Task-Incremental Learning on Long Text Sequences

Natalia Graziuso[1], Andrea Zugarini[2,*] and Stefano Melacci[1]

[1]Department of Information Engineering and Mathematics, University of Siena, Italy
[2]expert.ai, Italy

Abstract

The extraordinary results achieved by Large Language Models are paired with issues that are critical in real-world applications. The costs of inference and, in particular, training are extremely large, both in terms of time and computational resources, and they become prohibitive when working in dynamic environments, where data and tasks are progressively provided over time. The model must be able to adapt to new knowledge, new domains, new settings, without forgetting the previously learned skills. Retraining from scratch easily becomes too costly, thus Continual Learning strategies are of crucial importance. This is even more evident when data consist of "long" documents, that require several resources to be processed by modern neural models, leading to very long prompts. This paper investigates LLM-based Task-Incremental Learning in the case of tasks exploiting long sequences of text, as it is typical in summarization, question-answering on long documents, reviewing long contracts, and several others. We show how adapting the model by Task Arithmetic with LoRA, which was proposed for visual data, yields promising results also in the case of such "long" text data. To our best knowledge, this is the first work along this challenging direction. The outcome of the investigation of this paper is generic enough to represent an important starting point for further research in processing linguistic data in every language.

Keywords

Continual Learning, Task-Incremental Learning, Long Sequences of Text, Large Language Models

1. Introduction

The quality of Language Models (LMs) has been rapidly improving in the last decade, showing outstanding skills when scaled to large data and networks [1], leading to the nowadays popular Large Language Models (LLMs). Solving more complex tasks with LLMs often requires processing "long" documents and articulated long instructions. However, handling lengthy prompts can be a significant obstacle for real-world applications, raising costs and resources required during both inference and, in particular, training. This issue can become critical when the LLM needs to be specialized to many different tasks, domains, and, more generally, when it is applied to dynamic settings that require multiple adaptations. For instance, in real-world applications, models need to be re-trained from time to time, as new data/tasks become available. In such scenarios, the need for Continual Learning (CL) [2, 3] strategies becomes imperative. From a very generic perspective, CL focuses on the development of algorithms capable of sequentially learning from a stream of data, while preserving what was learnt in past experiences, avoiding catastrophic forgetting [4].

In this work, motivated by the aforementioned issues, we study the problem of Continual Learning from "long" sequences of text, exploiting LLMs. We investigate several strategies based on LoRA [5] to adapt an LLM to multiple tasks that are sequentially proposed over time. In particular, we first follow the route of training a single adapter in a sequential manner, then we explore Task Arithmetic to fuse multiple adapters trained independently [6]. We consider the possibility of assigning different weights to each task, and we shed some light on what are the factors that contribute the most to catastrophic forgetting and to effective task adaptation. The outcomes of such an investigation reveals that: (1) there is limited sensitivity to task-order, i.e., regardless of the sequence in which tasks are presented, the overall average performance remains relatively stable, a property that, to our best knowledge, was never evaluated in the case of tasks composed of long documents; (2) despite its simplicity, Task Arithmetic demonstrates effectiveness in addressing forgetting phenomena when learning from long texts, strongly reducing the gap from multiple models independently adapted to the task data. Moreover, (3) we are the first to evaluate a recently proposed benchmark (SCROLLS [7]) in a CL setting, offering reference results for further activity in processing long sequences of text. We remark that while our experiments are based on data in English language, the generic issues we explore about handling long sequences of text are intrinsically shared by every language.

2. Related Work

In the last few years, a variety of approaches were proposed by the scientific community in the context of CL (see [3] and references therein). The main goal is the one

CLiC-it 2024: Tenth Italian Conference on Computational Linguistics, Dec 04 — 06, 2024, Pisa, Italy

*Corresponding author.
natalia.graziuso@student.unisi.it (N. Graziuso); azugarini@expert.ai (A. Zugarini); stefano.melacci@unisi.it (S. Melacci)

of learning from newly provided information, with models that are capable of acquiring new knowledge without forgetting the previously learned one, and, more importantly, without storing the full dataset and retraining from scratch every time [8]. Several efforts are dedicated to the case of lifelong Reinforcement Learning [9] and of Supervised Learning [10], distinguishing among scenarios and categories of approaches [11], ranging from parameter isolation, regularization methods, and replays [12]. Unsupervised or Self-Supervised Learning approaches are also becoming popular [13, 14, 15], and the case of adaptation of pre-trained backbones [16].

