Comparing Knowledge Injection Methods for LLMs in a Low-Resource Regime

Hugo Abonizio[1,4], **Thales Almeida**[2,4], **Roberto Lotufo**[1,3], **Rodrigo Nogueira**[4]

[1] Faculdade de Engenharia Elétrica e de Computação (FEEC)
University of Campinas (Unicamp) – Campinas, SP – Brazil

[2]Instituto de Computação (IC)
University of Campinas (Unicamp) – Campinas, SP – Brazil

[3]NeuralMind – Brazil

[4]Maritaca AI – Brazil

Abstract. *Large language models (LLMs) often require massive corpora to effectively acquire new knowledge, yet updating a model with only thousands – or even a few million – tokens remains difficult. In this work, we study how to inject such small, unstructured knowledge and how this affects catastrophic forgetting. Using a recent-news corpus, we probe models with question-answer pairs to assess knowledge acquisition. Alongside a continued-pre-training baseline, we evaluate data augmentation strategies that create synthetic variations. Our results show that continued pre-training alone yields modest gains, whereas diverse rephrasings significantly boost learning – especially when prompts maximize variability. We also evaluate forgetting in this low-data regime and find that retrieval-augmented generation (RAG) updates, though effective, degrade out-of-domain performance more than parametric approaches. Finally, we show that the model can produce high-quality synthetic data itself, pointing toward self-improving updates. Code and data are available at* `https://github.com/hugoabonizio/knowledge-injection-methods/`.

1. Introduction

Previous work has shown that large language models (LLM) trained with a self-supervised objective learn large amounts of information, effectively acting as knowledge bases [Petroni et al. 2019]. Likewise, updating the knowledge of a model through continued pre-training [Gururangan et al. 2020], with the goal of specializing an existing LLM in a domain such as medicine [Singhal et al. 2023], has also shown to be successful. In these scenarios, the amount of training data spans from billions to trillions of tokens.

However, perhaps surprisingly, incorporating relatively small amounts of information (e.g., thousands or millions of tokens) has proven to be more challenging, often resulting in performance degradation or only marginal gains when done naively [Ovadia et al. 2023], and potentially even increasing hallucinations.

An additional complication is the forgetting problem, where new information can often be successfully injected at the expense of forgetting previously learned knowledge, a phenomenon commonly referred to as catastrophic forgetting [Goodfellow et al. 2014].

To address these problems, recent literature on knowledge injection often concentrates on two extremes. In the first, also referred to as model editing, methods that

require the information to be learned can be expressed as well-defined entities and relations [Meng et al. 2022]. However, the techniques proposed in these works cannot be easily applied to real documents without modifications, as it is challenging to convey the complex information as a knowledge graph with a finite number of possible relationships. For example, converting the following headline into a set of discrete triples presents a non-trivial challenge:
"*Advocates for Ukraine's surprise incursion into the Russian territory say it will provide Kyiv with vital leverage for any future peace talks (December 2nd, 2024)*".

On the other end of the spectrum, works on domain adaptation methods often build upon datasets on the order of billions of tokens to learn new information [Cheng et al. 2024]. These approaches require substantial computational resources and are, therefore, only feasible for organizations with large-scale infrastructure and access to vast amounts of data. In many private applications or niche domains, however, data can be scarce, and compute resources are limited, making such large-scale methods impractical in some real-world settings.

In parallel, retrieval-augmented generation (RAG) [Lewis et al. 2020] methods aim to inject knowledge through in-context learning, rather than by updating the model's parametric knowledge. Some studies have compared these two approaches, highlighting different advantages in each case [Balaguer et al. 2024, Mecklenburg et al. 2024]. However, it is important to note that these approaches are orthogonal: parametric knowledge injection and in-context learning methods can be combined and are not mutually exclusive.

In this work, we focus on the middle ground between the two extremes. Our goal is to study the learning dynamics of small, unstructured information in an efficient manner, without forgetting previously acquired knowledge. We investigate the learning–forgetting tradeoff using a small corpus and evaluate different continual pre-training techniques including augmentation algorithms that leverage synthetic variations of the original data, aiming to overcome the challenges of learning in a small-data regime.

To measure the learning effectiveness, we need a dataset containing information that is both new to the model – i.e., it was not seen during its pre-training – and complex, to make the results impactful for real-world applications. Thus, we chose the TiEBe dataset [Almeida et al. 2025], which contains news articles with the required recency, and question-answer pairs to measure knowledge and understanding.

