

Novel-WD: Exploring acquisition of Novel World Knowledge in LLMs Using Prefix-Tuning

Anonymous ACL submission

Abstract

Teaching new information to pre-trained large language models (PLM) is a crucial but challenging task. Model adaptation techniques, such as fine-tuning and parameter-efficient training have been shown to store new facts at a slow rate; continual learning is an option but is costly and prone to catastrophic forgetting. This work studies and quantifies how PLM may learn and remember new world knowledge facts that do not occur in their pre-training corpus, which only contains world knowledge up to a certain date. To that purpose, we first propose NOVEL-WD, a new dataset consisting of sentences containing novel facts extracted from recent Wikidata updates, along with two evaluation tasks in the form of causal language modeling and multiple choice questions (MCQ). We make this dataset freely available to the community, and release a procedure to later build new versions of similar datasets with up-to-date information. We also explore the use of prefix-tuning for novel information learning, and analyze how much information can be stored within a given prefix. We show that a single fact can reliably be encoded within a single prefix, and that the prefix capacity increases with its length and with the base model size.

1 Introduction

Pre-trained language models (PLM or LLM) (Chiang et al., 2022) are typically trained on raw texts with a self-supervised loss and further adapted to downstream tasks with, e.g., finetuning (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2019). Hence, the world knowledge that PLM have acquired is prior to the cut-off date of their pre-training corpus (Alivanistos et al., 2022; Kucharyav et al., 2023). A major challenge is then how to reliably teach PLMs novel factual knowledge. Fine-tuning has been one of the main proposed approaches to adapt pre-trained models to new tasks and domains. However, full model fine-tuning

can lead to catastrophic forgetting (French, 1999; Kirkpatrick et al., 2017), and can be costly when performed on large models (Strubell et al., 2020). Furthermore, Wei et al. (2023) showed that when fine-tuning a model on a small corpus with new information, the model may instead learn to hallucinate unseen facts. Parameter-efficient fine-tuning (PEFT) methods have emerged as a lightweight alternative to full model fine-tuning, in which only a fraction of the parameters of the original model are modified. PEFT allows for efficiently modifying a small fraction of model parameters using methods such as prefix-tuning (Li and Liang, 2021), adapter-tuning (He et al., 2021) or LoRA (Hu et al., 2021). In-context learning (Logan IV et al., 2022), prompting (Liu et al., 2023b) and prompt-tuning (Lester et al., 2021) are currently amongst the most reliable ways to inject new knowledge in PLM.

In this study, we focus on prefix-tuning (Li and Liang, 2021), a fine-tuning method in which the pre-trained model parameters are kept frozen, but a few small continuous vectors called the *prefix* are optimized. Based on the idea that context can steer a language model without changing its parameters, prefix-tuning optimizes the model’s context as one or several continuous vectors corresponding to either embeddings or to key-query pairs in attention layers, whose effects will be propagated to all activation layers and subsequent tokens.

Wang et al. (2022) and Liu et al. (2022a) showed that novel knowledge can efficiently be contextually fed into large language models through prompting. However, the size of a prompt in a given model is limited by the context size of that model. In this paper, we view prefix-tuning as a generalized form of prompting taking continuous values, and having controllable depth and length, and as such, we hypothesize that this method can reliably store significant amounts of factual information. This is backed by the findings of Kossen et al. (2023), which argue that in-context learning enables a model to

learn information. Our goal is therefore to investigate this question in the case of prefix-tuning, and more specifically how much knowledge can be compressed into the prefix. In addition, by using prefix-tuning rather than LoRA, fine-tuning or adapters, we hope to avoid the hallucination problem mentioned in (Wei et al., 2023) by working with (generalized) prompts without modifying the existing model weights.

Figure 1 summarizes our proposed approach, which exploits recent Wikidata updates to automatically generate a corpus of new facts: NOVEL-WD. We then propose a nearly automatic procedure to create a dynamic benchmark from this corpus of facts that evaluates updated LLMs in terms of perplexity, new facts generation and accuracy on multiple-choice question-answering. We then evaluate and show that prefix-tuning performs better than LoRA for new facts learning on this dataset.

2 Related work

Adapting models to new tasks is a relatively old problem. Yoon et al. (2018) showed that dynamically expandable networks can obtain good performance in this setting by slowly increasing model capacity. Lin et al. (2022a) explored the task of improving accuracy of Transformer models on out-of-data streams using continual model refinement (CMR) to maximize the diversity of training samples in a non-stationary distribution. Razdaibiedina et al. (2023) showed that using a collection of progressively growing prompts alleviates catastrophic forgetting and increases model generalization capacities across tasks.

Many studies have explored how information storage functions within the Transformer architecture. Elhage et al. (2022) gave a comprehensive overview of the Transformers architecture under the lens of mechanistic interpretability. Geva et al. (2021) showed that the feedforward layers of Transformers models act similarly to key-value memories in information retrieval systems. Based on that work, Mitchell et al. (2021) introduced MEND, a framework that leverages a group of small networks to successfully perform local factual edits within the feedforward layers of a large Transformers model. Meng et al. (2022b,a) expanded on this idea by using causal inference to locate the attention feedforward layer containing a given fact and editing the corresponding matrix as a constrained optimization problem.

In contrast, several approaches for storing new information within a language model have been proposed. One such approach is the use of flexible external memories, as exemplified in (Wu et al., 2021, 2022). Another, dynamic method is that of retrieval systems, which can leverage external knowledge bases, including the Web, to that purpose. Examples of such works include (Guu et al., 2020), (Lewis et al., 2020), (Borgeaud et al., 2021) and (Liu et al., 2023a). Finally, new information can be stored in the short-term through methods such as prompt-tuning (Liu et al., 2021, 2022b).

