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

Maxime Méloux, Christophe Cerisara

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

Maxime Méloux[1,2], **Christophe Cerisara**[1, 3]
[1]Loria / Vandœuvre-lès-Nancy, France
[2]Université de Lorraine / Nancy, France
[3]CNRS / Vandœuvre-lès-Nancy, France
maxime.meloux@protonmail.com, christophe.cerisara@loria.fr

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, are often prone to catastrophic forgetting, and most existing benchmarks focus on task adaptation rather than acquiring new information. 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 beyond the dataset itself, we release a procedure to build again later on new versions of similar datasets with up-to-date information. In a second part, we 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 capacity of the prefix increases with its length and with the base model size.

Introduction

Since the introduction of the Transformers architecture in Vaswani et al. (2017), pre-trained language models (PLMs) have become the de facto standard for most natural language processing tasks (Chiang, Chuang, and Lee 2022). Since those models are typically trained in a semi- or self-supervised setting, adaptations such as fine-tuning are required to adapt them to downstream tasks (Dai and Le 2015; Howard and Ruder 2018; Radford et al. 2019).

In addition to requiring task-specific adaptation, large language models are usually unknowing of recent events or novel world knowledge which is not contained in their training set (Alivanistos et al. 2022; Kucharavy et al. 2023). As a result, applications that rely on up-to-date and accurate knowledge require further updates or adaptation of PLMs after their initial training.

More generally, the question arises as to how models can be taught novel factual knowledge, and how to evaluate the effectiveness of the adaptation.

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, Ganesh, and McCallum 2020). Furthermore, Wei et al. (2023) showed that when fine-tuning a model on a small corpus with new information, the model instead learns to generate previously unseen information (hallucinations), and that once learned, this behavior is then repeated for other information. Fine-tuning may therefore not enable a model to learn new information that was not seen during training.

Parameter-efficient fine-tuning (PEFT) methods have emerged as an 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). Information can be stored using in-context learning (Logan IV et al. 2022), prompting (Liu et al. 2023b) or prompt-tuning (Lester, Al-Rfou, and Constant 2021) amongst others. PEFT methods have recently experienced a surge of popularity[1].

In this study, we focus on prefix-tuning (Li and Liang 2021), a parameter-efficient fine-tuning method in which the pre-trained model parameters are kept frozen, but a small, continuous vector called the *prefix* is 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

[1]As of August 12, 2023, the `peft` Python package has been downloaded over 430,000 times during the last month.

of factual information. This is backed by the findings of Kossen, Rainforth, and Gal (2023), which argues 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.

The main contributions of this paper are as follows:

- We introduce NOVEL-WD, a framework for generating new, curated datasets and benchmarks of novel/rare information extracted from Wikidata in order to evaluate the learning of new facts in PLMs.
- We evaluate the performance of prefix-tuning for novel knowledge acquisition, and measure the extent to which prefix-tuning can efficiently compress information in different situations.

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. (2022) 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 intepretability. Geva et al. (2021) showed that the feedforward layers of Transformers models act similarly to key-value memories in the context of 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, sometimes 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, Hilton, and Evans (2022) focused on measuring model uncertainty as a way to distinguish properly known facts from hallucinated ones. Jang et al. (2021, 2022) introduced the framework TEMPORALWIKI, which includes a process to generate datasets and benchmarks from information extracted from Wikipedia and aligned with Wikidata triples, with the goal of evaluating the performance of models on new factual knowledge.. 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 to the new task of text classification, while Ma, Nguyen, and Ma (2022) used the same method for speech-to-text translation. Prefix-tuning was also shown to obtain good performance in natural language understanding (Lester, Al-Rfou, and Constant 2021), summarization (Chen, Zhang, and Shakeri 2023) and sentiment analysis (Balakrishnan, Fang, and Zhu 2022) *inter alia*. Zhao et al. (2022) showed that prefix-tuning may also be used for efficient domain adaptation.

The main difference between NOVEL-WD and the datasets found in TEMPORALWIKI lies in the scope and intended use of our dataset. NOVEL-WD is constructed from Wikidata alone, and uses synthetic data for the training and evaluation tasks, therefore limiting the need to download and process large Wikipedia dump files. We intend our approach to be particularly useful to generate a moderate amount of data at a high frequency. Furthermore, we include two evaluation tasks in the form of sentence completion and MCQ. Finally, our training set contains minimal training sentences for each fact of our dataset rather than a larger snippet of text, in order to easily compare the learning capabilities of different models and adaptation techniques.

