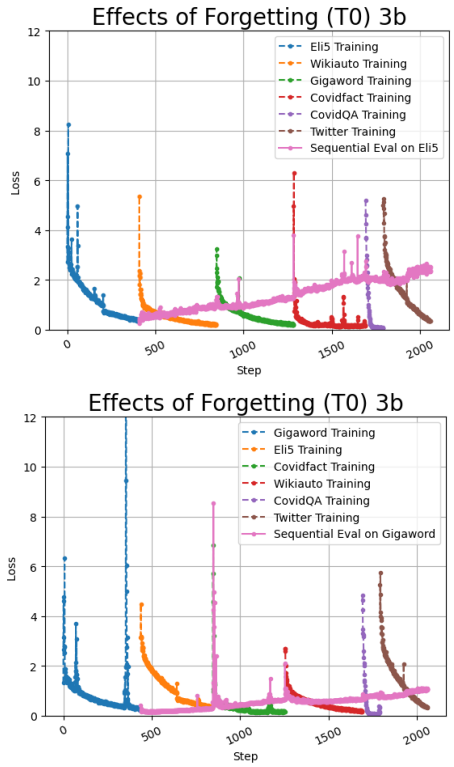

Figure 3. Another ordering for sequential training of tasks with evaluation on Eli5 and Gigaword for T5-3b

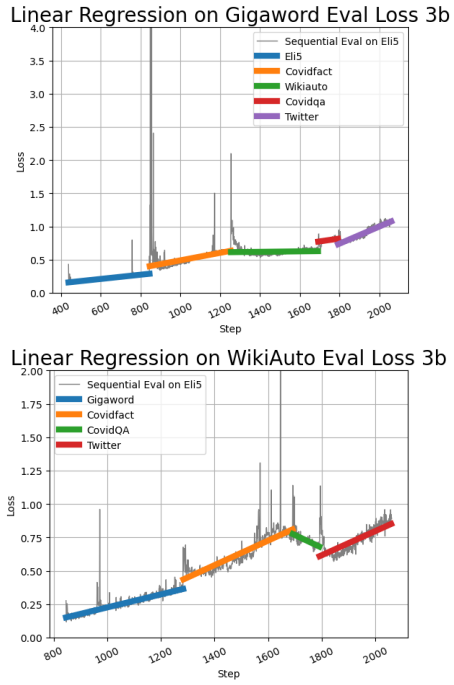

Figure 4. Linear regression on forgetting of Gigaword and Wikiauto for T5-3b

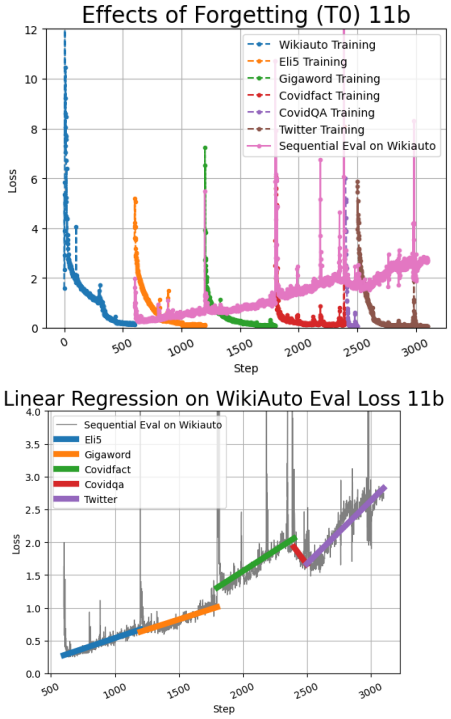

Figure 5. Sequential training of tasks, and linear regresion of evaluation for Wikiauto for T5-3b

of this linear function vary depending on validation/training task, task ordering and model scale.

These findings have practical implications for the design and optimization of LLM-based systems. By understanding the predictable nature of task forgetting, we can better manage and mitigate its impact, enhancing the long-term stability and robustness of LLMs in real-world applications. Future research can explore methods to mitigate forgetting. Additionally, investigating the influence of task similarity and its effect on performance degradation in LLMs can provide further insight into understanding how these models learn.

References

Aljundi, R., Rohrbach, M., and Tuytelaars, T. Selfless sequential learning, 2019.

Belouadah, E., Popescu, A., and Kanellos, I. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.12.003. URL https://www.sciencedirect.com/science/article/pii/S0893608020304202.

Bin Tareaf, R. Tweets Dataset - Top 20 most followed users in Twitter social platform, 2017. URL https://doi.org/10.7910/DVN/JBXKFD.