Of course, neural models for processing language are a subject of study in the context of CL [17]. We mention the case of language modeling in Lamol [18], which is trained to concurrently solve a task and mimic training examples, thereby preserving the distribution of previous tasks. Sun et al. [12] introduce Distill and Replay, which learns to solve the task, to generate training examples formatted as context-question-answer, and to distill knowledge from a model trained on the previous task(s). Differently, Reasoning-augmented Continual Learning [19] focuses on creating reasoning pathways to preserve and improve LLMs' reasoning abilities and information transfer.

Together with works that learn new models from scratch, several approaches devise fine-tuning strategies for pre-trained Transformers in language processing, that turn out to be efficiently adaptable to a downstream task by learning only a small number of task-specific parameters. It is the case of models that tune the input prompt [20] or of generic Adapters [21], such as the popular LoRA [5], which introduces new weight matrices, parametrized by the product of low-rank ones. Evaluating these models with long contexts [22] is not frequent in the scientific literature, especially in the case in which multiple fine-tunings are sequentially applied, typical of CL, which is the main focus of this paper. In particular, LoRA and Task Arithmetic [23] has been jointly studied to handle CL problems in vision [6], that is what this paper extends to the case of language and long sequences. We also mention works that focus instruction-based model for CL, such as ConTinTin [24], where each task is modelled by a specific instruction that directly defines the target concept along with a few instances that illustrate. Scialom et al. [25] and Luo et al. [4] investigate natural language instructions paired with memory buffers and replays.

3. Task-Incremental Learning on Long Sequences of Text

Task-Incremental Learning (TIL) is a continual learning scenario where the same model is trained on tasks that are presented in a sequential manner. The main challenge consists in profitably learning from the last-presented task without forgetting the previous ones [3]. In order to cope with TIL on Long Sequences of Text, specifically focusing on LLMs, we consider different learning strategies. In this Section we describe each of them in detail, after having formally introduce the TIL problem.

Problem. We are given a model parameterized by θ, which is a vector collecting the learnable variables. In TIL, a set $\mathcal{T}$ of k tasks is sequentially presented to the model, i.e. one at a time. Each task $t \in \mathcal{T}$, features data sampled from a task-specific distribution, collected into dataset $\mathcal{D}_t := (\mathcal{X}_t, \mathcal{Y}_t)$, composed of raw samples and labeling information, respectively. The model is not only expected to learn from $\mathcal{D}_t$, but also to not forget knowledge already acquired from the past tasks. In the following, to keep the notation simple, we indicate each task by a numerical index, thus $t \in \mathcal{T} = \{1, \dots, k\}$. In this case of study, the model is a pre-trained LLM with billions of parameters, and all the TIL tasks are characterized by long input sequences. Such a combination constitutes a computationally demanding mix, making offline/joint training potentially very expensive, that is where CL solutions are very convenient. We consider the case in which LLMs are fine-tuned exploiting adapters [26]. In particular, we focus on LoRA [5], that introduces additional learnable parameters while keeping the rest of the network freezed. This is both less resource demanding, and it also alleviates catastrophic forgetting, since the LoRA weights θ^l are usually of a number that is a small fraction with respect to total model parameters, i.e. $|\theta^l| \ll |\theta|$. Hence, it is a perfect candidate for the experience of this paper.

Single-model TIL with LoRA (S-TIL). In the straightforward implementation of a TIL problem, tasks are presented to the model sequentially starting from the first one up to the k-th one. The order may be given a priori, or established according to some criteria, such as tasks similarity or difficulty (curriculum-like learning [27]). At the beginning, when considering the first task, $t = 1$, we start from a model with freezed parameters θ and additional trainable weights θ^l_1 initialized as described in [5]. At task t, with $t > 1$ instead, the LoRA weights are initialized with the LoRA parameters from previous step, i.e., θ^l_{t-1}. It is worth noticing that in such a way, at the end of the k tasks, the final model parameters will be constituted by the original θ, still unchanged, and a *single* set of adapter parameters θ^l_k, that was sequentially trained over all the tasks.