To our knowledge, no previous studies have been conducted on the use of prefix-tuning to learn novel information, nor on quantifying information storage inside the prefix.

Methodology

Research question

In this study, we would like to investigate the following questions:

- Can a simple prefix (i.e. a prefix with a number of virtual tokens of $n = 1$ and a depth of $d = 1$) learn a single fact? Does this learning generalize to reformulations of this fact?
- Can a larger prefix ($n > 1$) learn multiple facts? What effect does prefix size have on learning and generalization? In-context learning suggests that the answer to this question and the previous one are positive.
- In the existing literature, the prefix is usually spread across all layers of the model. However, recent work

(Simoulin and Crabbé 2021) suggests that the deeper layers in Transformer models are associated with abstract and high-level capabilities, while factual information is stored in the lower layers. Does restricting the prefix depth d therefore affect the learning and generalization capacities of the model?

- Can training metrics be used to indicate overfitting and forgetting in a prefix-tuned model at training time?
- Do the answers to the previous questions remain true when model scales? The assumption is that a smaller prefix may be required when the model is bigger thanks to the extra information the model already has.

Information learning

We model a fact as a semantic triple of the form (subject, predicate, object), in which the subject and object are typically noun phrases, and the predicate a verb phrase. Given a baseline language model L, a modified language model L' and a triple $T = (s, p, o)$, we consider the following aspects, largely adapted from Meng et al. (2022a):

- Learning: Given a sentence or question S containing s and p and expecting the continuation o, we consider that L has learned T if $L(S) \neq o$ and $L'(S) = o$.
- Generalization: Given a sentence or question $\hat{S} \approx S$, we consider that L' can generalize if $L'(\hat{S}) = o$.
- Specificity: Given a second triple $T' = (s', p', o')$ such that $o \neq o'$ and a sentence or question S' expecting the continuation o', we consider that L' is specific if $L'(S) = o$ and $L'(S') \neq o$.
- Non-forgetting: We consider that L' has retained the fact contained in T if $L(S) = o$ and $L'(S) = o$.

Evaluation

Given a baseline model L and a list $T = T_1, ..., T_p$ of facts encoded as triples of the form $T_i = (s_i, p_i, o_i)$, we evaluate the efficiency of an adaptation technique by applying the following approach:

- We apply the adaptation technique on L, using as training set a dataset D containing a list of simple sentences previously generated from the triples of T.
- We evaluate learning in the resulting model L', by measuring its perplexity in a causal language modeling setting on the sentences of D, and comparing it to that of L.
- We evaluate generalization by measuring the perplexity of L' in a causal language modeling setting on complex, creative sentences created by reformulating the sentences of D.
- We measure specificity and non-forgetting by evaluating L and L' on existing MCQ benchmarks.

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)
Train sent.	Frances Allen is married to Jacob Schwartz.
Test sent. 1	Frances Allen's spouse is
Test sent. 2	The spouse of Frances Allen was
Test sent. 3	Frances Allen was married to
Test sent. 4	Frances Allen has been married to
Test sent. 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. In order to do so, we query a 8-bit quantized version of VICUNA-13B (Chiang et al. 2023) with a two-shot prompt. For each triple, we generate one such sentence.

Test sets

To evaluate the performance of models on our dataset, we provide two different evaluation tasks.

The first task is a causal language modeling task: For each triple, we used 8-bit VICUNA-13B in a two-shot setting to generate five 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 prompted the model for "creative sentences". Manual editing was then applied to the output sentences in the infrequent situation (occurring for fewer than 10 facts) where full sentences were generated rather than an incomplete one.

The second task is a question answering task in the multiple choice question setting (MCQ). For each triple, a two-shot 8-bit VICUNA-13B prompt was first applied to generate a question asking for the object of the triple. Then, a similar prompt was applied to generate four "likely answers" to the question. Among the four generated answers, we remove the ground-truth one if it is present, and select the three 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 four possible choices, exactly one of which being correct.

Final dataset

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.