In terms of evaluation, (Petroni et al., 2019) is an early attempt at measuring relational and factual knowledge within PLMs. Zhu et al. (2020) proposed new, information-theory based evaluation metrics for factual knowledge. Kadavath et al. (2022) and Lin et al. (2022b) focused on measuring model uncertainty as a way to distinguish known facts from hallucinated ones. Jang et al. (2021, 2022) introduced the framework TEMPORALWIKI, which like us, includes a process to generate datasets and benchmarks from information extracted from Wikipedia. However, their framework targets large scale continual learning while we focus on the factual knowledge acquisition point of view (detailed next). This difference in perspective leads to important differences in terms of types of inputs (facts vs texts), number of inputs, type and learning efficiency of the tested adaptation methods with respect to the number of parameters, and evaluation metrics (perplexity vs. factual MCQs accuracy). Yu et al. (2023) detailed the creation of a large and refined benchmark, specifically tailored to measure world knowledge within PLMs. Kasai et al. (2022) proposed a continual MCQ benchmark for world knowledge, updated every week with new questions about recent events extracted from news websites. Yang and Liu (2021) successfully used prefix-tuning to adapt a PLM for text classification, while Ma et al. (2022) used the same method for speech-to-text translation. Prefix-tuning was also shown to obtain good performance in natural language understanding (Lester et al., 2021), summarization (Chen et al., 2023) and sentiment analysis (Balakrishnan et al., 2022) *inter alia*. Zhao et al. (2022) showed that prefix-tuning may also be used for efficient domain adaptation.

Parameter-efficient training methods, such as LoRa and prefix-tuning, are often used both to continue pretrain an LLM and to adapt it to a domain. However, recent works suggest that, with LoRa and

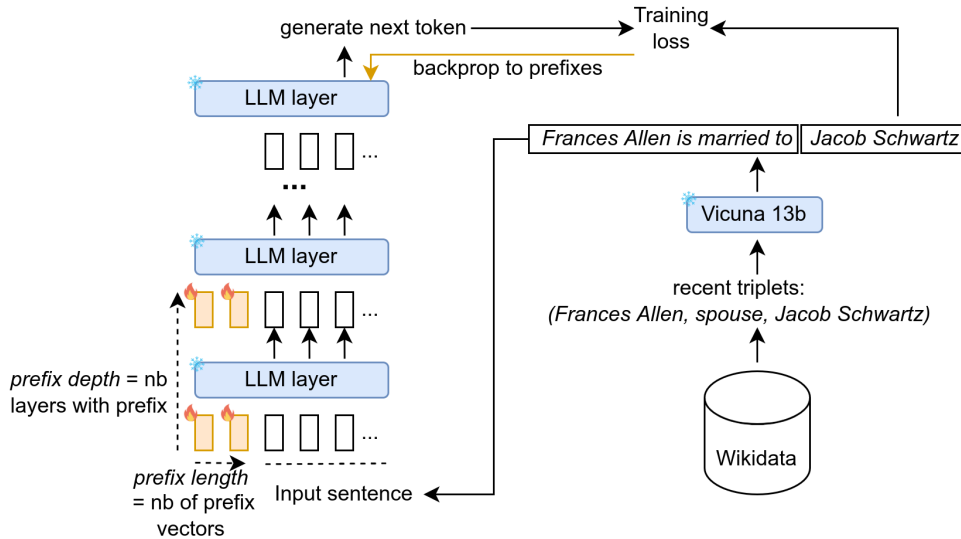


Figure 1: Proposed approach: new facts are extracted from Wikidata, transformed into sentences with Vicuna-13b and trained into prefixes. We claim and show that this architecture is better than LoRA to capture novel knowledge.

186 full finetuning, very few new factual knowledge are
 187 actually learned (Liu et al., 2024). We propose in
 188 this work to investigate this question with prefix-
 189 tuning, which is based on similar principles than
 190 in-context learning, a method that is known to be
 191 able to inject new knowledge. Compared to the
 192 past literature on prefix tuning, we focus on its
 193 properties with regard to factual knowledge learn-
 194 ing, and give concrete answers to the questions of
 195 whether and when does prefix tuning learn new
 196 factual knowledge.

197 3 Methodology

198 3.1 Research questions

199 As shown in the related works Section, there is still
 200 not a clear understanding about what is really learnt
 201 by finetuning methods like LoRA. In this study, we
 202 argue that prefix tuning is a better solution to inject
 203 a small number of new facts into the LLM, which
 204 may potentially be extended (in a future work) to
 205 support many facts either by retrieving the best
 206 prefix from a prefix-store (à la RAG), or by select-
 207 ing prefixes with gating networks (à la mixture-of-
 208 experts) or by generating prefixes with a dedicated
 209 model. Concretely, the target research questions
 210 of this work are: (i) Can a single prefix vector on
 211 the first layer learn a single fact? Does this learn-
 212 ing generalize to reformulations of this fact? (ii)
 213 Can a longer prefix ($n > 1$) learn multiple facts?
 214 What effect does prefix size have on learning and
 215 generalization? In-context learning suggests that
 216 the answer to this question and the previous one

217 are positive. (iii) In the existing literature, the pre-
 218 fix is usually spread across all layers of the model.
 219 However, Simoulin and Crabbé (2021) suggest that
 220 the deeper layers in Transformer models are as-
 221 sociated with abstract and high-level capabilities,
 222 while factual information is stored in the lower lay-
 223 ers. Does restricting the prefix depth d therefore
 224 affect the learning and generalization capacities of
 225 the model? (iv) Do the answers to the previous
 226 questions remain true with bigger models?

227 3.2 Facts learning

228 We model a fact as a semantic triple of the form
 229 (subject, predicate, object), in which the subject
 230 and object are typically noun phrases, and the pred-
 231 icate a verb phrase. We consider the following
 232 important properties, largely adapted from (Meng
 233 et al., 2022a):

234 **Learning:** The updated LLM has learnt the fact
 235 when it can predict the object from a sentence con-
 236 taining the subject and predicate after being up-
 237 dated, while it could not predict the object before;

238 **Generalization:** The LLM is able to generalize the
 239 learned fact when it can predict the object from a
 240 paraphrase of the subject and predicate.

