Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora

Xisen Jin[†1] **Dejiao Zhang**[2] **Henghui Zhu**[2] **Wei Xiao**[2]
Shang-Wen Li[‡2] **Xiaokai Wei**[2] **Andrew Arnold**[2] **Xiang Ren**[1]

[1]University of Southern California [2]AWS AI Labs

{xisenjin,xiangren}@usc.edu

{dejiaoz, henghui, weixiaow, shangwenl, xiaokaiw, anarnld}@amazon.com

Abstract

Pretrained language models (PTLMs) are typically learned over a large, static corpus and further fine-tuned for various downstream tasks. However, when deployed in the real world, a PTLM-based model must deal with data distributions that deviate from what the PTLM was initially trained on. In this paper, we study a *lifelong language model pretraining* challenge where a PTLM is continually updated so as to adapt to emerging data. Over a domain-incremental research paper stream and a chronologically-ordered tweet stream, we incrementally pretrain a PTLM with different continual learning algorithms, and keep track of the downstream task performance (after fine-tuning). We evaluate PTLM's ability to adapt to new corpora while retaining learned knowledge in earlier corpora. Our experiments show distillation-based approaches to be most effective in retaining downstream performance in earlier domains. The algorithms also improve knowledge transfer, allowing models to achieve better downstream performance over the latest data, and improve temporal generalization when distribution gaps exist between training and evaluation because of time. We believe our problem formulation, methods, and analysis will inspire future studies towards continual pretraining of language models.

1 Introduction

Pretrained language models (PTLMs) have achieved remarkable performance on benchmark datasets for a range of NLP tasks (Liu et al., 2019b; Brown et al., 2020). However, when deployed in the wild, NLP systems must deal with emerging data that have constantly shifting data distribution, different from the text corpora they were initially pretrained on — for example, when new data domains are introduced (upper part of Fig. 1) (Gururangan et al., 2020), or when the language uses and vocabulary change over time (lower part of Fig. 1) (Lazaridou et al., 2021). Fine-tuning from a static and possibly "outdated" PTLM may limit the model performance on downstream tasks, as the PTLM may no longer provide an effective model initialization (Beltagy et al., 2019; Müller et al., 2020). Here we look to understand whether continuously adapting a PTLM to emerging data can yield gains on various downstream tasks, and how to achieve better downstream performance for such lifelong PTLM adaptation.

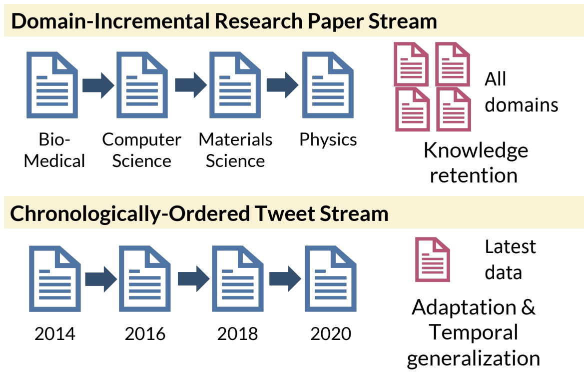


Figure 1: Two data streams created for studying lifelong language model pre-training. We focus on evaluating knowledge retention on the domain-incremental research papers stream; we focus on adaptation to the latest data and temporal generalization on the chronologically ordered tweet stream.

A number of recent works make attempts on adapting PTLMs to a new data domain. Gururangan et al. (2020); Yao et al. (2021) adapt language models to corpora of different genres and topics and observe performance improvement in domain-specific downstream tasks. Arumae et al. (2020) further show that by regularizing the parameters of PTLMs, the downstream tasks performance on the general domain can be preserved. Another line of works focuses on temporal domain shift (Hombaiah et al., 2021), which analyzes the effect of pretraining over up-to-date data to the downstream tasks. Röttger and Pierrehumbert (2021) further study vocabulary composition approaches for improving adaptation to up-to-date corpora. However, these work focus their study on adapting PTLM to a single new domain; while in practice, corpora from distinct domains and time stamps may emerge sequentially. Whether one can maintain a single, up-to-date PTLM remains an open problem. Related to this, Lazaridou et al. (2021) study adaptation of PTLMs over temporal data streams, but solely focus on language modeling instead of fine-tuning performance. It is also important to understand multiple aspects of the utility of lifelong PTLM pretraining, such as knowledge retention over all the seen data, and study what methods can improve the utility of PTLMs in such a continual pretraining process.