Experimental setup

The baseline model chosen for our experiments is the 7.1-billion parameter version of BLOOMZ (Muennighoff et al. 2023), BLOOMZ-7B1. 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 $\max(5, \lfloor D/k \rfloor)$. The resulting number of experiments can be found in Table 2. 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. Due to the restricted dataset size, there exists significant overlap between the training sets of experiments when $k \geq 50$.

k	1	2	3	4	5	8
Subsets	338	169	112	84	67	42
k	10	20	50	100	200	Total
Subsets	33	16	6	5	5	877

Table 2: The number of subsets for each number of facts. In each macro-experiment, one model was trained on each split.

For consistency, during the evaluation phase, we only evaluated a model using the sentences and questions of NOVEL-WD corresponding to the k triples it had been trained on.

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, Liu, and Gardner 2017) and MMLU (Hendrycks et al. 2020a; Hendrycks et al. 2020b) 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.

Results and analysis

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.

The proportion of prefix-tuned models with increased accuracy in the prediction setting is given in Figure 1, and Figure 2 contains the mean accuracy obtained in the prediction setting for different numbers of facts.

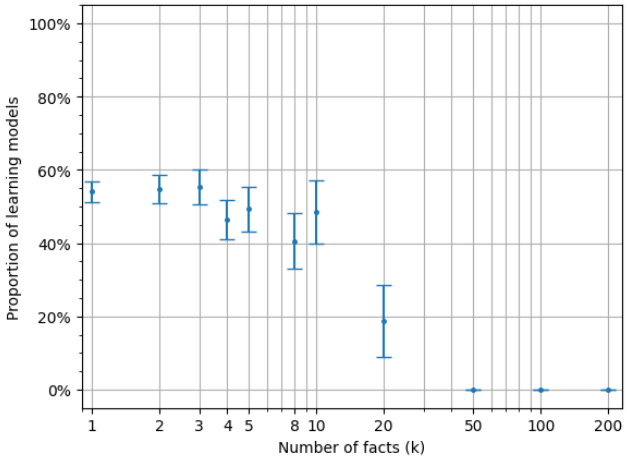

Figure 1: Percentage of prefix-tuned models obtaining increased accuracy over the baseline in the prediction setting, with error bars spanning 95% confidence intervals.

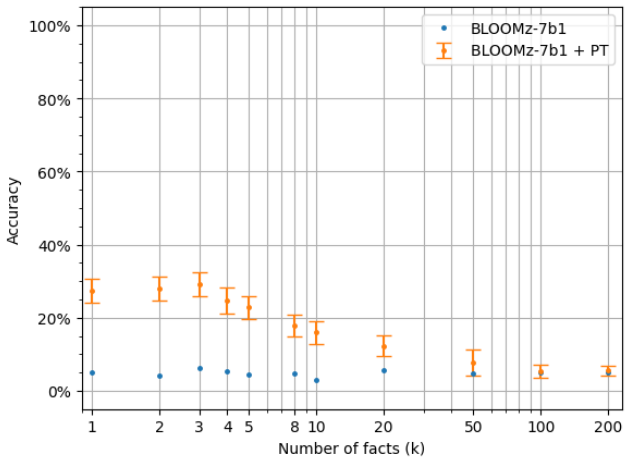

Figure 2: Mean accuracy of prefix-tuned (PT) models and of the baseline in the prediction setting, with error bars spanning 95% confidence intervals.

For $1 \leq k \leq 3$, between 54.1% and 55.4% of the models are successfully able to learn at least one information over the baseline. This amount stays relatively consistent 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.

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.

Error analysis In the case of $k = 1$, close to half of the facts found in NOVEL-WD were not successfully learned by a single prefix. While we could not identify any meaningful semantic or content differences between the types of facts that were learned and those that were not, we report in Table 3 quantitative statistics between those two categories. For each reported statistic, the NL value was found to be significantly larger than the L 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.

	Train set		Test set	
Metric	NL	L	NL	L
Length (characters)	57.8	51.0	73.5	66.2
Length (tokens)	15.5	13.3	18.2	15.9
Length of o (characters)	17.8	15.6	-	-
BLOOMZ-7B1 per-token ppl	4.56	4.30	4.26	4.18

Table 3: Quantitative comparison of the facts of NOVEL-WD that were successfully learned (L) and those which were not (NL) within a single prefix. Reported values are averaged per category.

Detecting overfitting and forgetting

We report in Figure 3 the final training loss of each experiment, and in Figure 4 the norm of the two vectors contained in the prefix at the end of the training phase of each experiment.