Caruana, R. Multitask learning. *Machine Learning*, 28, 07 1997. doi: 10.1023/A:1007379606734.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/tpami.2021.3057446. URL https://doi.org/10.1109%2Ftpami.2021.3057446.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346.

Hadsell, R., Rao, D., Rusu, A., and Pascanu, R. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24:1028–1040, 12 2020. doi: 10.1016/j.tics.2020.09.004.

Jiang, C., Maddela, M., Lan, W., Zhong, Y., and Xu, W. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7943–7960, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.709. URL https://aclanthology.org/2020.acl-main.709.

Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., and Díaz-Rodríguez, N. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges, 2019.

Lesort, T., Ostapenko, O., Misra, D., Arefin, M. R., Rodríguez, P., Charlin, L., and Rish, I. Challenging common assumptions about catastrophic forgetting, 2023.

Möller, T., Reina, A., Jayakumar, R., and Pietsch, M. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.nlpcovid19-acl.18.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning, 2017.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Biderman, S., Gao, L., Bers, T., Wolf, T., and Rush, A. M. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207, 2021. URL https://arxiv.org/abs/2110.08207.

Scialom, T., Chakrabarty, T., and Muresan, S. Fine-tuned language models are continual learners, 2022.

Datasets

- **WikiAuto**: a set of aligned sentences from English Wikipedia and Simple English Wikipedia as a resource to train sentence simplification systems (Jiang et al., 2020) (∼450,000 datapoints)
- **Eli5**: an English-language dataset of questions and answers gathered from three subreddits where users ask factual questions requiring paragraph-length or longer answers about general topics, science, and history (Fan et al., 2019) (∼210,000 datapoints)
- **Gigaword**: headline-generation on a corpus of article pairs from Gigaword with additional constraints abou specific words in the output. (Scialom et al., 2022) (∼450,000 datapoints)
- **Covidfact**: A collection of statements about Covid scraped from web sources with positive and negative labels (”Yes”, ”No). We can not find a direct citation to this specific dataset from (Scialom et al., 2022) and thus assume that it was created by (Scialom et al., 2022) (∼210,000 datapoints)
- **COVID-QA**, a Question Answering dataset consisting of annotations by volunteer biomedical experts on scientific articles related to COVID-19, as well as annotations by experts on scientific articles from the CORD-19 dataset. (Möller et al., 2020) (∼350,000 datapoints)
- **Twitter Top 20**: A dataset consisting of hashtags and a an author, where the goal is to generate a relevant tweet by fine-tuning data from the top 20 most followed users in Twitter (Bin Tareaf, 2017) (∼140,000 datapoints)

Experimental Setup

We use the following hyperparameters for all the models we finetune, many of which are based on (Scialom et al., 2022) hyperparameters:

3B Model

- WikiAuto Training Epochs: 1
- Eli5, Gigaword, Covidfact, Twitter Training Epochs: 2
- COVID-QA Max Steps: 100

11B Model

- WikiAuto Eli5, Gigaword, Covidfact, Twitter Max Steps: 600
- COVID-QA Max Steps: 100

Both

- Max Sequence Length: 64
- Train Batch Size: 1024
- Devices (GPUs): 4
- Gradient Accumulation Steps: 32
- Constant Learning Rate: 0.001

We train and evaluate our models using 4 A100 GPUs. Additionally, we use DeepSpeed ZeRO-Stage 3, which shards optimizer states, gradients, model parameters across data parallel workers/GPUs. Furthermore, we also offload the gradients and optimizer states to CPU/Disk building, and the model parameters to CPU/Disk building.

Orderings

For the T0_3b model we train the following random orderings:

1. WikiAuto, Eli5, Gigaword, Covidfact, CovidQA, Twitter

2. Eli5, WikiAuto, Gigaword, Covidfact, CovidQA, Twitter

3. Gigaword, Eli5, Covidfact, Wikiauto, CovidQA, Twitter

For the T0_pp model we train the following random orderings:

1. WikiAuto, Eli5, Gigaword, Covidfact, CovidQA, Twitter

2. Eli5, WikiAuto, Gigaword, Covidfact, CovidQA, Twitter

Evaluation on Additional Tasks

Sequential Loss and Linear Regressions on Loss (3B)