In this paper, we formulate a *Lifelong Language Model Pretraining* task to simulate practical scenarios of maintaining and adapting a PTLM over emerging corpora, create a testbed (along with pretraining data streams and downstream tasks) for studying continual pretraining algorithms, and present a systematic evaluation protocol for measuring the progress made on this challenging problem (see Figure 2 for an illustration). We consider two types of text corpus sequences when constructing pretraining data streams, each of which simulates a representative use case and that has slightly different focuses on the evaluation: continuously learning a single model that is applicable to both old and new domains; and improving the model's ability to handle latest data. Specifically, we construct 1) a domain-incremental text stream that consists of academic papers published in four research fields, and 2) a temporal tweet stream that consists of tweets collected from four different years. By conducting systematic experiments on these two data streams, we look to answer a series of analysis questions: 1) whether continual pretraining retains fine-tuning performance over earlier corpora compared to traditional offline pretraining, 2) whether pretraining improves downstream performance on the latest data, and 3) whether pretraining improves temporal generalization where training and evaluation have distribution gaps because of time.

To address the research questions above, we conduct a systematic evaluation of existing continual learning (CL) algorithms, spanning over model-expansion based, memory-based, and distillation-based approaches. Our results show distillation-based approaches are most effective in knowledge retention in the research paper stream, while simultaneously improve adaptation to latest data and temporal generalization in the tweet stream. We believe our problem formulation, evaluation setup, methods and analysis can inspire more future work on continual pretraining of language models.

2 Problem Formulation

Here we present the problem formulation for lifelong pretraining of PTLM, provide details about the data stream construction process and downstream tasks, and introduce the evaluation protocol.

2.1 Lifelong Pretraining of PTLMs

We consider the scenario where one needs to deploy and/or maintain NLP models over a sequence of T data domains. At each time step t the model visits an unlabeled text corpus D_t from a domain with a data distribution $P(D_t)$. The data distribution $P(D_t)$ evolves as the time step t, forming a *data stream* $D_{1..T} = \{D_1, D_2, ...D_T\}$. In practice, the data domain shift can refer to the topic change of the text content (from computer science research papers to biomedical papers), or temporal evolution of the text (from past to recent tweets). The task of *lifelong pretraining of PTLM* looks to continuously adapt a language model f as the model visits (unlabeled) text corpus D_t from the data stream $D_{1..T}$, in order to provide a good model initialization for fine-tuning on downstream tasks from the same domain. With slight abuse in notations, we also use D_t to directly refer to a data domain.

Here, we assume a language model f is updated sequentially over each pretraining corpora D_t, without accessing the full earlier corpora $\{D_i\}_{i<t}$ in the data stream $D_{1..T}$. This aims to capture practical constraints such as privacy restriction for storing earlier data, or computation budget for training over all the text corpora in $D_{1..T}$. We use f_t to denote the language model right *after* updating on the domain D_t. In our study, f is a RoBERTa-base transformer (Liu et al., 2019b) and the model (f_0) is initialized with pretrained RoBERTa weights.

The utility of the PTLMs $\{f_t\}$ is evaluated based on their fine-tuned model performance on various downstream tasks. After updating on a domain D_i, the model f_i can be fine-tuned over downstream tasks from visited domains D_t where $t \leq i$. We note the set of downstream tasks related to domain D_t as $S_t = \{S_t^j\}_{j=1}^{N_t}$, assuming the number of downstream tasks is N_t. Note that in the fine-tuning stage, model f_t has no access to any of the pretraining corpus $D_{1..T}$.

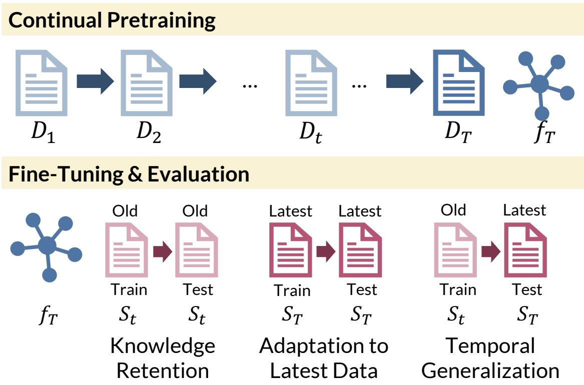


Figure 2: **Training, evaluation setups, and metrics of lifelong language model pretraining.** The model sequentially visits each corpus, and is fine-tuned on downstream datasets related to the domains of pretraining. We evaluate knowledge retention and adaptation to new data with downstream fine-tuning performance on old and latest domains respectively. Besides, we evaluate temporal generalization where training/test examples are drawn from different time steps.

2.2 Data Streams & Downstream Datasets

We construct data streams to simulate two representative scenarios of data domain shifts in practice (also see Fig. 1): one *domain-incremental* stream to simulate the sequential changes of research paper areas; and one *chronologically-ordered* stream to simulate tweets emerging over time.

Domain-incremental Paper Stream. This paper stream consists of the full text of research papers published in four research areas: biomedical, computer science, material science, and physics, filtered from the S2ORC dataset[1], which are presented sequentially to the model. For each domain, we evaluate downstream performance over two datasets. The downstream tasks span over various tasks such as relation extraction and named entity recognition, and are summarized in Table 1. We detail these datasets in Appendix D.