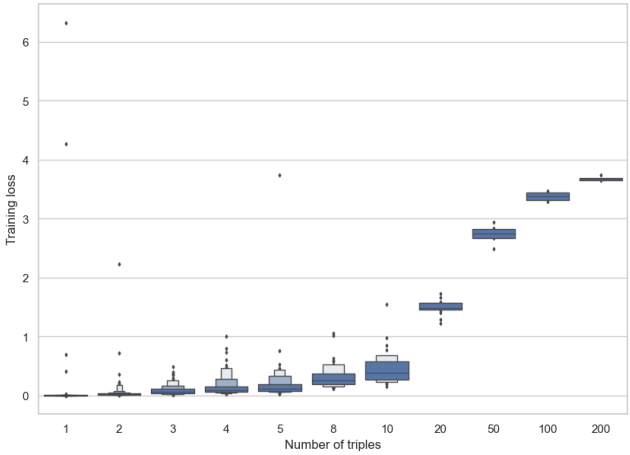

Figure 3: Final training loss of each experiment in the basic setup.

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 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, we hypothesize that the model is no longer able to store all facts and instead unsuccessfully attempts to learn a combined representation of the training set. These findings are also consistent with the evolution of the prefix norm given: For $1 \leq n \leq 3$, 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

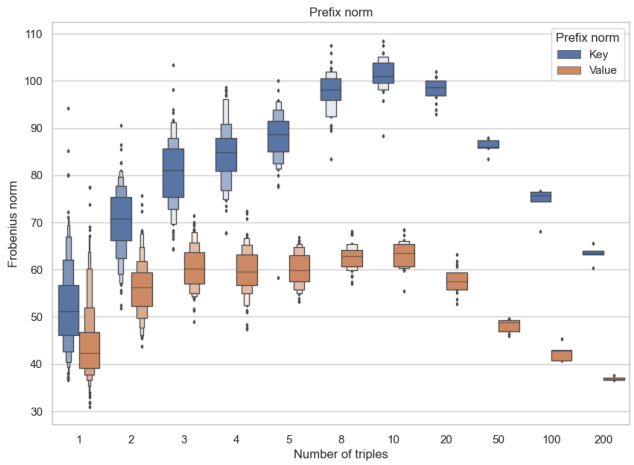

Figure 4: Frobenius norm of the key and value vectors of the prefix at the end of each experiment in the basic setup.

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 4 the results of the evaluation over SciQ and MMLU, which shows that the prefix-tuned models do not seem to forget facts learned during pretraining 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. We did not have time to study this difference in detail, and leave this question open for future work.

	SciQ acc.	MMLU acc.		
k		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 4: Accuracy of the models on the MMLU and SciQ datasets, averaged over 5 random models 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.

Effect of prefix size

Table 5 contains the results obtained when prefix-tuning instances of BLOOMZ-7B1 while varying the number of virtual tokens n contained in the prefix.

We observe significant improvement in accuracy for

	n=1		n=20		n=100	
k	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 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 $n = 1$, using a one-sided z-test for proportions for pLM and a one-sided t-test for the accuracy ($p = 0.05$).

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 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.

Effect of prefix depth

We report in Table 6 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).

We observe that increasing the prefix depth as 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 $k = 3$,

	d=1		d=30	
k	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 6: 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 ($p = 0.05$).

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 $k = 100$. Generally, we hypothesize that increasing the prefix depth allows for much more complex information to be learned, and enables the model to learn at least one information for all but the highest amount of facts in the training set.

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.

Effect 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 two additional models: BLOOMZ-1B7, the 1.7 billion parameter version of BLOOMZ, was chosen for scale comparisons.

We report in Table 7 the prediction accuracy obtained on the entire dataset with no prefix-tuning, and in Table 8 the results obtained after prefix-tuning.

Model	BLOOMZ-1B7	BLOOMZ-7B1
Acc	0.044	0.050
Params/prefix	4,096	8,192

Table 7: Comparison of the baseline models through their accuracy in the prediction setting over the entirety of NOVEL-WD, and the number of parameters contained within a single prefix ($n = 1, d = 1$).

We first observe that BLOOMZ-1B7 and BLOOMZ-7B1 share a similarly low baseline accuracy, despite the latter being significantly larger than the former.

	BLOOMZ-1B7		BLOOMZ-7B1	
k	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 8: 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 ($p = 0.05$).

In terms of scaling, we first 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.

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 when training, but to limit the prefix length to a maximum of 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 Falcon (Almazrouei et al. 2023). 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 entirety of the code used to create NOVEL-WD and perform our experiments can be found on GitHub.