Ordering 1

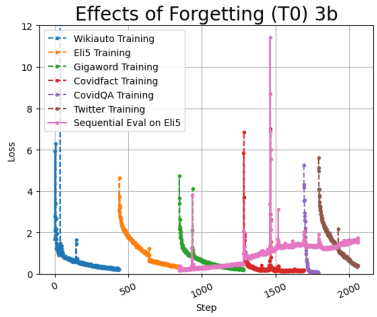

Figure 6. The sequential training of tasks with evaluation for T5-3b on Eli5

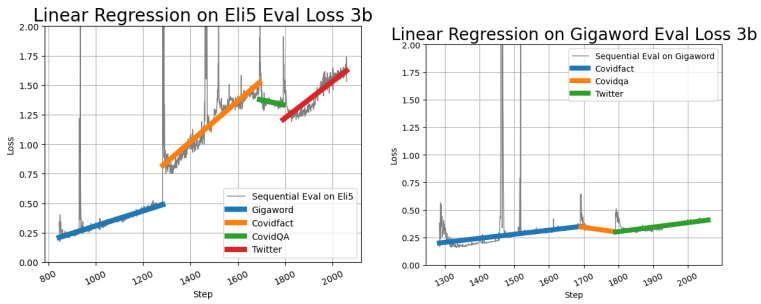

Figure 7. Linear regression on loss for T5-3b on Eli5 and Gigaword

Ordering 2

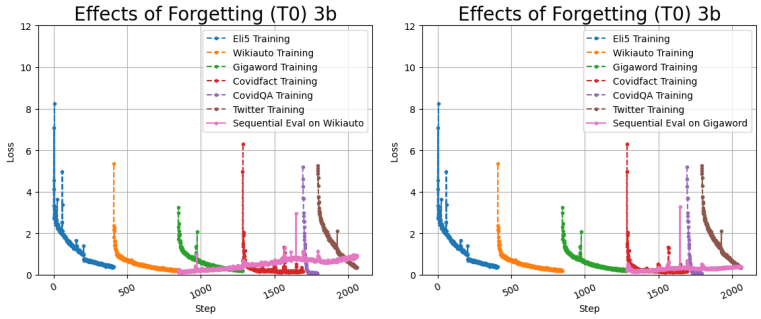

Figure 8. The sequential training of tasks with evaluation on Wikiauto for T5-3b on Wikiauto and Gigaword

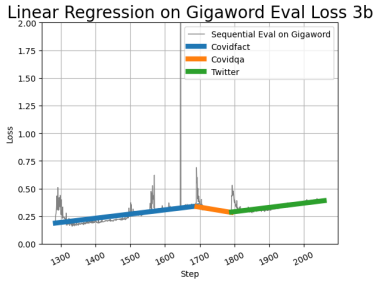

Figure 9. Linear regression on loss for T5-3b on Gigaword

Ordering 3

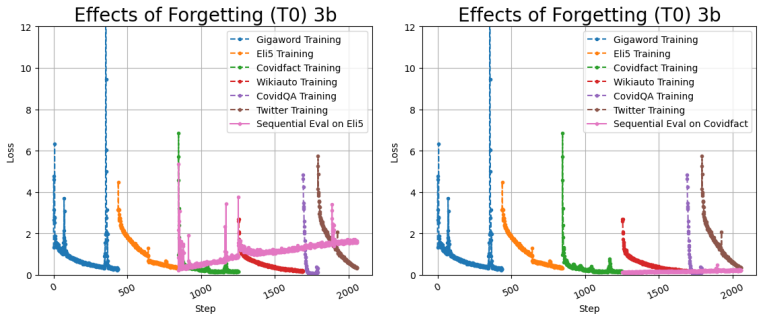

Figure 10. The sequential training of tasks with evaluation for T5-3b on Eli5 and Covidfact

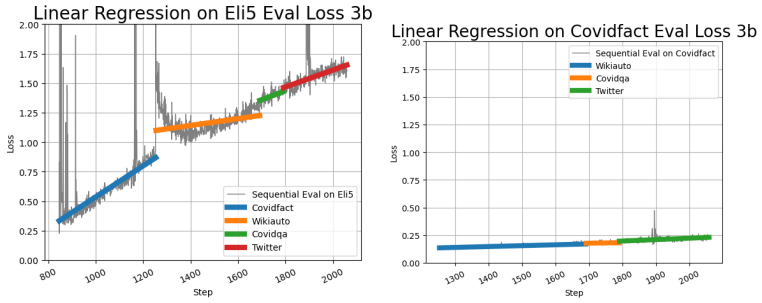

Figure 11. Linear regression on loss for T5-3b on Eli5 and Covidfact

Sequential Loss and Linear Regressions on Loss (11B)