Chronologically-ordered Tweet Stream. This tweet data stream consists of tweets from the year 2014, 2016, 2018 and 2020, collected by the Archive Team[2] and preprocessed following Nguyen et al. (2020). These four tweet corpora are presented sequentially to the language model following the chronological order of the tweet year. For downstream tasks, we hold out 1M tweets from each year's corpus to construct multi-label hashtag prediction datasets (Gong and Zhang, 2016) and single-label emoji prediction datasets (Barbieri et al., 2018). On two datasets, we report label ranking average precision scores (a multi-label version of MRR) of models (Azeemi and Waheed, 2021) and Macro-F1 respectively. The detailed dataset construction process is included in Appendix D.

[1]We use the 20200705v1 version of the S2ORC dataset at https://github.com/allenai/s2orc

[2]https://archive.org/details/twitterstream

Domains	Downstream Datasets	Metrics
Bio-Medicine	Chemprot (Vindahl, 2016)	Micro-F1
	RCT-Sample (Dernoncourt and Lee, 2017)	Micro-F1
Comp. Science	ACL-ARC (Jurgens et al., 2018)	Macro-F1
	SciERC (Luan et al., 2018)	Macro-F1
Mat. Science	Synthesis (Mysore et al., 2019)	Macro-F1
	MNER (Olivetti et al., 2020)	Micro-F1
Physics	Keyphrase (Augenstein et al., 2017)	Macro-F1
	Hyponym (Augenstein et al., 2017)	Macro-F1

Table 1: Summary of downstream datasets relevant to each domain in the research paper stream.

2.3 Evaluation Protocol

We consider three key aspects for evaluating the utility of the language models $\{f_t\}$ that are continuously updated over the data stream $D_{1..T}$, also illustrated in Figure 2: 1) knowledge retention and transfer over the pretraining corpora seen earlier; 2) adaptation to the latest data domain, and 3) temporal generalization when training and evaluation data are from different time steps.

Knowledge Retention. A key utility of continual language model pretraining is to obtain a single model applicable to all domains. We focus on the evaluation of the ability with the domain-incremental paper stream, because for the tweet stream, the practical need of performance over outdated data is limited. Knowledge retention is measured with the downstream task performance from earlier or the current domains that the pretrained model has visited. More formally, for each pretrained model checkpoint in $\{f_i\}$, we fine-tune f_i over downstream tasks $\{S_t\}$ where $t \leq i$ and evaluate the corresponding test set performance. It is important that the models do not suffer from catastrophic forgetting (Robins, 1995), *i.e.,* significantly reduced helpfulness when f_i is fine-tuned for downstream tasks S_t from earlier domains with $t < i$.

Adaption to Latest Data Domain. In certain scenarios, performance of downstream models over the latest data domain should be emphasized. For example, classifiers in the tweet domain are usually trained and evaluated with up-to-date data for practical deployment. Formally, we focus on the downstream task performance of models fine-tuned from the final pretrained model checkpoint f_T, where the downstream tasks S_T are also from the latest domain. To succeed in these metrics, it is crucial for the model to transfer knowledge from earlier domains to the latest domain.

Temporal Generalization Ability. We consider another practical fine-tuning scenario in the tweet stream where the model is trained on outdated data and evaluated on the latest data (Rijhwani and Preotiuc-Pietro, 2020; Huang and Paul, 2018), referred to as the *temporal generalization* ability. Formally, we fine-tune the final pretrained model checkpoint f_T over the training set of downstream tasks S_t from an earlier time step ($t < T$), and evaluate on the test set of the downstream tasks S_T from the latest time step T.

3 Methods

Lifelong language model pretraining introduces novel challenges because of the large training sets and more comprehensive evaluation protocols compared to classification tasks. We establish several strong baselines, and evaluate the performance of continual learning algorithms from different categories spanning over model-expansion, memory-based, and distillation-based approaches, We illustrate the approaches in Figure 3.

3.1 Simple Baselines

We consider several simple baselines which continual learning algorithms will be compared against. `RoBERTa-base` (f_0) corresponds to not pretraining on any of the domain-specific corpora. By separately pretraining f_0 on each corpus $D_1, D_2, ...D_T$, we obtain T `Task-Specific` pretrained models. We also pretrain f_0 sequentially over $D_{1..T}$, which we refer to as `sequential pretraining`. While it allows knowledge transfer between domains compared to domain-specific models, without any continual learning algorithms, sequential pretraining is prone to catastrophic forgetting (Robins, 1995). Finally, we randomly shuffle corpora from all domains $D_{1..T}$ before pretraining, noted as `Multi-Task Learning (MTL)`. MTL corresponds to an offline training paradigm that models new corpora by re-training over all corpora seen before. The drawback is that it requires storing full data from earlier domains, and that it can be extremely costly to repetitively retrain over earlier data if new data keeps emerging.