Multi-model TIL with LoRA (M-TIL). Another way to face the problem of learning the multiple tasks in TIL, is to build a specialized model per task, independently on the other ones. This usually yields strong performance on each sub-problem, guaranteeing no catastrophic forgetting issues, since the model to use is simply retrieved

Table 1
Selected datasets from the SCROLLS benchmark and their main features.

Dataset	*Task*	*Domain*	*Metric*	*#Examples*	
				Train	Validation
Contract NLI	Natural Language Inference	Legal	EM	7191	1097
Qasper	QA	Science	F1	2567	1726
QuALITY	Multi Choice QA	Literature, Misc	EM	2523	2086
QMSUM	Query-based Summarization	Meetings	ROUGE-L	1257	272
SummScreenFD	Summarization	TV	ROUGE-L	3673	338

in function of the task to solve. At the same time, such a strategy requires the storage, deployment and maintenance of k independent models, which is unsustainable with billion-sized models like current LLMs. Even when using adapters such as LoRA, maintaining many of them can be still hard to handle.

Task Arithmetic TIL with LoRA (TA). Based on the concept of "task vectors", Task Arithmetic (TA) [23] was proposed to combine together the weights learned in a multi-model continual learning scenario. A task vector represents the direction in the weights space of a pre-trained model toward a certain task. In TA, multiple directions are fused together via a simple linear combination of them. Similarly, LoRA adapters steers the model behavior to improve performance on a specific task. Therefore, LoRA weights trained separately (multi-model) can be updated with task arithmetic [6]:

$$\theta^l_{\text{final}} = \sum_{t \in \mathcal{T}} \lambda_t \theta^l_t, \tag{1}$$

where λ_t is a scalar weighting the importance of task t.

Fine-tuning by Memory Buffer (FTB). In principle, TA can be applied as it is, without requiring further fine-tuning. However, we also consider refining the parameters using a memory buffer with examples from all the tasks. Indeed, experience replay is a well-known and effective strategy in Reinforcement Learning and Continual Learning problems. Examples were chosen randomly, evenly distributed across the given tasks. Since we are dealing with long documents, we keep it small.

4. Experiments

We experimented LLMs in TIL exploiting sequences of long texts from a benchmark made public to the scientific community in the last few years [7]. Notice that these benchmarks *are not* designed for TIL. Thus, using them in TIL is indeed a novel experience off the beaten track.

4.1. Datasets

We consider five out of seven datasets of SCROLLS [7], that is the reference benchmark for tasks composed of long documents. Datasets belong to different domains, and they are about different tasks, that we adapted to TIL by means of instruction tuning. An overview of the benchmark is provided in Table 1, and here we briefly describe each dataset.

Qasper. Qasper [28] (QSPR) is Question Answering (QA) dataset on academic papers. Crafted by NLP experts, it contains questions based on title and abstract of the paper. There are different kind of inquiries: abstractive, extractive, yes/no questions, including unanswerable ones. To answer the question, the entire paper must be read.

QuaLITY. QuALITY [29] (QALT) is a multiple-choice QA dataset, drawing upon English source articles with an average length of about 5,000 tokens. Original texts are provided in HTML format, retaining paragraph breaks and basic formatting such as italics, but with images removed. Questions are designed to require details from different parts of the text to properly answer them.

QMSum. QMSum, presented in [30], is a question-based document summarization benchmark. The dataset is characterized by long meetings transcripts, collecting 1,808 query-summary pairs from 232 different meetings.

ContractNLI. Contract NLI [31] (CNLI) is the first dataset for Natural Language Inference in contracts. Given a premise and a contract, a model has to classify whether the premise is entailed by, contradicting to or not mentioned by the contract. There are 607 contracts and 17 unique hypotheses, combined to get 10,319 examples.

SummScreenFD. SummScreen [32] (SumScr) is a summarization dataset of TV series transcripts and human written recaps. Examples come from two different sources, but in SCROLLS, authors only kept Forever-Dreaming (FD), due to its greater variety of shows.