Our results indicate that directly continuing pre-training on a set of documents using self-supervised learning [Gururangan et al. 2020] (i.e., next-token prediction) has limited effectiveness. Additionally, augmentation methods suggest that training on multiple versions of the same source document is necessary. Our findings also shed light on why LLMs require vast amounts of training data: intuitively, learning a small piece of information should not require 20 variations – one or two should suffice. Addressing this inefficiency could make training these models – which currently cost millions of dollars – significantly more affordable.

In summary, our main contributions are as follows:

- We conduct an extensive analysis of different knowledge injection approaches, including RAG, continued pre-training and augmentation techniques. We compare

different techniques and their variations, providing evidence that models benefit from exposure to diverse data variations to effectively learn new information.
- We propose an evaluation methodology to assess the effectiveness of knowledge injection using small and unstructured datasets.
- We provide insights into the challenges of continued pre-training in small-scale data regimes and propose strategies to address training instabilities in this context.
- We release a set of synthetically augmented corpora along with the code to reproduce and expand them, supporting future research.

2. Related Work

Recent work on injecting new knowledge into LLMs can be divided into three categories: (1) model editing techniques that modify entity relationships within the model parameters, (2) knowledge injection through in-context learning, and (3) knowledge injection via some form of continual training. In this paper, we turn our attention to the latter two, examining the dichotomy between retrieval-augmented generation and fine-tuning.

The majority of model editing works rely on structured entity triple data. The ZsRE and CounterFact datasets used in [Meng et al. 2022], as well as KnowEdit [Zhang et al. 2024b], are examples of such approaches, where models must learn to connect entities through predefined relations. However, we consider these techniques to be out of scope for the present work, as they diverge from real-world use cases, where textual data is mostly unstructured.

Another common approach to knowledge ingestion involves coupling the LLM with a retriever that has access to a database containing the relevant information [Zhang et al. 2024a, Gao et al. 2024b], a method generally referred to as RAG [Lewis et al. 2020]. Because it does not require model training, it is relatively cost-effective for real-world scenarios. However, later in this work, we will demonstrate that this approach has some drawbacks. Some frequently discussed drawbacks in the literature include dependence on retriever quality and the chunking strategy, as well as the model's ability to handle provided context [Liu et al. 2024].

Finally, in the last category of related work are methods that leveraged continued training, either by continual pre-training or fine-tuning. Here, we refer to continual pre-training as the self-supervised training using next-token prediction, while fine-tuning methods are the ones trained using instruction-tuning [Wei et al. 2022].

[Ovadia et al. 2023] compared RAG with continual pre-training on synthetic paraphrases, finding that exposing the new information in diverse ways through paraphrases play an important role. Their results found that RAG knowledge injection to be more effective than self-supervised finetuning. The work also generated questions to probe the effectiveness of knowledge injection.

[Cheng et al. 2024] proposed methods for synthetically augmenting the pre-training corpus by transforming the original examples using different tasks. Their results show advantages over vanilla pre-training in the original documents, highlighting the importance of variations for effective learning.

[Balaguer et al. 2024] and [Mecklenburg et al. 2024] compared RAG and fine-tuning methods by training on synthetic pairs of questions and answers. Both work

showed a significant increase in performance after fine-tuning and [Balaguer et al. 2024] showed that RAG and fine-tuning can be combined synergistically.

[Wu et al. 2024] also investigated the knowledge injection through fine-tuning on question-answer (QA) pairs and used recent news as one of their evaluation datasets. Their results showed that learning from this fine-tuning is limited.

[Yang et al. 2024] introduced the EntiGraph, a knowledge graph-based augmentation technique for generating diverse synthetic text. They found that their approach is complementary to RAG, improving downstream performance even when the original documents were available at inference time. While their work is closely related to ours, we place a stronger emphasis on mitigating data contamination risks and explore knowledge injection with an even smaller training corpus.

3. Methods

In this section, we describe our evaluation methodology, including the dataset used and the knowledge injection techniques evaluated.

3.1. Dataset

We chose to inject knowledge related to recent news articles for two reasons. First, news articles carry complex forms of knowledge expression, which we argue are more aligned with real-world challenges researchers and practitioners will face when keeping an LLM continuously up-to-date. This contrasts with simpler forms of knowledge, such as facts encoded as knowledge triples (e.g., "John Smith works at ACME Corporation"). A news article might be incomplete (e.g., it mentions key people participating in an ongoing event but does not define their roles, assuming the reader has been following the event for a while), or it might contradict other documents (e.g., "investment X is no longer recommended due to fraud scandals").