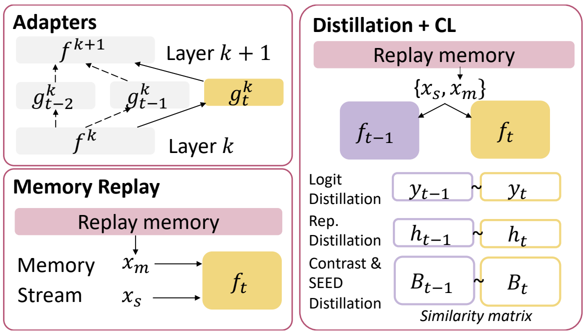


Figure 3: **Comparison of adapter, memory replay, and distillation-based continual learning algorithms.** Details of the methods are introduced in Sec. 3.

3.2 Model-expansion and Regularization-based Methods

We first introduce model-expansion based approaches, which add small trainable modules (*e.g.,* multi-layer perceptron) to the model per new domain while keeping other parts of the model frozen. The `Adapter` approach is a representative approach that learns a set of "adapter" layers $g_t = \{g^k_t\}^K_{k=1}$ for each domain D_t and each of the K transformer layers (Houlsby et al., 2019). We also experiment with a simple `Layer Expansion` approach, which learns separate top two layers of the transformer and the prediction head for each domain. We also involve a regularization-based continual learning baseline, online EWC (Schwarz et al., 2018), which directly penalize change of model parameters.

3.3 Memory Replay Methods

We also experiment with Experience Replay (ER) (Chaudhry et al., 2019), which alleviates forgetting by storing a subset of earlier examples and periodically re-training (replaying) over them. We maintain a fixed-size memory M ($100k$ examples by default) and populate the memory M each time pretraining on a domain D_t finishes with examples in the current domain. We ensure M always contains a balanced sample of examples from all seen domains $D_{1..t}$. We replay a mini-batch of examples from the memory every 10 training steps.

3.4 Distillation-based CL Methods

While knowledge distillation (KD) (Hinton et al., 2015) techniques have been studied intensively for pretrained language models (Sun et al., 2019), applying them to continual learning has been underexplored outside image classification tasks (Li and Hoiem, 2018; Rebuffi et al., 2017; Hou et al., 2018). Distillation based CL approaches store one previous model checkpoint of the model (noted as f_{t-1}) and regularize the differences between f_{t-1} and the current model f_t. We adapt several existing knowledge distillation techniques to PTLMs and utilize them for continual learning. We note, while individual distillation techniques are not original, their adaptation to CL algorithms can be novel.

We perform distillation with examples from the current domain D_t and a replay memory M (similar to ER). Despite the potential gap between D_t and the training data of f_{t-1}, the approach allows utilizing more data for distillation. Formally, each time the model receives a mini-batch of stream examples $\boldsymbol{x}_s$ or a draws mini-batch of memory examples $\boldsymbol{x}_m$ from M (both noted as $\boldsymbol{x}$), we collect certain outputs of the model (*e.g.*, output logits or intermediate representations) with f_{t-1} and f_t. We compute a distillation loss $\ell_{\text{KD}}(\boldsymbol{x}, f_{t-1}, f_t)$ that penalizes the differences between the model outputs, and jointly optimize it with the masked language modeling loss ℓ_{MLM}. The final objective is written as $\ell = \ell_{\text{MLM}} + \alpha \ell_{\text{KD}}$, where α is a hyperparameter to weight the distillation loss.

Logit Distillation. In logit distillation (Hinton et al., 2015), we collect the output logits of f_t and f_{t-1}, noted as $\boldsymbol{y}_t$ and $\boldsymbol{y}_{t-1}$ respectively. The distillation loss is computed as $D_{\text{KL}}(\boldsymbol{y}_t, \boldsymbol{y}_{t-1})$, where D_{KL} is the Kullback–Leibler divergence function.

Representation Distillation. We also consider minimizing the representational deviation of sentences between previous and current models (Sun et al., 2019; Jiao et al., 2020). We extract the representation of each word of two models, noted as $\boldsymbol{h}_{t-1}^{1:N}$ and $\boldsymbol{h}_t^{1:N}$, before the masked language modeling prediction head, where N is the length of the sentence. Then, we compute MSE loss $||\boldsymbol{h}_{t-1}^{1:N} - \boldsymbol{h}_t^{1:N}||_2^2$ as the distillation loss.

Contrastive Distillation. In addition to output logits and hidden representations, we further look into *representational similarity within a batch of examples* as additional knowledge to distill. The approach is adapted from (Cha et al., 2021), which is originally studied for supervised image classification tasks. We briefly introduce the adapted algorithm and leave the details in Appendix E. During continual pretraining, in addition to the language model pretraining objective, we add an unsupervised contrastive learning objective, namely the SimCSE (Gao et al., 2021) objective to encourage sentence representations to reflect semantic similarities between sentences. Then, we compute the intra-batch representational similarity matrices of sentence representations (*i.e.* between each pair of examples in the mini-batch) with f_{t-1} and f_t, noted as $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$, and minimize the cross entropy loss $\ell_{\text{distill}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{B}_{ij}^{t-1}\log\mathbf{B}_{ij}^{t}$

Self-Supervised Distillation (SEED). SEED distillation proposed by (Fang et al., 2021) has a similar spirit as the contrastive distillation. The only difference is that it distills representational similarity *between the batch and a large set of other examples*. We leave the details of the algorithm in Appendix E. We further combine SEED Distillationwith logit distillation and refer to the approach as SEED-Logit Distillation.