4.2. Experimental Setup and Results

We consider Mistral-7B-v0.1 [33] as the backbone LLM for all the fine-tuned models in our TIL experiments. Albeit trained on a restricted context length of at most 8,192 tokens, it supports longer inputs of size up to 32,768. The LLM was quantized via 4-bit quantization in order to fit long sequences on a single A6000 GPU. During train-

ing, the micro batch size was set to 1, with 32 gradient accumulation steps. LoRA adapters were updated with AdamW for 3 epochs in all the experiments, regardless of the dataset. At inference time, outputs were generated using Beam Search with beam size set to 2. We compared: (*i*) Mistral-7B-v0.1-Instruct, the instruction-tuned version of mistral, referred to as Mistral-7b-instruct; (*ii*) The case of multiple independent LoRA adapters, each of them trained in a single dataset, i.e., M-TIL (Section 3); (*iii*) Classic TIL with a single model, progressively updated on the sequence of tasks, i.e., S-TIL (Section 3), considering both the case in which tasks are provided in a certain order (S-TIL$_\downarrow$) or in the opposite one (S-TIL$_\uparrow$); (*iv*) Task Arithmetic (Section 3) with evenly values λ's (TA) or with tasks-specific λ's based on prior knowledge (WTA).

Evaluation. Due to the different nature of each task in SCROLLS, there are different metrics to take into account for each of them. In particular, summarization-like tasks (QMSum and SummScreenFD) are evaluated with ROUGE score [34] (1,2 and L) , whereas, ContractNLI and QuaLITY are assessed with Exact Match (EM). Finally, results on Qasper are measured by F1. A global overview of the metrics can be found in Table 1. We indicate with S_i the score yielded by the associated metric for task i. Following the way the SCROLLS benchmark was proposed, scores are averaged to provide a unique index of Overall Performance OP. Since we focus on TIL, we evaluate OP after each task t, and we also compute the Overall Forgetting at task t (OF_t), also known as index of negative backward transfer [35], which tells how strongly the previously considered tasks have been negatively affected by learning from the current task t, i.e., a measure of catastrophic forgetting [4]. Formally,

$$OP_t = \frac{1}{t}\sum_{i=1}^{t} S_{t,i}, \quad OF_t = \left[\frac{1}{t-1}\sum_{i=1}^{t-1}(S_{i,i} - S_{t,i})\right]_+,$$

where $[\cdot]_+$ keeps the positive part, and $S_{t,i}$ is the score of task i after having learned from task $t \in \mathcal{T}$. Since the test set of SCROLLS is not public, we used the SCROLLS validation set as test set, and sampled a sub-portion of the training data to build a validation set. After cross-validation, we set the rank of LoRA to 8, dropout-rate to 0.05, and α to 16 (see [5] for param description) and learning rate $3 \cdot 10^{-4}$ (linearly decaying).

Investigating S-TIL. Dealing with long sequences of text might affect the TIL procedure in function of the order in which tasks are presented. We study different task orderings based on the average length of the sequences of text in each task, from tasks involving shorter output sequences to the ones involving longer sequences and vice-versa. As anticipated, we named them S-TIL$_\uparrow$ and S-TIL$_\downarrow$, respectively. Results of this experience are presented in detail in Table 2. The training order does strongly affect the final performance on single tasks, promoting higher scores on more recently seen datasets. On one hand, this is expected, since the older ones are more likely affected by catastrophic forgetting. Catastrophic forgetting (last columns of Table 2) at $t = k = 5$ is below 10% in both cases. On the other hand, there is an evident peak of forgetting in S-TIL$_\downarrow$ at $t = 3$, which is then reduced when learning from the following tasks. The peak is due to a strong reduction of performance in the first two tasks after having learned from Qasper (QSPR). We investigated this aspect, and found that the model fails in generating the perfectly-formatted output string that is then exploited in the EM metric. When moving to the following task, this skill is partially recovered. We hypothesize that the presence of unanswerable questions in Qasper negatively bias the types of answers in SummmScreenFD (SumScr) and QMSum, where all the questions have an answer instead.

Comparing Instances S-TIL and M-TIL. Figure 1 compares the models of Table 2 (for $t = k$) with M-TIL, which is composed of multiple adapters, each of them specifically trained on a task, and thus forgetting-free. Performance of both S-TIL's are lower of M-TIL, as expected, but sometimes not far from it. Comparing S-TIL$_\uparrow$ and S-TIL$_\downarrow$, we see that they get similar overall performances, but the latter yields better results in three out of five tasks. The quality of S-TIL$_\uparrow$ (w.r.t. S-TIL$_\downarrow$) improves going right-to-left, and, symmetrically, the one of S-TIL$_\downarrow$ increases going left-to-right, as expected, since they were trained in opposite order (relative gain is > 1 in SumScr due to forward transfer).