References

Alivanistos, D.; Santamaría, S. B.; Cochez, M.; Kalo, J.-C.; van Krieken, E.; and Thanapalasingam, T. 2022. Prompting as Probing: Using Language Models for Knowledge Base Construction. Publisher: arXiv Version Number: 3.

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, E.; Heslow, D.; Launay, J.; Malartic, Q.; Noune, B.; Pannier, B.; and Penedo, G. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Balakrishnan, S.; Fang, Y.; and Zhu, X. 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)*, 78–88. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Driessche, G. v. d.; Lespiau, J.; Damoc, B.; Clark, A.; Casas, D. d. L.; Guy, A.; Menick, J.; Ring, R.; Hennigan, T.; Huang, S.; Maggiore, L.; Jones, C.; Cassirer, A.; Brock, A.; Paganini, M.; Irving, G.; Vinyals, O.; Osindero, S.; Simonyan, K.; Rae, J. W.; Elsen, E.; and Sifre, L. 2021. Improving language models by retrieving from trillions of tokens.

Chen, C.; Zhang, W. E.; and Shakeri, A. S. 2023. Incorporating Knowledge into Document Summarization: an Application of Prefix-Tuning on GPT-2. ArXiv:2301.11719 [cs].

Chiang, C.-H.; Chuang, Y.-S.; and Lee, H.-y. 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*, 8–15. Taipei: Association for Computational Linguistics.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality | LMSYS Org.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Elhage, N.; Nanda, N.; Olsson, C.; Henighan, T.; Joseph, N.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; DasSarma, N.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2022. A Mathematical Framework for Transformer Circuits.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5484–5495. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: retrieval-augmented language model pretraining. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, 3929–3938. JMLR.org.

He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; and Neubig, G. 2021. Towards a Unified View of Parameter-Efficient Transfer Learning.

Hendrycks, D.; Burns, C.; Basart, S.; Critch, A.; Li, J.; Song, D.; and Steinhardt, J. 2020a. *Aligning AI With Shared Human Values*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020b. *Measuring Massive Multitask Language Understanding*.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339. Melbourne, Australia: Association for Computational Linguistics.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models.

Jang, J.; Ye, S.; Lee, C.; Yang, S.; Shin, J.; Han, J.; Kim, G.; and Seo, M. 2022. TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6237–6250. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Jang, J.; Ye, S.; Yang, S.; Shin, J.; Han, J.; Kim, G.; Choi, S. J.; and Seo, M. 2021. Towards Continual Knowledge Learning of Language Models.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; El-Showk, S.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. ArXiv:2207.05221 [cs].

Kasai, J.; Sakaguchi, K.; Takahashi, Y.; Bras, R. L.; Asai, A.; Yu, X.; Radev, D.; Smith, N. A.; Choi, Y.; and Inui, K. 2022. RealTime QA: What's the Answer Right Now? ArXiv:2207.13332 [cs].

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526. Publisher: Proceedings of the National Academy of Sciences.

Kossen, J.; Rainforth, T.; and Gal, Y. 2023. In-Context Learning in Large Language Models Learns Label Relationships but Is Not Conventional Learning. ArXiv:2307.12375 [cs].

Kucharavy, A.; Schillaci, Z.; Maréchal, L.; Würsch, M.; Dolamic, L.; Sabonnadiere, R.; Percia David, D.; Mermoud, A.; and Lenders, V. 2023. *Fundamentals of Generative Large Language Models and Perspectives in Cyber-Defense*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: 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.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *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)*, 4582–4597. Online: Association for Computational Linguistics.

Lin, B. Y.; Wang, S.; Lin, X.; Jia, R.; Xiao, L.; Ren, X.; and Yih, S. 2022. On Continual Model Refinement in Out-of-Distribution Data Streams. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3128–3139. Dublin, Ireland: Association for Computational Linguistics.

Lin, S.; Hilton, J.; and Evans, O. 2022. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research*.

Liu, J.; Jin, J.; Wang, Z.; Cheng, J.; Dou, Z.; and Wen, J.-R. 2023a. *RETA-LLM: A Retrieval-Augmented Large Language Model Toolkit*.

Liu, J.; Liu, A.; Lu, X.; Welleck, S.; West, P.; Le Bras, R.; Choi, Y.; and Hajishirzi, H. 2022a. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3154–3169. Dublin, Ireland: Association for Computational Linguistics.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023b. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 55(9): 195:1–195:35.

Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; and Tang, J. 2022b. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68. Dublin, Ireland: Association for Computational Linguistics.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021. GPT Understands, Too. ArXiv:2103.10385 [cs].

Logan IV, R.; Balazevic, I.; Wallace, E.; Petroni, F.; Singh, S.; and Riedel, S. 2022. Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2824–2835. Dublin, Ireland: Association for Computational Linguistics.

Ma, Y.; Nguyen, T. H.; and Ma, B. 2022. CPT: Cross-Modal Prefix-Tuning for Speech-To-Text Translation. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6217–6221. ISSN: 2379-190X.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022a. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 35: 17359–17372.

Meng, K.; Sharma, A. S.; Andonian, A.; Belinkov, Y.; and Bau, D. 2022b. Mass-Editing Memory in a Transformer. ArXiv:2210.07229 [cs].

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2021. Fast Model Editing at Scale.

Muennighoff, N.; Wang, T.; Sutawika, L.; Roberts, A.; Biderman, S.; Le Scao, T.; Bari, M. S.; Shen, S.; Yong, Z. X.; Schoelkopf, H.; Tang, X.; Radev, D.; Aji, A. F.; Almubarak, K.; Albanie, S.; Alyafeai, Z.; Webson, A.; Raff, E.; and Raffel, C. 2023. Crosslingual Generalization through Multitask Finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15991–16111. Toronto, Canada: Association for Computational Linguistics.

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)*, 2463–2473. Hong Kong, China: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Razdaibiedina, A.; Mao, Y.; Hou, R.; Khabsa, M.; Lewis, M.; and Almahairi, A. 2023. Progressive Prompts: Continual Learning for Language Models. ArXiv:2301.12314 [cs].

Simoulin, A.; and Crabbé, B. 2021. How Many Layers and Why? An Analysis of the Model Depth in Transformers. In *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*, 221–228. Online: Association for Computational Linguistics.

Strubell, E.; Ganesh, A.; and McCallum, A. 2020. Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09): 13693–13696. Number: 09.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, J.; Huang, W.; Qiu, M.; Shi, Q.; Wang, H.; Li, X.; and Gao, M. 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*, 3164–3177. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wei, J.; Huang, D.; Lu, Y.; Zhou, D.; and Le, Q. V. 2023. Simple synthetic data reduces sycophancy in large language models. ArXiv:2308.03958 [cs].

Welbl, J.; Liu, N. F.; and Gardner, M. 2017. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 94–106. Copenhagen, Denmark: Association for Computational Linguistics.

Wu, Y.; Rabe, M. N.; Hutchins, D.; and Szegedy, C. 2021. Memorizing Transformers.

Wu, Y.; Zhao, Y.; Hu, B.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2022. An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks. 5184–5196.

Yang, Z.; and Liu, Y. 2021. On Robust Prefix-Tuning for Text Classification.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong Learning with Dynamically Expandable Networks.

Yu, J.; Wang, X.; Tu, S.; Cao, S.; Zhang-Li, D.; Lv, X.; Peng, H.; Yao, Z.; Zhang, X.; Li, H.; Li, C.; Zhang, Z.; Bai, Y.; Liu, Y.; Xin, A.; Lin, N.; Yun, K.; Gong, L.; Chen, J.; Wu, Z.; Qi, Y.; Li, W.; Guan, Y.; Zeng, K.; Qi, J.; Jin, H.; Liu, J.; Gu, Y.; Yao, Y.; Ding, N.; Hou, L.; Liu, Z.; Xu, B.; Tang, J.; and Li, J. 2023. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. ArXiv:2306.09296 [cs].

Zhao, L.; Zheng, F.; Zeng, W.; He, K.; Xu, W.; Jiang, H.; Wu, W.; and Wu, Y. 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*, 4848–4862. Seattle, United States: Association for Computational Linguistics.

Zhu, C.; Rawat, A. S.; Zaheer, M.; Bhojanapalli, S.; Li, D.; Yu, F.; and Kumar, S. 2020. Modifying Memories in Transformer Models.

Acknowledgments

The authors would like to thank the ORION program for its contribution to the funding of MM's research internship. This work has benefited from a French government grant managed by the Agence Nationale de la Recherche with the reference ANR-20-SFRI-0009.

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011011668R3 made by GENCI.