Second, using the news domain we can ensure that the model has not been exposed to that specific information previously. Given the recency of the news, we can mitigate the contamination problem using news about events that occurred after the model's training cutoff.

To address these two criteria, we used the TiEBe dataset [Almeida et al. 2025][1], a dataset of news articles spanning from 2015 to 2024, along with a corresponding set of question-answer pairs, to evaluate models on their knowledge of specific events. The questions and answers were generated using `GPT-4o-2024-08-06`, by prompting it to produce four pairs per article in a single generation. More details on the creation of the data set can be found in [Almeida et al. 2025].

To run the knowledge injection experiments, we chose the Llama-2 model [2], which has an old enough knowledge cutoff – September 2022 – while having strong performance in the question-answering task due to being instruction-finetuned. This mitigates the risk of data contamination because the model was not exposed to the specific events covered in the documents during the pre-training.

[1] `https://huggingface.co/datasets/TimelyEventsBenchmark/TiEBe`
[2] `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`

However, since Llama-2 has a limit of 4k tokens in its context, we selected a subset of the *World* category filtering articles with up to 3,500 tokens to fit the article, the instruction template, and the generated answer within the model's context length. Additionally, we selected only the recent documents, from 2023 to 2024, as the training corpus. The final dataset comprises 117 documents, each paired with four QA pairs – 468 QA pairs, in total.

For the automatic evaluation of the models' answers, we followed the methodology described in [Almeida et al. 2025], applying the process commonly known as LLM-as-a-judge [Zheng et al. 2023]. This approach leverages the expected answers provided in the dataset, prompting an LLM to assess the correctness of candidate answers based on these true answers. This more sophisticated way of evaluating the answers is required to check whether the model has learned the complex facts contained in the documents, whereas strict approaches, such as exact matching, may fail to capture these nuances. The prompt and evaluation code we used are available in the released repository.

3.2. Control Datasets

One of the challenges in updating a model's knowledge is ensuring it retains previously learned capabilities. Large models can easily memorize (i.e., fully reconstruct) a small set of documents, which is a key goal of the knowledge injection task. However, this often comes at the cost of significant performance drops on unrelated tasks where the original model previously excelled.

To quantify this forgetting gap, we use the average accuracy across the following seven datasets (OpenBookQA, ARC-Easy, ARC-Challenge, WinoGrande, HellaSwag, PIQA, and BoolQ), collectively referred to as the *Control datasets*. These datasets are implemented in the Language Model Evaluation Harness [Gao et al. 2024a], and have been used in prior work for similar evaluations.

3.3. Knowledge Injection Techniques

We investigate the following knowledge injection techniques:

Retrieval-Augmented Generation (RAG): One of the simplest and most straightforward methods for injecting knowledge into an LLM is to use a retrieval mechanism to locate relevant information within a corpus and include this information in the prompt provided to the LLM. Specifically, we employ BM25 [Rosa et al. 2021] to retrieve the top-N documents from the corpus of recent news, which consists of 117 documents.

To evaluate different configurations, we tested a document retrieval approach, where the best-matching document is prepended to the prompt, followed by the test question. Additionally, we evaluated a chunking-based approach, where documents are divided into chunks of 512 tokens with an overlap of 64 tokens. In this setup, we retrieved the top-5 chunks and prepended to the question.

Continual Pre-training (CPT): This approach involves continuing the pre-training of the LLM directly on the target document or article using the causal language modeling objective (i.e., next-token prediction) applied to the unmodified corpus [Gururangan et al. 2020].

Rephrasing the Web (RTW): Following the four prompts proposed by [Maini et al. 2024], we generate rephrased versions of a training example. Three of

these prompts instruct the LLM to rephrase the input document in styles with varying levels of complexity: easy (simplified, suitable for a toddler), medium (clear and high-quality, similar to Wikipedia), and hard (terse and abstruse). The fourth prompt asks the LLM to generate QA pairs that the document is likely to address.