241 **Specificity:** The updated LLM is specific when it
 242 correctly generates another expected object that is
 243 different from the learned triplet from a slightly
 244 different subject and predicate input.

245 **Non-forgetting:** The updated LLM generates the
 246 correct objects that were already known by the
 247 baseline LLM.

3.3 Evaluation

Let L be a baseline LLM and $T = T_1, \dots, T_p$ a list of recent facts (triples). We first build a training set containing a list of simple sentences generated from the triples in T (see Figure 1). We then update the model on this training set, either with prefix-tuning (our proposal) or LoRA (the baseline). The perplexity of the updated LLMs are computed on the same training set and compared: although it is largely debated in the community, we nevertheless consider that this perplexity is a relevant indicator of whether the LLM has learnt this training set or not. We then evaluate generalization by measuring the perplexity of the updated LLMs on complex, creative sentences created by reformulating the training sentences. We finally measure specificity and non-forgetting by evaluating the LLMs on existing MCQ benchmarks.

4 Dataset

In this section, we describe the steps used to create NOVEL-WD and give an overview of the resulting dataset. A sample output of each step of the full process is given in Table 1.

Element	Value
Triple	(Frances Allen, spouse, Jacob Schwartz)
Training sentence	Frances Allen is married to Jacob Schwartz.
Test sentence 1	Frances Allen’s spouse is
Test sentence 2	The spouse of Frances Allen was
Test sentence 3	Frances Allen was married to
Test sentence 4	Frances Allen has been married to
Test sentence 5	The name of Frances Allen’s spouse is
Question	Who was Frances Allen’s spouse?
Distractor 1	Charles Householder
Distractor 2	David Padua
Distractor 3	John Cocke

Table 1: A sample of the dataset for a single triple.

Triple extraction We begin by extracting RDF triples that were newly added to Wikidata. To do so, we retrieve new triples from a daily incremental database dump. We restrict ourselves to items and exclude lexemes, which represent lexicographical data. We also do not take into account complex triples, in which the subject or object is a Wikimedia template, as well as triples in which the subject is a numerical identifier, a filename or a URI. We then resolve eventual internal Wikidata links in the subject, predicate or object by replacing them with the English name of the associated item. Finally, when multiple triples share the same subject and predicate, we randomly select one such triple and discard the other ones, so as to limit the risk of models trying to learn multiple conflicting facts.

Training set To generate a training set, we convert each triple into a simple sentence, by querying a 8-bit quantized version of VICUNA-13B (Chiang et al., 2023) with a two-shots prompt (see code on github). For each triple, we generate one such sentence.

Two evaluation tasks The first evaluation is a causal language modeling task (perplexity): for each triple, we ask 8-bit VICUNA-13B in a two-shots setting to generate 5 sentences in which the object of the triple is missing. In order to test for generalization capabilities and to avoid repeating the training sentence, we specifically prompt *Vicuna* for "creative sentences". Manual editing may then be applied to the output sentences in the infrequent situation (occurring for less than 10 facts) where full sentences are generated rather than incomplete one.

The second task is a multiple choice question answering task (MCQ). For each triple, a two-shots 8-bit VICUNA-13B prompt is first applied to generate a question asking for the object of the triple. Then, a similar prompt is applied to generate 4 "likely answers" to the question. Among the 4 generated answers, we remove the ground-truth one if it is present, and select the 3 first remaining ones as distractors. After manually checking and editing the generated answers in rare cases (3 occurrences) where they semantically overlap, we then add in the correct answer. We therefore obtain a question with 4 possible choices, exactly one of which being correct.

After all the steps above have been applied, NOVEL-WD consists of 338 distinct triples, and each triple contains one associated training sentence, five incomplete validation sentences, one question and three distractors.

5 Experimental setup

The baseline model chosen for our experiments is BLOOMZ-7.1B (Muennighoff et al., 2023). The training was ran for up to 450 epochs using the AdamW optimizer with a weight decay of 0.1 and an initial learning rate of $3 * 10^{-2}$, decreasing by a factor of 10 after 10 epochs of non-decreasing training loss. We did not project the prefix through an intermediate MLP as mentioned in (Li and Liang, 2021), as we found that it did not increase training stability and generally resulted in lower performance. For all of our models, prefix-tuning was implemented by learning the value of the

previous key and value vectors in attention layers, resulting in two vectors per layer and per virtual token being learned, for a total of $2 * d * n$ vectors.

For each macro-experiment and number of facts k , we divided the $D=338$ facts of NOVEL-WD into non-fully overlapping subsets of length k , and trained one copy of the baseline model on each subset. For a given k , the number of subsets was computed as $\max(5, \lfloor D/k \rfloor)$. For example, for $k = 3$, we sampled 112 subsets of 3 facts, and trained a separate copy of BLOOMZ-7B1 on each of those 112 subsets. Training subsets were generated for values of k in $\{1, 2, 3, 4, 5, 8, 10, 20, 50, 100, 200\}$.

5.1 Evaluation

To evaluate our models in the text prediction setting, we prompt them with each of the five incomplete sentences associated with each fact from the training set, and generate the following ten tokens without sampling and with a temperature of 1. We only count an answer as correct if the model’s output contains the exact answer’s text, capitalization excepted, and we report the accuracy over every sentence of the test set for a given model. We also measure the **proportion of learning models** for a given k , by selecting only facts of the test set for which the baseline model does not output any correct prediction, and counting the proportions of the prefix-tuned models trained on those questions for which the test set accuracy is non-zero. In other words, learning models are models which are able to correctly predict at least one sentence completion for facts that were not known by the baseline.

To perform regression tests, we selected the SciQ (Welbl et al., 2017) and MMLU (Hendrycks et al., 2020a,b) datasets. For SciQ, we measure the accuracy of the baseline and prefix-tuned models in the MCQ setting, by using the same prompt as for NOVEL-WD, and selecting the lowest per-token perplexity choice. We apply this method on all 1,000 questions of the test set. For MMLU, we append each of the possible four completions to each sentence, and then select the one with the lowest per-token perplexity as the model’s answer. This is applied to the test sets from each of the 57 categories found in the dataset. Due to computational costs, regression tests were ran on a random sample of 5 prefix-tuned models for each value of k .