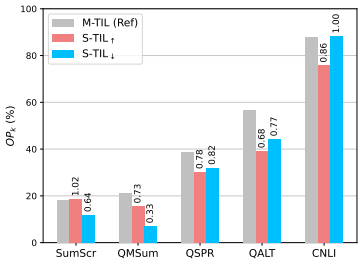

Figure 1: Test results in TIL: overall performance at $t = k = 5$, i.e., OP_k. We compare the cases of S-TIL$_\uparrow$ and S-TIL$_\downarrow$ (see Table 2), with the ones of multiple-independently trained adapters, i.e., M-TIL. **Relative Gain is indicated on the bars**.

The Role of TA. We compared all the introduced models with the case of merging independently-trained adapters with TA. Table 3 shows that TA results to be a simple yet competitive solution, with average performance on par with S-TIL$_\downarrow$. Actually, observing task-wise performance, we can see how TA outperforms S-TIL$_\downarrow$ across all the datasets, with the exception of ContractNLI

Table 2
Evaluation score (%) on test data, for each task, after having learned from task t (i.e., $S_{t,i}$) in S-TIL$_\uparrow$ (**left**) and S-TIL$_\downarrow$ (**right**). The order of columns (dataset names) reflect the task-order followed during training. Tasks becomes available in order, thus — indicate that the value cannot be computed yet. The OF_t column is about catastrophic forgetting (the lower the better).

$i\rightarrow$ $t\downarrow$	1.**CNLI**	2.**QALT**	3.**QSPR**	4.**QMSum**	5.**SumScr**	OF_t
	$S_{t,i}$					
1	88.0	-	-	-	-	-
2	85.7	49.5	-	-	-	2.31
3	79.7	43.2	37.1	-	-	7.31
4	82.9	40.7	27.6	21.9	-	7.82
5	75.7	39.1	30.2	15.5	18.6	8.99

$i\rightarrow$ $t\downarrow$	1.**SumScr**	2.**QMSum**	3.**QSPR**	4.**QALT**	5.**CNLI**	OF_t
	$S_{t,i}$					
1	18.2	-	-	-	-	-
2	16.1	22.2	-	-	-	2.06
3	0.04	0.45	37.4	-	-	19.94
4	13.6	13.3	35.8	47.7	-	5.00
5	11.8	7.0	32.0	44.2	88.2	7.60

Table 3
Results involving all the competitors. In ROUGE-based evaluations, we also report unigram overlap (ROUGE-1), bigram overlap (ROUGE-2), together with the longest overlapping subsequence (ROUGE-L) – the last one is what is considered when computing OP_k. Reference results (baseline, and "upper bound") are in italic.

Method	**SumScr** ROUGE-1/2/L			**QMSum** ROUGE-1/2/L			**QSPR** F1	**QALT** EM	**CNLI** EM	OP_k
Ref1: Mistral-7b-instruct	*18.1*	*2.3*	*10.8*	*16.2*	*2.7*	*11.8*	*5.4*	*0.0*	*0.0*	*5.6*
Ref2: M-TIL	*29.2*	*7.1*	*18.2*	*29.6*	*8.5*	*21.1*	*38.7*	*56.7*	*88.0*	*44.5*
S-TIL$_\uparrow$	**30.0**	**7.8**	**18.6**	20.6	5.7	15.5	30.2	39.1	75.7	35.8
S-TIL$_\downarrow$	15.6	3.6	11.8	8.7	2.3	7.0	32.0	44.2	**88.2**	36.7
TA	20.7	4.56	13.9	18.8	5.6	14.2	36.0	45.6	72.6	36.5
WTA	19.4	4.26	13.4	18.5	5.5	14.1	34.7	47.9	74.7	36.9
TA-FTB	28.6	6.21	17.5	**28.0**	**8.1**	**20.1**	**38.3**	47.8	75.1	39.8
WTA-FTB	28.6	6.09	17.2	26.9	7.6	19.7	35.6	**50.5**	78.5	**40.3**

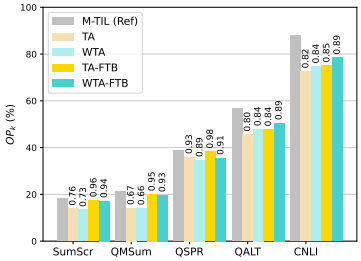

Figure 2: Test results in TIL with Task Arithmetic (TA). TA is explored with or without Fine-tuning by Memory Buffer (FTB), and also in the case of task-specific weights provided in advance (WTA). Same setting of Figure 1.