4 Results

We summarize our findings over the created data streams. We ask whether lifelong pretraining and continual learning algorithms are effective base on our evaluation protocol proposed in Sec. 2.3.

4.1 Experiment Settings

We use the `RoBERTa-base` model (Liu et al., 2019b), initialized with RoBERTa-base weights throughout the experiments. We set the maximal sequence length to 128 and an effective training batch size of 2,048. On the research paper stream, models are trained for $8k$ steps in the first domain and $4k$ steps in the subsequent domains. On the Tweet stream, we train the models for $4k$ steps in each domain. These correspond to less than a single pass of data in each domain. See Appendix A for detailed setups.

4.2 Domain Incremental Data Stream

As we introduced in Sec. 2.2, in the domain incremental research paper stream, we expect a model f_t to perform well on all downstream tasks $S_{1..t}$ from domains $D_{1..t}$. In Table 2, we report the performance of models on all downstream tasks $S_{1..T}$ fine-tuned from the final pretraining checkpoint, f_T. We visualize more complete change of downstream task performance over different time steps of pretraining (*i.e.,*, f_1, f_2, f_3, f_4) in Fig. 4. We also report the log perplexity of masked language modeling (MLM) in Table 2 as additional information. With these results, we address the research questions below.

Task	D_1 - Biomedical			D_2 - Computer Science			D_3 - Materials Science			D_4 - Physics		
Dataset	Chemprot	RCT-Sample	MLM	ACL-ARC	SciERC	MLM	MNER	Synthesis	MLM	Keyphrase	Hyponym	MLM
Roberta-base	$82.03_{\pm0.7}$	$78.07_{\pm0.7}$	1.993	$64.32_{\pm2.8}$	$79.07_{\pm1.6}$	2.153	$83.15_{\pm0.3}$	$91.25_{\pm0.6}$	2.117	$66.21_{\pm1.0}$	$67.59_{\pm4.5}$	2.278
Sequential Pretraining	$82.09_{\pm0.5}$	$79.60_{\pm0.5}$	1.654	$72.73_{\pm2.9}$	$81.43_{\pm0.8}$	1.807	$\mathbf{83.99}_{\pm0.3}$	$92.10_{\pm1.0}$	1.590	$\mathbf{67.57}_{\pm1.0}$	$\mathbf{74.68}_{\pm4.4}$	**1.381**
ER	$82.73_{\pm0.3}$	$79.98_{\pm0.3}$	1.737	$72.50_{\pm1.0}$	$81.64_{\pm1.1}$	1.857	$\mathbf{83.99}_{\pm0.4}$	$92.65_{\pm0.4}$	1.621	$66.11_{\pm1.1}$	$72.82_{\pm4.3}$	1.391
Online EWC	$81.83_{\pm0.2}$	$78.84_{\pm0.5}$	1.655	$71.81_{\pm2.6}$	$80.79_{\pm0.5}$	1.803	$83.43_{\pm0.4}$	$91.89_{\pm0.5}$	1.571	$66.70_{\pm0.6}$	$72.98_{\pm6.0}$	1.388
Adapter	$83.30_{\pm0.4}$	$80.41_{\pm0.4}$	1.417	$69.32_{\pm3.5}$	$80.22_{\pm1.5}$	**1.633**	$83.91_{\pm0.3}$	$91.69_{\pm0.6}$	1.522	$66.23_{\pm1.4}$	$69.65_{\pm4.5}$	1.554
Layer Expansion	$83.74_{\pm0.3}$	$81.10_{\pm0.5}$	**1.210**	$65.17_{\pm2.9}$	$79.35_{\pm0.8}$	1.756	$82.48_{\pm0.4}$	$92.33_{\pm1.0}$	1.389	$65.70_{\pm1.1}$	$73.34_{\pm3.7}$	1.534
Logit-KD	$83.39_{\pm0.4}$	$\mathbf{81.21}_{\pm0.1}$	1.392	$\mathbf{73.70}_{\pm3.4}$	$81.92_{\pm0.8}$	1.699	$83.96_{\pm0.3}$	$92.20_{\pm1.0}$	**1.425**	$64.75_{\pm1.1}$	$71.29_{\pm3.6}$	1.460
Rep-KD	$82.34_{\pm0.3}$	$79.59_{\pm0.5}$	1.684	$71.17_{\pm2.5}$	$78.78_{\pm1.1}$	1.810	$84.13_{\pm0.3}$	$92.02_{\pm0.8}$	1.585	$65.96_{\pm1.6}$	$73.93_{\pm5.5}$	1.389
Contrast-KD	$82.29_{\pm0.5}$	$79.92_{\pm0.4}$	1.722	$71.15_{\pm1.1}$	$80.49_{\pm1.6}$	1.856	$83.26_{\pm0.4}$	$92.62_{\pm0.7}$	1.612	$65.95_{\pm1.7}$	$72.26_{\pm3.1}$	1.428
SEED-KD	$82.78_{\pm0.3}$	$80.38_{\pm0.4}$	1.720	$69.98_{\pm2.4}$	$81.61_{\pm0.7}$	1.829	$82.99_{\pm0.4}$	$92.35_{\pm0.7}$	1.609	$65.35_{\pm1.0}$	$74.79_{\pm4.1}$	1.401
SEED-Logit-KD	$\mathbf{83.72}_{\pm0.4}$	$81.05_{\pm0.2}$	1.391	$69.90_{\pm4.5}$	$\mathbf{83.03}_{\pm0.6}$	1.703	$83.28_{\pm0.5}$	$\mathbf{92.87}_{\pm1.0}$	1.428	$65.96_{\pm1.5}$	$71.92_{\pm5.5}$	1.460
Task-Specific LM	$83.74_{\pm0.3}$	$81.10_{\pm0.5}$	1.210	$72.20_{\pm2.6}$	$81.24_{\pm1.7}$	1.629	$84.02_{\pm0.2}$	$91.56_{\pm0.4}$	1.418	$65.95_{\pm1.1}$	$69.43_{\pm4.5}$	1.426
MTL	$82.91_{\pm1.6}$	$80.67_{\pm0.4}$	1.289	$69.46_{\pm1.8}$	$81.12_{\pm0.8}$	1.616	$83.92_{\pm0.3}$	$92.66_{\pm0.6}$	1.355	$65.37_{\pm1.6}$	$73.31_{\pm5.2}$	1.418