We experiment with two models for generating these rephrases: (1) GPT-4o,[3] which provides a high-quality upper bound for rephrases but may represent an unrealistic setup since it could have been exposed to the content during its pre-training; and (2) the model itself (i.e., Llama-2-7B), representing a more practical scenario in which the model does not have prior knowledge of the document being rephrased.

We report the results using all the four styles of prompts, using only the first three, and using only the QA-style prompt. This way we can keep only the rephrasing prompts and isolate the possibility of cross-contamination by generating similar questions as the ones used during the test phase.

Instruction Pre-training (IPT): We applied the instruction generation method proposed by [Cheng et al. 2024] to our training corpus. Using their instruction generation model,[4] we generated synthetic instructions for each training document in a 1-shot setting. A 1-shot approach was necessary, as using more examples would result in truncation due to exceeding the model's context length.

Paraphrasing (Para): To evaluate the effect of simple paraphrasing on training examples, we adapted the prompt proposed by [Ovadia et al. 2023]. The prompt instructed the LLM to rephrase the content while maintaining factual accuracy and maintaining the original text length. We generated multiple paraphrases by applying token sampling with a specified temperature. For this process, we used two models: GPT-4o, providing high-quality paraphrases as an upper bound, and Llama-2-7B, representing a more realistic, self-contained approach.

3.4. Training Setup

The knowledge injection methods relying on continued pre-training were implemented by mixing synthetic augmented examples with the original documents. Therefore, all reported results, along with the corresponding variations, refer to the original 117 documents supplemented with an additional N synthetic variations in the training set.

Starting from the instruction-tuned Llama-2-7B checkpoint, we further trained the model using the causal language modeling objective with the traditional cross-entropy loss. Training was conducted on batches of 8 examples, with a learning rate of `5e-5`, and the AdamW [Loshchilov and Hutter 2019] optimizer. Given the task of injecting small amounts of knowledge, our training runs were intentionally short, often involving fewer than 15 training steps per epoch. To ensure stable training and reduce variance in results, we carefully tuned the hyperparameters in preliminary experiments, determining that training for two epochs with a relatively large learning rate warmup yielded the best performance.

We applied a re-warmup and re-decay strategy to the learning rate, utilizing linear warmup and cosine decay. Specifically, the warmup phase was applied throughout the first

[3] `https://platform.openai.com/docs/models`

[4] `https://huggingface.co/instruction-pretrain/instruction-synthesizer`

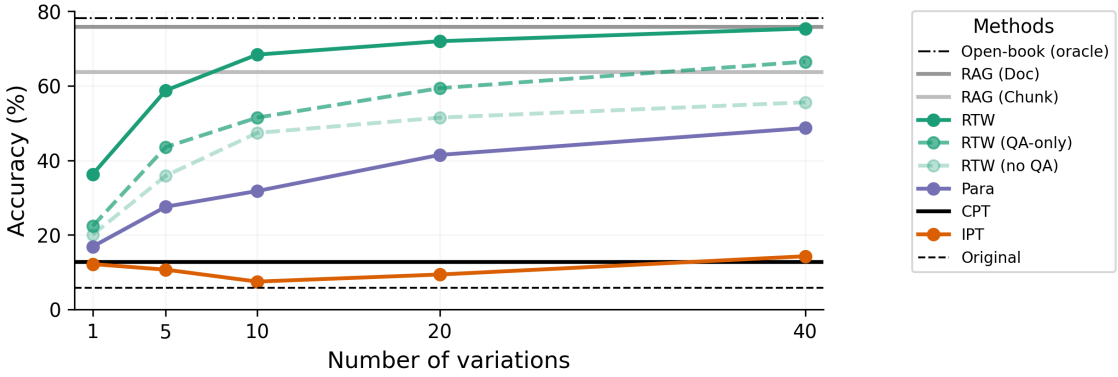

Figure 1. Comparison of different knowledge injection methods.

epoch, while the decay phase occurred during the second and final epoch. This approach allowed us to re-warm the learning rate during the first half of training and re-decay it during the second half, optimizing the stability and effectiveness of the training process.

4. Results

In this section, we describe and analyze the results of our experiments and conduct ablations comparing variations of the studied methods.

4.1. Which method is the best?

Fig. 1 summarizes the comparison of the methods evaluated in this work. The dashed black lines show the lower and upper bounds, which correspond to the closed-book answering performance (when the model relies solely on its parametric knowledge) and the open-book answering performance (when the model has access to the context that answers the given question) using the original model.