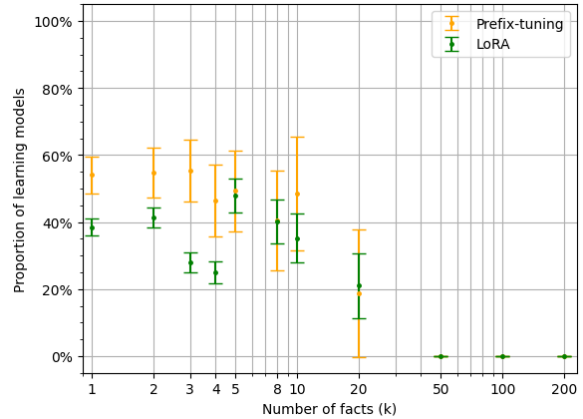


Figure 2: Percentage of prefix-tuned models obtaining increased accuracy over the baseline. Error bars span 95% confidence intervals.

6 Results and analysis

6.1 Base setup

Our initial experiment focuses on a single prefix ($n = 1, d = 1$), corresponding to 8,192 trainable parameters, or 0.000116% of the baseline model’s parameters. For comparison, we also perform the same experiment using LoRA (rank= 8, $\alpha = 8$) instead of prefix-tuning. We use the same training hyperparameters for both LoRA and prefix-tuning.

The proportion of prefix-tuned models with increased accuracy in the prediction setting is given in Figure 2, along with the mean accuracy (see Appendix B Figure 4) obtained in the prediction setting for different numbers of facts.

For $1 \leq k \leq 3$, between 54.1% and 55.4% of the models are able to learn at least one information over the baseline. This amount stays stable for $k \leq 10$, with the proportion of learning models ranging from 40.5% to 55.4%. For $k = 20$, this proportion drops to 18.8%, and none of the models trained for $k > 20$ achieved any accuracy gains over the baseline. Note that a recent work applying control theory to LLMs has shown that WikiText can be nearly perfectly predicted (at 97%) with less than 10 additional prompt tokens (Bhargava et al., 2024), which also somehow confirms from a different point of view this limit of $k \leq 10$ tokens than we have found.

The baseline model obtains a consistent accuracy ranging from 3.0% to 6.3%, suggesting that a small number of facts found in the dataset are either already known or easily deducible by the model. In contrast, the prefix-tuned models obtain a mean accuracy peaking at 29.1% for $k = 3$, and

gradually decreasing for $k > 3$ until $k = 50$, for which the results are no longer significantly better than the baseline. This initial result suggests that during training, the prefix is usually able to select and remember 1 to 3 facts well, and up to 20 with decreasing accuracy. Furthermore, this learning is conditional on having a low enough number of facts present in the training data; having more than 10 facts seems to hamper the model’s ability to learn even a single fact.

In comparison, models trained with LoRA systematically underperform prefix-tuned ones for all values of k , with a prediction accuracy reaching 20.4% for $k = 2$, and values ranging from 4.6% to 14.4% for other values of k . Furthermore, they typically obtain pLM scores that are similar or lower than the ones of prefix-tuned models. This may be due to the low rank value of 8 used in our experiments; however, rank 8 LoRA adds 3,932,160 parameters to the base model, a number which is 480 times higher than the parameters contained in a single prefix. We therefore argue that while LoRA may outperform prefix-tuning at higher matrix ranks, it does so in a much less cost-efficient manner than prefix-tuning.

6.1.1 Error analysis

With $k = 1$, about half of the facts found in NOVEL-WD were not learned by a single prefix. While we could not identify meaningful semantic or content differences between the types of facts that were learned and those that were not, we report in Table 6 in appendix A quantitative statistics between those two categories. For each reported statistic, the non-learned value was found to be significantly larger than the learned one, as measured using a one-sided Welch’s t-test ($p = 0.05$). This suggests that the facts that were not successfully learned are typically longer and are farther from the baseline model’s distribution, both in their sentence form and in the text completion setting, which might result in an inability for prefix-tuning to sufficiently steer the model towards learning them.

6.2 Detecting overfitting and forgetting

We report the training loss in Figure 3 and norm of the two prefix vectors in Appendix B Figure 5 measured post-training in each experiment.

We observe that for $k = 1$, almost all experiments end with a training loss approaching zero, with the exceptions of a few outliers for which

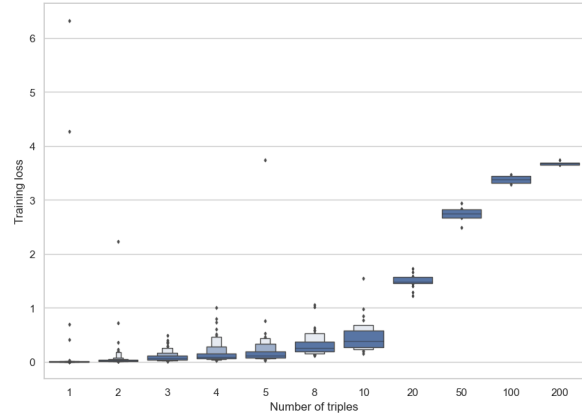


Figure 3: Training loss in the basic setup, measured post-training.

the loss remains high. This confirms our previous finding that the prefix is almost always able to learn a single fact, but may not be able to generalize in the prediction setting. When increasing k , the losses increase linearly up to $k = 10$ (median value: $L_{train} = 0.38$). For $n \geq 20$, the loss increases sharply and quickly approaches the baseline model’s loss of 4.38. We interpret this inflection as consistent with our previous observations, suggesting that a change of learning mode occurs in the vicinity of $k = 15$: For lower values, the model is efficiently able to learn and generalize novel information, while for higher values, the model may no longer be able to store all facts and instead unsuccessfully attempt to learn a combined representation of the training set. These findings are also consistent with the evolution of the prefix norm given: For $n \leq 3$, we observe a linear increase in prefix norm, which may indicate that the model does not make full use of the available prefix capacity. For $3 \leq n \leq 10$, the prefix norm is nearly constant and may signal increasing compression within the prefix. Finally, for $n \geq 10$, the prefix norm decreases rapidly.