Table 2: **Results on the Research Paper stream.** We report log perplexity of MLM and the performance of downstream models fine-tuned from the final checkpoint of the pretrained model ($t = 4$). Performance of the best performing CL algorithm is marked bold.

Does lifelong pretraining help retain knowledge across different domain corpora? We first examine whether task-specific or lifelong pretraining improves performance over domain-specific downstream tasks. Comparing Task-Specific LMs with RoBERTa-base in Table 2, we notice consistent performance improvements, especially on Biomedical and Computer Science domains (D_1, D_2). We also see Sequential Pretraining could consistently outperform RoBERTa-base. However, the comparison between Sequential Pretraining and Task Specific LMs are mixed: on D_1, D_2, D_3, Sequential Pretraining could outperform Task-Specific LMs only except MNER; while on the earliest biomedical domain (D_1), Sequential Pretraining achieves substantially lower performance. From Figure 4, we see the performance of Sequential Pretraining on Chemprot and RCT (from D_1) drops significantly from $t = 1$ to 4. The results imply lifelong pretraining allows later domains to benefit from knowledge transfer from earlier domains, but the performance on earlier domains is limited because of forgetting.

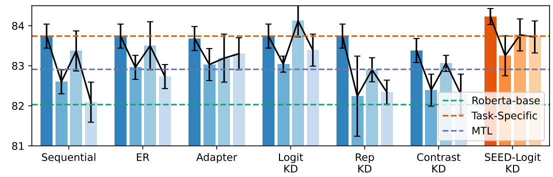
Does continual learning algorithms help retain knowledge in sequential pretraining? Next, we compare different kinds of CL algorithms and investigate the effect of CL algorithms in alleviating forgetting and improving knowledge transfer. Table 2 shows that Online-EWC slightly improves MLM perplexity compared to Sequential PT, but brings no improvement to the fine-tuning performance. We hypothesize that regularization directly in the parameter space as in Online-EWC is not effective when the parameter space is very high dimensional. Adapter improves downstream task F1 scores on the bio-medical domain (D_1) by 1.2% and 0.8%, but does not outperform Sequential Pretraining in other domains (similarly for Simple Layer Expansion approach), likely because a great portion of the model is kept frozen.

$\lvert M \rvert$, k	Chemprot	RCT	ACL-ARC	SciERC	MLM-$D_{1,2}$
$100k, 10$	82.73	79.98	72.50	81.64	1.737/1.857
$100k, 100$	82.06	78.64	71.97	81.62	1.599/1.789
$10M, 10$	82.87	79.98	71.80	81.63	1.438/1.732

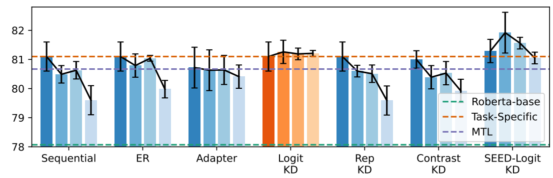
Table 3: Downstream task and MLM performance of f_T under different memory sizes $\lvert M \rvert$ and the frequency of replay k (replaying every k steps of training) in ER.