Next, still using open-book answering but without providing the exact correct context (oracle), we evaluated RAG approaches. This is a more realistic scenario because in real-world applications the pairing of a question and its relevant context is not known a priori. For RAG evaluation, we used the top-1 most similar document according to the BM25 similarity score, as well as the top-5 most similar chunks. The results show very similar performance between the upper-bound oracle and RAG using the top-1 document. However, a significant drop is observed when using chunks, highlighting a known caveat of RAG systems and their sensitivity to chunking strategies.

The solid black line represents the CPT performance, serving as the baseline method for injecting knowledge from unstructured text. The other colored lines correspond to each augmentation method under comparison. For all methods, we repeated the synthetic generation N times (shown on the x-axis), using a sampling temperature of 1.0 to introduce variation in the generated examples.

For the RTW method, we report separate results for different prompt configurations: (i) using all four proposed prompts (easy, medium, hard, and QA-style), and (ii) two variations – one excluding the QA-style prompt and another using only the QA-style prompt [Maini et al. 2024]. This separation was implemented to measure the impact of the generated QA pairs and to mitigate the risk of indirect contamination, where the model

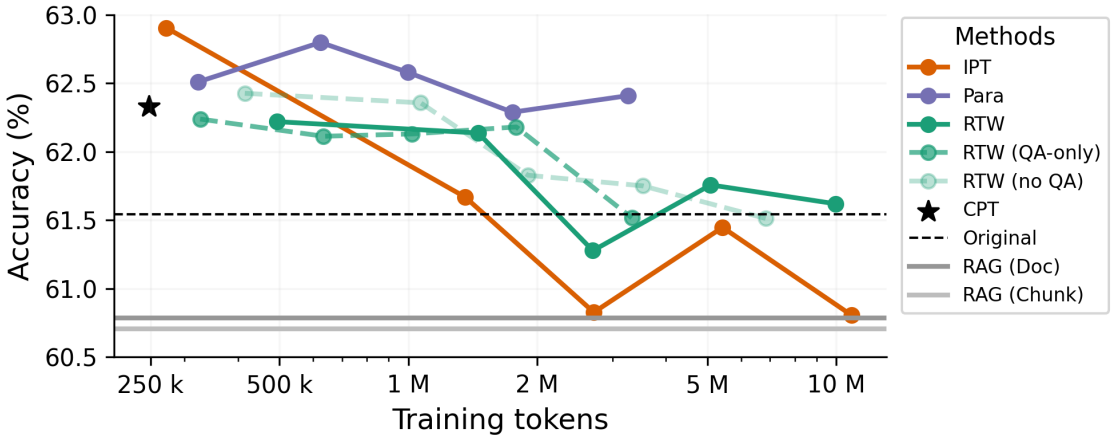

Figure 2. Comparison of the average accuracy on the control dataset against the amount of training tokens.

might generate QA pairs similar to those encountered during testing, thus making the task artificially easier. Although we ensured that the QA pairs generated by RTW do not overlap with those in the test set, we report these results separately as a precaution.

The results indicate a monotonic increase in performance when using the RTW and Para methods, with RTW achieving a performance level comparable to document-level RAG, only a few points below the upper bound. The RTW variant that excludes the QA-style prompt and the Para method both yield similar performance, approaching that of chunk-based RAG. However, the superior performance of RTW suggests that leveraging multiple prompts to generate textual variations is more effective than relying on a single paraphrasing prompt.

The RTW variation that uses only QA-style prompts outperforms the configuration using only the other three prompts. This result indicates that synthetically generated QA pairs play an important role in increasing the model's ability to answer questions based on learned information. However, concerns about indirect contamination remain, suggesting that the no-QA variant may be better suited for real-world applications.

The IPT method scored lower than expected, hovering around the CPT baseline. This might be due to using a one-shot approach instead of the few-shot approach that worked best in the original study – a strategy that was not feasible here because of the longer texts used, which would exceed the model's context-length limit.

4.2. Learning-forgetting tradeoff

To measure the possible catastrophic forgetting, we evaluated each checkpoint on the control dataset and averaged the accuracy across seven different tasks. This gives us a measure of how the model performs on tasks it previously knew.

Fig. 2 compared the control dataset accuracy with the total training tokens of each method on each number of generated variations (1, 5, 10, 20, and 40). The CPT baseline is indicated by a star, since its amount of training tokens is fixed, and we compared with the original performance and the RAG variants.