Finally, we report in Table 2 the results of the evaluation over SciQ and MMLU, which shows that the prefix-tuned models do not seem to forget facts learned during pre-training or incur any loss of reasoning capabilities, for any value of k . Surprisingly, our prefix-tuned models even perform consistently and significantly better than the baseline for all values of k . Our hypothesis is that, by "finetuning" (through a prefix) the LLM on Wikipedia-like sentences, we specialize the LLM to interpret its inputs in a more "factual way" and in the Wikipedia domain, which is useful for the type of factual

k	SciQ acc.	MMLU acc.		
		Min	Max	Avg
Baseline	0.757	0.130	0.463	0.307
1	0.833	0.184	0.512	0.343
2	0.864	0.189	0.517	0.341
3	0.840	0.189	0.517	0.340
4	0.838	0.184	0.517	0.339
5	0.827	0.191	0.509	0.339
8	0.833	0.184	0.509	0.341
10	0.834	0.193	0.509	0.341
20	0.808	0.185	0.515	0.328
50	0.835	0.190	0.518	0.335
100	0.826	0.192	0.512	0.340
200	0.828	0.189	0.524	0.342

Table 2: Accuracy of the models on the MMLU and SciQ datasets, averaged over 5 random runs for each value of k . For MMLU, we report the score obtained by the lowest and highest accuracy as well as the average across categories.

MCQ questions that occur in SciQ and MMLU. However, we did not study this hypothesis in detail and leave this question open for future work.

6.3 Impact of prefix size

Table 3 contains the results obtained when prefix-tuning BLOOMZ-7B1 while varying the number of virtual tokens n contained in the prefix.

k	n=1		n=20		n=100	
	Acc	pLM	Acc	pLM	Acc	pLM
1	0.274	0.541	0.353	0.601	0.365	0.619
2	0.279	0.548	0.333	0.613	0.357	0.607
3	0.291	0.554	0.315	0.589	0.358	0.616
4	0.247	0.464	0.321	0.607	0.337	0.619
5	0.227	0.493	0.316	0.582	0.304	0.612
8	0.177	0.405	0.256	0.524	0.270	0.452
10	0.159	0.485	0.245	0.601	0.268	0.512
20	0.123	0.188	0.199	0.500	0.218	0.500
50	0.076	0	0.116	0.167	0.113	0.167
100	0.053	0	0.086	0.400	0.096	0.400
200	0.055	0	0.063	0	0.070	0

Table 3: Proportion of learning models (pLM) and mean prediction accuracy for different number of virtual tokens n in the prefix. Bold values denote statistically significant improvements over $n = 1$, using a one-sided z-test for proportions for pLM and a one-sided t-test for the accuracy ($p = 0.05$).

We observe significant improvement in accuracy for nearly all values of k when increasing the prefix size from 1 to 20, as well as significant gains in the proportion of learning models for $k \in \{1, 4, 20, 100\}$. Similar results are obtained when further increasing the prefix size from 1 to 100. However, none of the variation in accuracy or proportion of learning models between $n = 20$ and $n = 100$ are statistically significant.

We interpret those results as follows: Increasing the prefix size only modestly increases the chances for a model to be able to learn at least one fact. However, such an increase has a strong impact on the prediction capabilities of the model, which suggests that the model is able to learn more facts and to generalize better.

We hypothesize that the former may stem from the varying complexity of the facts in our dataset: for some facts, the base model may already contain information about the subject and predicate, and prefix-tuning might only be needed to learn the value of the object. A typical example of this situation can be found in facts of the type "[historical figure] was born on [date]". On the contrary, there exist more complex facts for which the subject and predicate themselves might be novel, and for which the base model might not contain information. We also note that increasing the prefix size past 20 brings no further improvement to the learning and generalization capacities of our model, which may indicate that prefixes are inherently limited in terms of information capacity.

6.4 Impact of prefix depth

We report in Table 4 the results obtained by increasing the number of layers spanned by the prefix in our initial setup from $d = 1$ (minimal depth) to $d = 30$ (full-depth prefix).

k	d=1		d=30	
	Acc	pLM	Acc	pLM
1	0.274	0.541	0.354	0.590
2	0.279	0.548	0.441	0.667
3	0.291	0.554	0.520	0.768
4	0.247	0.464	0.467	0.690
5	0.227	0.493	0.470	0.731
8	0.177	0.405	0.487	0.690
10	0.159	0.485	0.476	0.789
20	0.123	0.188	0.401	0.813
50	0.076	0	0.275	0.333
100	0.053	0	0.130	0.800
200	0.055	0	0.101	0.000

Table 4: Proportion of learning models (pLM) and mean prediction accuracy for different prefix depths d in the prefix. Bold values denote statistically significant improvements over $d = 1$, using a one-sided z-test for proportions for pLM and a one-sided t-test for the accuracy ($p = 0.05$).

We observe that increasing the prefix depth has a significant effect on both the accuracy and the proportion of learning models. For all values of k , the average accuracy is increased by 8 to 31%, with the highest increase reached for $k = 10$. The highest average accuracy is obtained for $k = 3$, which

once more suggests that up to three facts can be efficiently stored within a prefix, but performance stays comparable up to $k = 10$.

The second main observation is the fact that the proportion of learning models significantly increases for all values of k except $k = 1$, with gains of up to 80% for $k = 100$. We hypothesize that increasing the prefix depth allows for more complex information to be learned and enables the model to learn at least one information for all but the highest amount of training facts. Increasing the value of d from 1 to 30 effectively multiplies the number of trainable parameters by 30, but far surpasses the results obtained by increasing the prefix length by a factor of 100. We therefore remark that prefix depth seems to have a much stronger effect on model performance than prefix length.