In contrast, the memory-replay based approach (ER) allows training the full parameters of the model and has been shown to be highly effective in continual learning of classification tasks (Wang et al., 2019; Chaudhry et al., 2019). However, we surprisingly find that ER could hardly improve over Sequential Pretraining except D_1. A similar pattern can be found in the MLM perplexity. We hypothesize that the positive effect of example replay has diminished because of the overfitting to the memory examples. Table 3 summarizes the effect of tuning hyperpameters in ER. When we reduce the frequency of replay (from every 10 steps to 100 steps), the MLM performance improves, which implies reduced overfitting; however, the performance of downstream task performance does not improve. When we increase the size of the memory $\lvert M \rvert$ from $100k$ to $10M$, the MLM perplexity also improves; still, there are still no improvements in downstream tasks. It may imply ER itself is not an effective approach for continual pretraining.

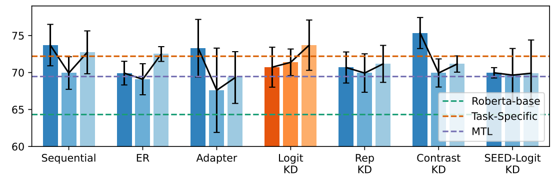
Unlike ER, distillation approaches utilize richer information such as output logits or representation similarity to preserve past knowledge. We find either Logit KD or SEED-Logit KD to be most effective depending on the task, while Rep-KD and Contrastive-KD are less effective. The best performing distillation approach improves F1 over Sequential Pretraining on downstream tasks from



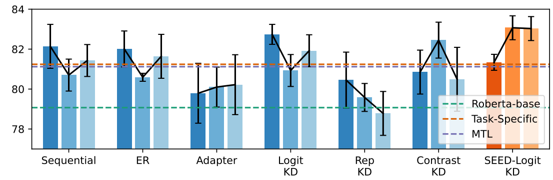
(a) Chemprot



(b) RCT-Sample



(c) ACL-ARC



(d) SciERC

Figure 4: **Performance evolution of downstream models.** Models are fine-tuned from checkpoints of lifelong pretrained LMs at different time steps t. For Chemprot and RCT-Sample from D_1, we use $t \in \{1, 2, 3, 4\}$; while for ACL-ARC and SciERC from D_2, $t \in \{2, 3, 4\}$. Methods achieving the best performance at the end of training ($t = 4$) is highlighted.

D_1, D_2 at least by 1.0%. However, performance on D_3, D_4, which come later in the data stream, does not improve over Sequential Pretraining, possibly because the distillation loss term makes the model rigid in obtaining new knowledge.

What is the gap between lifelong pretraining and multi-task learning across all the domains? Multi-Task Learning refers to the offline training paradigm, which retrain PTLMs over all corpora ($D_{1..t}$) each time a new corpus D_t becomes available. We examine whether lifelong pretraining is comparable to multi-task pretraining in terms of performance. From Table 2 and Figure 4, we see Sequential Pretraining in general underperforms MTL except for the final domain. However, certain CL approaches, such as Logit-Distillation, could improve over MTL on all downstream tasks from the first and the second domain. We speculate the reason is that continual learning naturally provides a curriculum (Xu et al., 2020; Shi et al., 2015) to models where each individual task is easier to learn.

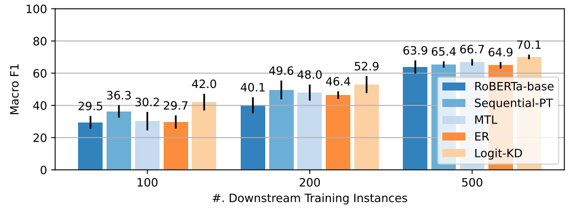


Figure 5: **Performance of downstream models with various number of training examples**, exemplified with SciERC. The models are fine-tuned from the final pretrained model (f_4).

The results have a positive implication that lifelong pretraining is not only more computationally efficient and requires less storage of past data, but may also improve the performance of pretraining.

Does lifelong pretraining make models more data efficient? In Table 5, we further examine the performance of final pretrained models under different amounts of training examples. We include full results in Appendix B. We find in general, performance improvements are more significant in the low-resource setup.

Computational Costs. We analyze computational costs of different CL algorithms and present additional experiments with controlled computational costs. We find additional computational cost is necessary for performance improvement of distillation-based CL. However, it is not possible to trade performance simply by investing more computation budget with arbitrary CL algorithms. We leave detailed discussions in Appendix F.

4.3 Temporal Data Stream

We conduct analysis on pretraining PTLM on chronologically-ordered tweet corpora, to understand whether lifelong pretraining helps adaptation to the latest data and improves temporal generalization ability. The results are summarized in Table 4.

Will LMs be outdated? We compare the performance of Task-Specific (2014) to the Task-Specific models pretrained on the year of downstream datasets (noted as Task-Specific (Latest)) and notice consistent improvements in downstream tasks in 2018 and 2020 (first two columns in Table 4). Sequential Pretraining could also outperform the Task-Specific (2014) model. It verifies that language models may get outdated, which can be addressed by task-specific or lifelong pretraining over the latest corpora.