Ordering 1

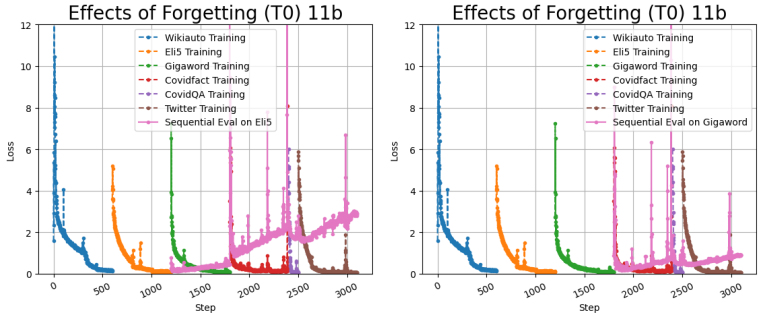

Figure 12. The sequential training of tasks with evaluation for T5-11b on Eli5 and Gigaword

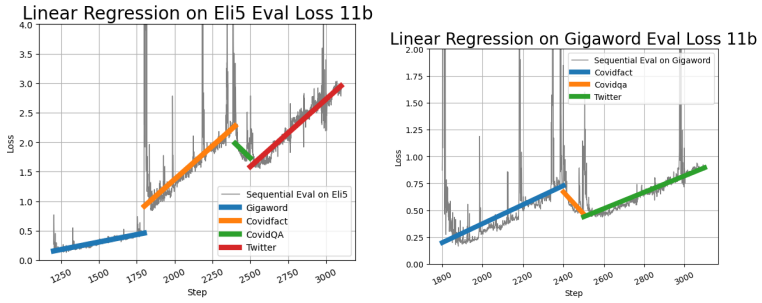

Figure 13. Linear regression on loss for T5-11b on Eli5 and Gigaword

Ordering 2

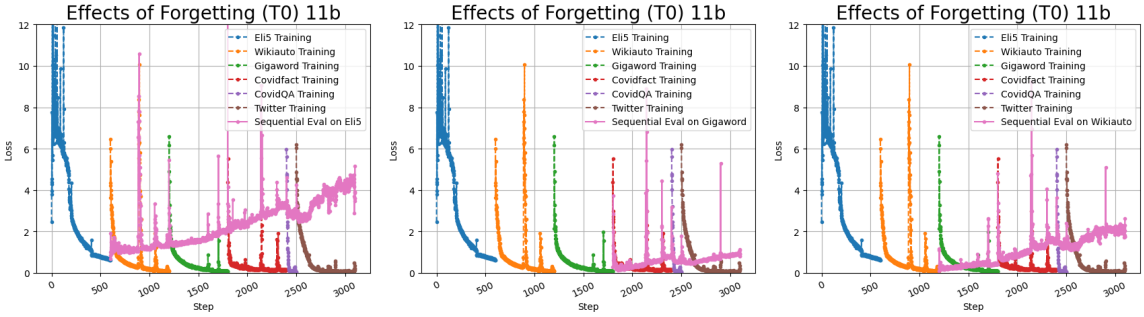

Figure 14. The sequential training of tasks with evaluation for T5-11b on Eli5, Gigaword, and Wikiauto

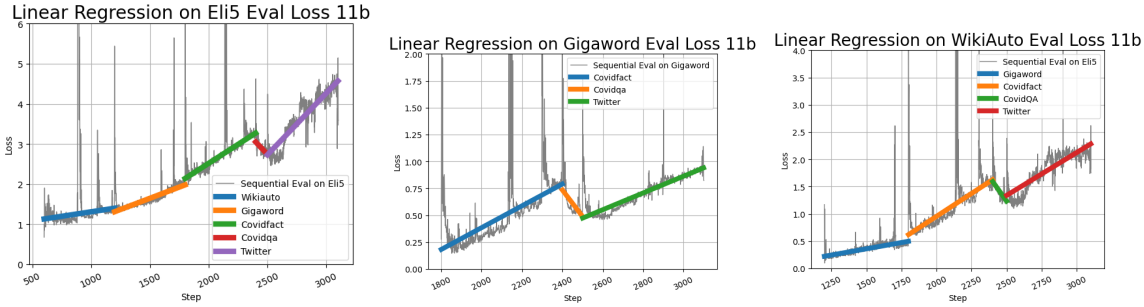

Figure 15. Linear regression on loss for T5-11b on Eli5, Gigaword and Wikiauto