6.5 Impact of base model

To investigate the effect that the type and size of the base model may have on prefix-tuning, we repeat our initial experiments on BLOOMZ-1B7, the 1.7 billion parameter version of BLOOMZ, chosen for scale comparisons. We measure the accuracy of the baseline models in the prediction setting over the entirety of NOVEL-WD. BLOOMZ-1B7 obtained an overall accuracy of 4.4%, while BLOOMZ-7B1 reached a similarly low value of 5.0%.

The results obtained after prefix-tuning are reported in Table 5. In terms of scaling, we first

k	BLOOMZ-1B7		BLOOMZ-7B1	
	Acc	pLM	Acc	pLM
1	0.293	0.565	0.274	0.541
2	0.273	0.556	0.279	0.548
3	0.262	0.589	0.291	0.554
4	0.213	0.464	0.247	0.464
5	0.189	0.403	0.227	0.493
8	0.152	0.286	0.177	0.405
10	0.112	0.394	0.159	0.485
20	0.085	0.189	0.123	0.188
50	0.053	0	0.076	0
100	0.045	0	0.053	0
200	0.039	0	0.055	0

Table 5: Proportion of learning models (pLM) and mean prediction accuracy for different number of virtual tokens n in the prefix. Bold values denote statistically significant improvements over the previous column, using a one-sided z-test for proportions for pLM and a one-sided t-test for the accuracy ($p = 0.05$).

note that there are no significant improvements in terms of the proportion of learning models between BLOOMZ-1B7 and BLOOMZ-7B1. This strengthens the intuition that this may be due to the

inherent complexity of some facts in the dataset, and to the fact that the ability to learn a fact is already present in smaller models. However, increasing the model size has a noticeable effect on the prediction accuracy, which increases by several percentage points for $k \in \{4, 5, 10, 20, 50\}$. We believe that this is partially due to the scaling generalization capabilities of the models. However, as the number of trainable parameters almost doubles between BLOOMZ-1B7 and BLOOMZ-7B1, these improvements may also be explained by an increase in prefix capacity.

Finally, to give an idea of the extracted facts, the quality of the synthetic generated sentences and which facts are correctly classified by the baseline model, Table 7 in Appendix C shows a random extract of known facts and generated sentences: some facts may "leak" from the LLM pretraining corpus (e.g., *Frederik Storm in Denmark*), or may be guessed (e.g., *Vitale Faliero, language spoken, Italian*) or may be answered by chance (e.g., *A View to a Kill, MPA rating, PG*). This question of leakage vs. actual forecasting is discussed in more details in (Halawi et al., 2024).

7 Conclusion

In this study, we have developed a dataset for novel fact learning in pre-trained language models. We have shown that prefix-tuning can be used to learn new facts, and investigated the effect of various factors on prefix-tuning performance. Our main recommendation is to use full-depth prefixes, but to limit the prefix length to 20 virtual tokens.

We see several major avenues for future research based on this work. While we measured the effect of different factors independently, their combined effect might be different. In particular, it is hard to predict how prefix length and depth may interact together. Another research direction is the use of different and more recent baseline architectures such as Mixtral (Jiang et al., 2024). Finally, a long-term goal could be to scale our approach to larger datasets, for example by using a mixture of prefixes at capacity along with a routing module. This could allow the use of a small, regular stream of new information to continually update a model. The code used to create NOVEL-WD and perform our experiments can be found on GitHub.

8 Limitations

While this paper addresses the challenge of updating LLMs with novel facts, there are other types of "updates" that should be achieved to make the updated LLM as useful as a new LLM pretrained from scratch on an up-to-date corpus, such as language and topic drifts. The method described in this work can not solve this issue. More generally, representing knowledge with triples is very limited, and can hardly for instance encode time-dependent and location-dependent cultural preferences, common sense and beliefs. This work is thus strongly limited in terms of the type of knowledge it can capture, but it is only a first step towards a more general LLM updating paradigm.

Another limitation is that only a few facts are injected in the LLM with our method, while continual updating of the LLM would require a constantly increasing number of facts to be added. To achieve this, our method would require an additional step to select or generate the appropriate prefixes, depending on the observed context, in a similar way as what is done with RAG or alternatively mixture of experts. We have not tested in this work such an enhancement, and we have only focused so far on studying the usefulness of prefix tuning as an alternative to RAG and LoRA.

Finally, an apparent limitation may be the size of NOVEL-WD, which is quite small. However, this is mainly because of the high cost of running the large number of experiments required in this study. However, since 2020, Wikidata grows at a rate of 7 million entities per year (see https://en.wikisource.org/wiki/Wikidata:The_Making_Of), and the filtering that we apply leads to about 32000 remaining new facts per day (as checked for 14th March 2024), so getting data at scale should not be an issue. Furthermore, although we made a few manual interventions to check for generation errors when creating the dataset and benchmarks, we are convinced such interventions could be avoided when using better LLM, such as Llama3-70b or Qwen-72b.

References

Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. 2022. *Prompting as Probing: Using Language Models for Knowledge Base Construction*. Publisher: arXiv Version Number: 3.