To evaluate the RAG performance, we prepended the retrieved context on each

evaluated prompt, following the same prompt used on non-RAG evaluations. Thus, their evaluations are exactly the same as the other methods, the only difference is that the retrieval result is included in the context. It is noteworthy that both RAG approaches lead to the highest degradation compared to the other methods, with the exception of IPT. This performance degradation highlights the caveats of using RAG-based approaches, where the retrieved context may *confuse* the model on out-of-domain tasks.

Surprisingly, all continued pre-training methods evaluated result in an increase in performance when training with a small number of tokens. This overall performance improvement was observed across all datasets except BoolQ and WinoGrande. One possible explanation for this result is that we start from an instruction-tuned model and evaluate it by computing log probabilities on multiple-choice tasks using the Language Model Evaluation Harness framework [Gao et al. 2024a]. This evaluation approach may favor base models over instruction-tuned models due to probability calibration. Thus, this short continued pre-training on the next-token prediction task may resemble the original pre-training objective, leading to performance gains on some control datasets. However, we leave a deeper investigation of this phenomenon to future work.

The results indicate that, as the amount of training tokens increase, the average performance tends to drop. This result is consistent with the catastrophic forgetting phenomenon. This is not entirely true for the Para method, which oscillated on similar performances, but a more informed conclusion would require more tokens to check if the trend holds, since it is the method that resulted on the lowest amount of tokens.

Despite the difference in the in-domain accuracy, i.e., the accuracy on the recent news dataset, both RTW methods achieved similar performance on the control dataset, even though the dataset with all four types of prompt resulted in a slightly higher amount of generated tokens.

The IPT method exhibited the largest drop in performance, suggesting that incorporating synthetic instructions generated by its synthesizer accelerated the model's forgetting of previously learned capabilities. Moreover, IPT showed low performance both in-domain and out-of-domain, indicating that the one-shot generation approach used in our experiments may be more detrimental than beneficial in our evaluated scenario.

4.3. Can models augment themselves?

Previous results show that models can effectively learn new information by continuous pre-training on new data and benefits from synthetically augmented data. In this section, we investigate the role of the generator model used to augment the dataset. We used the RTW and Para techniques, since the IPT uses their specific synthesizer model, instead of a generic LLM.

Fig. 3 shows the comparison of the different generator models: GPT-4o and Llama-2-7B. For simply paraphrasing the content (Para), there was no significant difference of using a frontier model or the model itself to generate the synthetic training data. For the RTW, which uses varied prompts to augment the data, it is inconclusive whether one model performs better than the other because using all four prompts the GPT-4o lead to better results, but without the QA-style prompt, the model itself leads to better results.

One potential caveat is that GPT-4o was the same model used to generate the

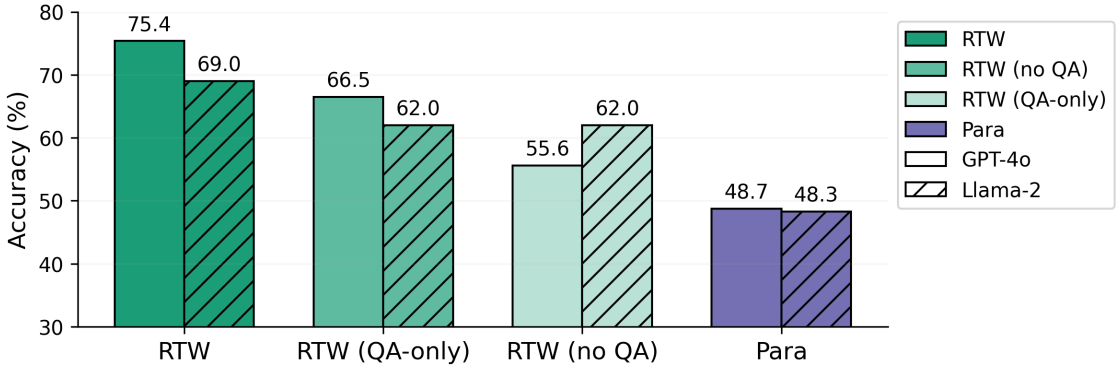

Figure 3. Performance on the TiEBe dataset using different models to generate the synthetic training data.

questions for the original TiEBe dataset [Almeida et al. 2025]. Thus, an unwanted indirect contamination may explain it leading to better performance, since the model might have generated similar questions for the knowledge injection methods and for our test set. This hypothesis also explains why removing the QA prompt from RTW resulted in a large drop in performance when using GPT-4o as a generator. However, this drop is smaller when using LLaMA-2-7B as the generator.