(CNLI), the last task in which S- TIL$_\downarrow$ was specialized. In WTA, λ's for non-QA datasets were halved, since there tasks involve generation of longer outputs that more strongly condition the behaviour of the LLM, as already discussed for Qasper. WTA yielded evident improvements in the last two datasets, despite being less weighed, keeping similar performance on the others. This suggests that appropriately weighing the task-vectors in Eq. 1 is a viable road to improve the model.

Impact of FTB. We also investigate the impact of rehashing the memory of the TA/WTA model via fine-tuning it on just 50 samples per the tasks (memory buffer). Despite being a simple refinement stage, results presented in Table 3 show a consistent boost of performance when using the memory buffer (FTB), reaching about 39.0 averaged score, when using the weighted TA version, significantly reducing the gap from the k-independent adapters solution of M-TIL. Figure 2 provides a quick view on the already presented results of all the TA methods we considered, reporting also the Relative Gain w.r.t. M-TIL. Indeed, we can observe that the relative drop in performance is always below the 11%.

5. Conclusions

We investigated Large Language Models in progressively learning from tasks involving long sequences of text. A pre-trained model was paired with one or more adapters (LoRA), and we analyzed the role of Task Arithmetic, showing that it yields performances that are not far from the ones of multiple models independently trained to solve each task. Our results suggests a viable road to mitigate the need of large computational resources when learning from tasks based on "long" documents. While we

exploited data in English language, the experiences of this paper can be interpreted as generic attempts to leverage long sequences in Continual Learning, in a sense going beyond the language barrier. Future work will consider schemes to automatically tune the Task Arithmetic [36].

Acknowledgments

The work was partially funded by:

- "ReSpiRA - REplicabilità, SPIegabilità e Ragionamento", a project financed by FAIR, Affiliated to spoke no. 2, falling within the PNRR MUR programme, Mission 4, Component 2, Investment 1.3, D.D. No. 341 of 03/15/2022, Project PE0000013, CUP B43D22000900004 [1];
- "enRichMyData - Enabling Data Enrichment Pipelines for AI-driven Business Products and Services", an Horizon Europe (HE) project, grant agreement ID: 101070284 [2].

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[2] R. Hadsell, D. Rao, A. A. Rusu, R. Pascanu1, Embracing change: Continual learning in deep neural networks, Trends in Cognitive Sciences 24 (2020) 1028–1040.

[3] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2024) 5362–5383. doi:10.1109/TPAMI.2024.3367329.

[4] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, Y. Zhang, An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2023. arXiv:2308.08747v2, [cs.CL].

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[6] R. Chitale, A. Vaidya, A. M. Kane, A. Ghotkar, Task Arithmetic with LoRA for Continual Learning, in: Workshop on Advancing Neural Network Training at 37th Conference on Neural Information Processing Systems (WANT@NeurIPS 2023), 2023.

[7] U. Shaham, E. Segal, M. Ivgi, A. Efrat, O. Yoran, A. Haviv, A. Gupta, W. Xiong, M. Geva, J. Berant, O. Levy, Scrolls: Standardized comparison over long language sequences, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics, 2022, pp. 12007–12021.

[8] M. Gori, S. Melacci, Collectionless artificial intelligence, arXiv preprint arXiv:2309.06938 (2023).

[9] K. Khetarpal, M. Riemer, I. Rish, D. Precup, Towards continual reinforcement learning: A review and perspectives, Journal of Artificial Intelligence Research 75 (2022) 1401–1476.

[10] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE transactions on pattern analysis and machine intelligence 44 (2021) 3366–3385.

[11] G. M. van de Ven, A. S. Tolias, Three continual learning scenarios, in: NeurIPS Continual Learning Workshop, volume 1, 2018.