- Sudhandar Balakrishnan, Yihao Fang, and Xiaodan Zhu. 2022. *Exploring Robustness of Prefix Tuning in Noisy Data: A Case Study in Financial Sentiment Analysis*. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 78–88, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Aman Bhargava, Cameron Witkowski, Manav Shah, and Matt Thomson. 2024. *What’s the magic word? a control theory of llm prompting*. *Preprint*, arXiv:2310.04444.
- Sebastian Borgeaud, A. Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, J. Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. Hennigan, Saffron Huang, Lorenzo Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, G. Irving, Oriol Vinyals, Simon Osindero, K. Simonyan, Jack W. Rae, Erich Elsen, and L. Sifre. 2021. *Improving language models by retrieving from trillions of tokens*.
- Chen Chen, Wei Emma Zhang, and Alireza Seyed Shakeri. 2023. *Incorporating Knowledge into Document Summarization: an Application of Prefix-Tuning on GPT-2*. *arXiv preprint*. ArXiv:2301.11719 [cs].
- Cheng-Han Chiang, Yung-Sung Chuang, and Hung-yi Lee. 2022. *Recent Advances in Pre-trained Language Models: Why Do They Work and How Do They Work*. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 8–15, Taipei. Association for Computational Linguistics.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality* | LMSYS Org.
- Andrew M Dai and Quoc V Le. 2015. *Semi-supervised Sequence Learning*. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. *A Mathematical Framework for Transformer Circuits*.
- R. M. French. 1999. *Catastrophic forgetting in connectionist networks*. *Trends in Cognitive Sciences*, 3(4):128–135.

744	Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories . In <i>Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing</i> , pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.	798
745		799
746		800
747		801
748		802
749		803
750		804
751	Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. In <i>Proceedings of the 37th International Conference on Machine Learning</i> , volume 119 of <i>ICML '20</i> , pages 3929–3938. JMLR.org.	805
752		806
753		807
754		808
755		809
756		810
757	Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. 2024. Approaching human-level forecasting with language models . <i>Preprint</i> , arXiv:2402.18563.	811
758		812
759		813
760		814
761	Junxian He, Chunting Zhou, Xuezhe Ma, Taylor BergKirkpatrick, and Graham Neubig. 2021. Towards a Unified View of Parameter-Efficient Transfer Learning .	815
762		816
763		817
764		818
765	Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2020a. Aligning AI With Shared Human Values .	819
766		820
767		821
768	Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020b. Measuring Massive Multitask Language Understanding .	822
769		823
770		824
771		825
772	Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification . In <i>Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 328–339, Melbourne, Australia. Association for Computational Linguistics.	826
773		827
774		828
775		829
776		830
777		831
778	Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models .	832
779		833
780		834
781		835
782	Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models . In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 6237–6250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	836
783		837
784		838
785		839
786		840
787		841
788		842
789		843
790		844
791	Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards Continual Knowledge Learning of Language Models .	845
792		846
793		847
794		848
795	Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L�lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th�ophile Gervet, Thibaut Lavril, Thomas Wang, Timoth�e Lacroix, and William El Sayed. 2024. Mixtral of experts . <i>Preprint</i> , arXiv:2401.04088.	849
796		850
797		851
	Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know . <i>arXiv preprint</i> . ArXiv:2207.05221 [cs].	852
		853
		854
		855
		856
	Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2022. RealTime QA: What’s the Answer Right Now? <i>arXiv preprint</i> . ArXiv:2207.13332 [cs].	
	James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks . <i>Proceedings of the National Academy of Sciences</i> , 114(13):3521–3526. Publisher: Proceedings of the National Academy of Sciences.	
	Jannik Kossen, Tom Rainforth, and Yarin Gal. 2023. In-Context Learning in Large Language Models Learns Label Relationships but Is Not Conventional Learning . <i>arXiv preprint</i> . ArXiv:2307.12375 [cs].	
	Andrei Kuchavay, Zachary Schillaci, Lo�c Mar�chal, Maxime W�rsch, Ljiljana Dolamic, Remi Sabonadiere, Dimitri Percia David, Alain Mermoud, and Vincent Lenders. 2023. Fundamentals of Generative Large Language Models and Perspectives in Cyber-Defense .	
	Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning . In <i>Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing</i> , pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.	
	Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K�ttler, Mike Lewis, Wen-tau Yih, Tim Rockt�schel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks .	

857	Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation . In <i>Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)</i> , pages 4582–4597, Online. Association for Computational Linguistics.	915
858		916
859		917
860		918
861		919
862		920
863		
864		
865	Bill Yuchen Lin, Sida Wang, Xi Lin, Robin Jia, Lin Xiao, Xiang Ren, and Scott Yih. 2022a. On Continual Model Refinement in Out-of-Distribution Data Streams . In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 3128–3139, Dublin, Ireland. Association for Computational Linguistics.	921
866		922
867		923
868		924
869		925
870		926
871		927
872	Stephanie Lin, Jacob Hilton, and Owain Evans. 2022b. Teaching Models to Express Their Uncertainty in Words . <i>Transactions on Machine Learning Research</i> .	928
873		929
874		930
875		931
876	James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. 2024. Bitdelta: Your fine-tune may only be worth one bit . <i>Preprint</i> , arXiv:2402.10193.	932
877		933
878		934
879	Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022a. Generated Knowledge Prompting for Commonsense Reasoning . In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.	935
880		936
881		937
882		938
883		939
884		940
885		941
886		942
887	Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023a. RETA-LLM: A Retrieval-Augmented Large Language Model Toolkit .	943
888		944
889		945
890		946
891	Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing . <i>ACM Computing Surveys</i> , 55(9):195:1–195:35.	947
892		948
893		949
894		950
895		951
896	Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks . In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)</i> , pages 61–68, Dublin, Ireland. Association for Computational Linguistics.	952
897		953
898		954
899		955
900		956
901		957
902		958
903		959
904	Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too . <i>arXiv preprint</i> . ArXiv:2103.10385 [cs].	960
905		961
906		962
907		963
908	Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models . In <i>Findings of the Association for Computational Linguistics: ACL 2022</i> , pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.	964
909		965
910		966
911		967
912		968
913		969
914		970
		971
	Yukun Ma, Trung Hieu Nguyen, and Bin Ma. 2022. CPT: Cross-Modal Prefix-Tuning for Speech-To-Text Translation . In <i>ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)</i> , pages 6217–6221. ISSN: 2379-190X.	
	Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and Editing Factual Associations in GPT . <i>Advances in Neural Information Processing Systems</i> , 35:17359–17372.	
	Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-Editing Memory in a Transformer . <i>arXiv preprint</i> . ArXiv:2210.07229 [cs].	
	Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2021. Fast Model Editing at Scale .	
	Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual Generalization through Multitask Finetuning . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.	
	Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In <i>Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)</i> , pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.	
	Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners .	
	Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabza, Mike Lewis, and Amjad Almahairi. 2023. Progressive Prompts: Continual Learning for Language Models . <i>arXiv preprint</i> . ArXiv:2301.12314 [cs].	
	Antoine Simoulin and Benoit Crabbé. 2021. How Many Layers and Why? An Analysis of the Model Depth in Transformers . In <i>Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop</i> , pages 221–228, Online. Association for Computational Linguistics.	
	Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and Policy Considerations for Modern Deep Learning Research . <i>Proceedings</i>	

of the AAAI Conference on Artificial Intelligence, 34(09):13693–13696. Number: 09.