Even though the results of the QA-only variant are higher with the GPT-4o generator, there is no difference when using the model itself as the generator. This highlights the performance achieved by the no-QA variant, which avoids the risk of exposing the model to similar QA pairs in the test, while showing the generalization of the learned information due to solely training the model on rephrased versions of the original text.

Achieving comparable results with the smaller LLaMA-2-7B model and a state-of-the-art model is particularly noteworthy. It shows that the model can generate synthetic data to enhance its own capabilities, suggesting the possibility of continuous or iterative self-improvement through the ingestion of newly generated data.

5. Conclusion

In this work, we investigated methods for injecting small-scale, unstructured knowledge into LLMs and examined the tradeoff between learning new facts and retaining prior knowledge. We found that simple continued pre-training yields modest improvements, while RAG can be effective but often degrades performance on unrelated tasks.

Our results highlight the importance of synthetic data augmentation: models trained on diverse rephrasings (e.g., RTW) learn new information more effectively while avoiding catastrophic forgetting. Notably, models can generate their own augmentation data, opening avenues for self-improving updates without external supervision.

Our findings emphasize the need for diverse training inputs to enhance knowledge acquisition while minimizing the degradation of previously learned information. We hope our released datasets and code will support future research on efficient knowledge injection.

References

Almeida, T. S., Bonás, G. K., Santos, J. G. A., Abonizio, H., and Nogueira, R. (2025). Tiebe: A benchmark for assessing the current knowledge of large language models.

Balaguer, A., Benara, V., de Freitas Cunha, R. L., de M. Estevão Filho, R., Hendry, T., Holstein, D., Marsman, J., Mecklenburg, N., Malvar, S., Nunes, L. O., Padilha, R., Sharp, M., Silva, B., Sharma, S., Aski, V., and Chandra, R. (2024). Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture.

Cheng, D., Gu, Y., Huang, S., Bi, J., Huang, M., and Wei, F. (2024). Instruction pre-training: Language models are supervised multitask learners. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. (2024a). A framework for few-shot language model evaluation.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024b). Retrieval-augmented generation for large language models: A survey.

Goodfellow, I. J., Mirza, M., Da, X., Courville, A. C., and Bengio, Y. (2014). An empirical investigation of catastrophic forgeting in gradient-based neural networks. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Maini, P., Seto, S., Bai, R., Grangier, D., Zhang, Y., and Jaitly, N. (2024). Rephrasing the web: A recipe for compute and data-efficient language modeling. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 14044–14072, Bangkok, Thailand. Association for Computational Linguistics.

Mecklenburg, N., Lin, Y., Li, X., Holstein, D., Nunes, L., Malvar, S., Silva, B., Chandra, R., Aski, V., Yannam, P. K. R., Aktas, T., and Hendry, T. (2024). Injecting new knowledge into large language models via supervised fine-tuning.

Meng, K., Bau, D., Andonian, A. J., and Belinkov, Y. (2022). Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*.

Ovadia, O., Brief, M., Mishaeli, M., and Elisha, O. (2023). Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Rosa, G. M., Rodrigues, R. C., Lotufo, R., and Nogueira, R. (2021). Yes, bm25 is a strong baseline for legal case retrieval.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. (2023). Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Wu, E., Wu, K., and Zou, J. (2024). Finetunebench: How well do commercial fine-tuning apis infuse knowledge into llms?

Yang, Z., Band, N., Li, S., Candès, E., and Hashimoto, T. (2024). Synthetic continued pretraining.

Zhang, J., Cui, W., Huang, Y., Das, K., and Kumar, S. (2024a). Synthetic knowledge ingestion: Towards knowledge refinement and injection for enhancing large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21456–21473.

Zhang, N., Yao, Y., Tian, B., Wang, P., Deng, S., Wang, M., Xi, Z., Mao, S., Zhang, J., Ni, Y., Cheng, S., Xu, Z., Xu, X., Gu, J.-C., Jiang, Y., Xie, P., Huang, F., Liang, L., Zhang, Z., Zhu, X., Zhou, J., and Chen, H. (2024b). A comprehensive study of knowledge editing for large language models.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.