[12] J. Sun, S. Wang, J. Zhang, C. Zong, Distill and replay for continual language learning, in: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, December 8-13, 2020, pp. 3569–3579.

[13] S. Marullo, M. Tiezzi, A. Betti, L. Faggi, E. Meloni, S. Melacci, Continual unsupervised learning for optical flow estimation with deep networks, in: Conference on Lifelong Learning Agents, PMLR, 2022, pp. 183–200.

[14] S. Paul, L.-J. Frey, R. Kamath, K. Kersting, M. Mundt, Masked autoencoders are efficient continual federated learners, arXiv preprint arXiv:2306.03542 (2023).

[15] M. Tiezzi, S. Marullo, L. Faggi, E. Meloni, A. Betti, S. Melacci, Stochastic coherence over attention trajectory for continuous learning in video streams, in: L. D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 3480–3486. URL: https://doi.org/10.24963/ijcai.2022/483. doi:10.24963/ijcai.2022/483, main Track.

[16] S. Marullo, M. Tiezzi, M. Gori, S. Melacci, T. Tuytelaars, Continual learning with pretrained backbones by tuning in the input space, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–9.

[17] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, G. Haffari, Continual learning for large language models: A survey, arXiv preprint arXiv:2402.01364 (2024).

[18] F.-K. Sun, C.-H. Ho, H.-Y. Lee, Lamol: Language

[1] RESPIRA: https://www.opencup.gov.it/portale/web/opencup/home/progetto/-/cup/B43D22000900004

[2] https://doi.org/10.3030/101070284

modeling for lifelong language learning, arXiv preprint arXiv:1909.03329 (2019).

[19] X. Wang, Y. Zhang, T. Chen, S. Gao, S. Jin, X. Yang, Z. Xi, R. Zheng, T. Yicheng Zou, X. H. QiZhang, Trace: A comprehensive benchmark for continual learning in large language models, 2023. arXiv:2310.06762v1.

[20] Q. Zhu, B. Li, F. Mi, X. Zhu, M. Huang, Continual prompt tuning for dialog state tracking, 2022. arXiv:2203.06654.

[21] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, L. Si, On the effectiveness of adapter-based tuning for pretrained language model adaptation, arXiv preprint arXiv:2106.03164 (2021).

[22] Y. Chen, S. Qian, Z. Liu, H. Tang, X. Lai, S. Han, J. Jia, Longlora: Efficient fine-tuning of long context large language models, 2023. arXiv:2309.12307v2.

[23] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, A. Farhadi, Editing models with task arithmetic, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.

[24] W. Yin, J. Li, C. Xiong, Contintin: Continual learning from task instructions, arXiv preprint arXiv:2203.08512 (2022).

[25] T. Scialom, T. Chakrabarty, S. Muresan, Fine-tuned language models are continual learners, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 6107–6122.

[26] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International conference on machine learning, PMLR, 2019, pp. 2790–2799.

[27] X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, IEEE transactions on pattern analysis and machine intelligence 44 (2021) 4555–4576.

[28] P. Dasigi, K. Lo, I. Beltagy, A. Cohan, N. A. Smith, M. Gardner, A dataset of information-seeking questions and answers anchored in research papers, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online. Association for Computational Linguistics, 2021, pp. 4599–4610.

[29] R. Y. Pang, A. Parrish, N. Joshi, N. Nangia, J. Phang, A. Chen, V. Padmakumar, J. Ma, J. Thompson, H. He, et al., Quality: Question answering with long input texts, yes!, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022.

[30] M. Zhong, D. Yin, T. Yu, A. Zaidi, R. Mutethia Mutuma, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, D. Radev, Qmsum: A new benchmark for query based multi-domain meeting summarization, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online. Association for Computational Linguistics, 2021, pp. 5905–5921.

[31] Y. Koreeda, C. D. Manning, Contractnli: A dataset for document-level natural language inference for contracts, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 1907–1919.

[32] M. Chen, Z. Chu, S. Wiseman, K. Gimpel, Summscreen: A dataset for abstractive screenplay summarization, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 8602–8615.

[33] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[34] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, Association for Computational Linguistics, 2004, pp. 74–81.

[35] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Advances in neural information processing systems 30 (2017).

[36] M. Tiezzi, S. Marullo, F. Becattini, S. Melacci, Continual neural computation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2024, pp. 340–356.