Jianing Wang, Wenkang Huang, Minghui Qiu, Qihui Shi, Hongbin Wang, Xiang Li, and Ming Gao. 2022. [Knowledge Prompting in Pre-trained Language Model for Natural Language Understanding](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3177, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. 2023. [Simple synthetic data reduces sycophancy in large language models](#). *arXiv preprint*. ArXiv:2308.03958 [cs].

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. [Crowdsourcing Multiple Choice Science Questions](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2021. [Memorizing Transformers](#).

Yuxiang Wu, Yu Zhao, Baotian Hu, Pasquale Minervini, Pontus Stenertorp, and Sebastian Riedel. 2022. [An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks](#). pages 5184–5196.

Zonghan Yang and Yang Liu. 2021. [On Robust Prefix-Tuning for Text Classification](#).

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2018. [Lifelong Learning with Dynamically Expandable Networks](#).

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2023. [KoLA: Carefully Benchmarking World Knowledge of Large Language Models](#). *arXiv preprint*. ArXiv:2306.09296 [cs].

Lulu Zhao, Fujia Zheng, Weihao Zeng, Keqing He, Weiran Xu, Huixing Jiang, Wei Wu, and Yanan Wu. 2022. [Domain-Oriented Prefix-Tuning: Towards Efficient and Generalizable Fine-tuning for Zero-Shot Dialogue Summarization](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4848–4862, Seattle, United States. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. [Modifying Memories in Transformer Models](#).

A Learned and non-learned facts

Table 6 gives some statistics about the facts that have been learned or not learned in our experiments.

B Impact from the number of novel facts

Figure 4 complements Figure 2 by showing the mean accuracy of the models as a function of the number of facts, confirming the diminishing returns when increasing the number of new facts beyond 10.

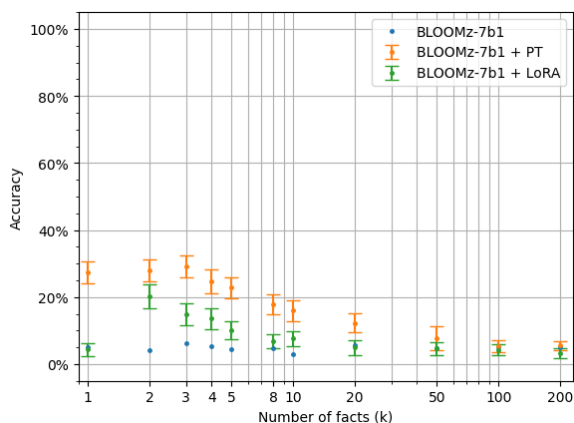


Figure 4: Mean accuracy of prefix-tuned (PT) models, LoRA models and of the baseline (right) in the prediction setting. Error bars span 95% confidence intervals.

Figure 5 complements Figure 3 by showing the observation of two phases with less and more than 10 new facts.

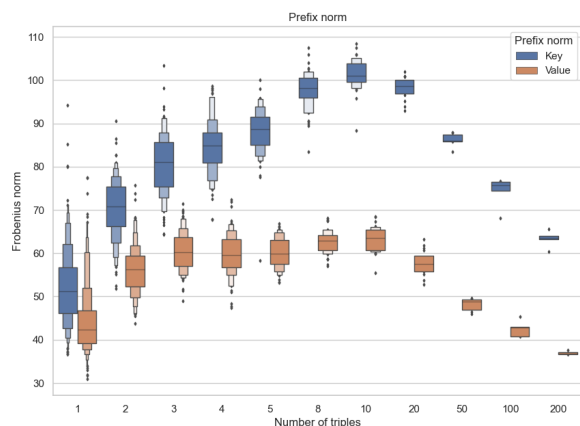


Figure 5: Frobenius norm of the key and value vectors of the prefix in the basic setup, measured post-training.

C Qualitative examples

Table 7 shows both examples of generated sentences and facts that are already known by the

Metric	Train set facts		Test set facts	
	Non-learned	Learned	Non-learned	Learned
Length (characters)	57.8	51.0	73.5	66.2
Length (tokens)	15.5	13.3	18.2	15.9
Length of <i>o</i> (characters)	17.8	15.6	-	-
BLOOMZ-7B1 per-token perplexity	4.56	4.30	4.26	4.18

Table 6: Quantitative comparison of the facts of NOVEL-WD that were successfully learned and those which were not within a single prefix. Reported values are averaged per category.

1043 model. All of these samples have been randomly
1044 extracted, without any cherry picking.

The Lesser hairy-footed dunnart is also known as *S. youngsoni*.
Milady de Winter died by homicide.
Garden Warbler is also known as *S. borin*.
Dylan and Cole Sprouse were born on 4 August 1992.
Yannick Aguemon is 180 centimetres tall.
Heinrich Hoffmann died of natural causes.

Chen Lin, occupation, writer
White Flag, language of work or name, English
A View to a Kill, MPA rating, PG
Corey Hart, language spoken, English
Extinction, mitigated by, conservation efforts
Frederik Storm, country for sport, Denmark

Table 7: Random samples of generated sentences (top) and "already known" facts (bottom)