Does lifelong pretraining help improve the downstream model's performance on latest data? We show that downstream model's performance over

Years	2018 (D_3)	2020 (D_4)	2014 (D_1) $\rightarrow$ 2020 (D_4)	2016 (D_2) $\rightarrow$ 2020 (D_4)
Hashtag Prediction				
RoBERTa-base	$48.08_{\pm1.0}$	$56.42_{\pm0.2}$	$39.31_{\pm2.7}$	$42.23_{\pm2.7}$
Sequential PT	$56.79_{\pm0.5}$	$59.85_{\pm0.4}$	$44.00_{\pm1.1}$	$49.87_{\pm1.8}$
ER	$56.93_{\pm0.1}$	$59.56_{\pm1.7}$	$43.31_{\pm0.2}$	$50.72_{\pm0.6}$
Logit-KD	$\mathbf{58.21}_{\pm0.5}$	$60.52_{\pm0.2}$	$44.26_{\pm0.9}$	$50.92_{\pm0.8}$
Contrast-KD	$57.94_{\pm0.4}$	$59.54_{\pm0.3}$	$45.22_{\pm0.1}$	$\mathbf{52.14}_{\pm1.1}$
SEED-KD	$56.87_{\pm0.2}$	$59.71_{\pm0.2}$	$43.39_{\pm0.4}$	$49.62_{\pm1.0}$
SEED-Logit-KD	$57.75_{\pm0.4}$	$\mathbf{60.74}_{\pm0.6}$	$\mathbf{45.35}_{\pm0.6}$	$51.56_{\pm0.7}$
Task-Specific (2014)	$56.16_{\pm0.6}$	$59.59_{\pm0.3}$	$44.34_{\pm0.6}$	$49.26_{\pm0.7}$
Task-Specific (Latest)	$56.61_{\pm0.4}$	$59.87_{\pm0.6}$	$43.44_{\pm0.5}$	$49.41_{\pm1.1}$
MTL	$57.89_{\pm0.4}$	$59.95_{\pm0.3}$	$44.04_{\pm0.3}$	$50.37_{\pm0.3}$
Emoji Prediction				
RoBERTa-base	$25.71_{\pm0.1}$	$24.42_{\pm0.2}$	$12.02_{\pm0.4}$	$13.24_{\pm0.2}$
Sequential PT	$29.30_{\pm0.1}$	$27.69_{\pm0.1}$	$14.20_{\pm0.2}$	$16.08_{\pm1.4}$
ER	$29.50_{\pm0.1}$	$27.75_{\pm0.1}$	$14.36_{\pm0.4}$	$16.82_{\pm0.3}$
Logit-KD	$29.77_{\pm0.1}$	$27.80_{\pm0.1}$	$14.20_{\pm0.3}$	$16.28_{\pm1.1}$
Contrast-KD	$29.48_{\pm0.2}$	$27.72_{\pm0.3}$	$\mathbf{14.42}_{\pm0.3}$	$\mathbf{17.52}_{\pm0.1}$
SEED-KD	$\mathbf{30.12}_{\pm0.1}$	$27.66_{\pm0.1}$	$14.36_{\pm0.1}$	$16.97_{\pm0.4}$
SEED-Logit-KD	$29.98_{\pm0.1}$	$\mathbf{27.84}_{\pm0.2}$	$14.36_{\pm0.1}$	$16.97_{\pm0.3}$
Task-Specific (2014)	$28.94_{\pm0.0}$	$26.98_{\pm0.2}$	$13.39_{\pm0.2}$	$15.14_{\pm0.2}$
Task-Specific (Latest)	$29.06_{\pm0.2}$	$27.19_{\pm0.1}$	$13.00_{\pm0.2}$	$14.48_{\pm0.3}$
MTL	$29.52_{\pm0.2}$	$27.47_{\pm0.0}$	$14.07_{\pm0.2}$	$16.64_{\pm0.2}$

Table 4: **Results on temporal data stream.** We show fine-tuning performance over years 2018 and 2020 (D_3, D_4) and the Temporal generalization from 2014 or 2016 to 2020 data ($D_1 \rightarrow D_4$, $D_2 \rightarrow D_4$) on Twitter Hashtag and Emoji prediction datasets. Models are fine-tuned from the final pre-trained model f_T. Full results are included in Appendix C.

later data (D_3, D_4) can be improved over Task-Specific models when continual learning algorithms are applied. From the first two columns of Table 4, we see Logit-KD and SEED-KD improve Hashtag prediction score over data of years 2018 and 2020. SEED-Logit KD further improves prediction F1 on Emoji prediction. Note that these findings are in contrast to the research paper stream, where CL algorithms do not improve performance in the latest domain D_4. The reason can be the higher similarity between domains in the tweet corpora making the knowledge transfer easier, which is further discussed in Appendix H.

Does lifelong pretraining improve temporal generalization? Temporal generalization evaluates downstream performance over latest test data when fine-tuned over outdated training data. We show lifelong pretraining brings clear improvement to temporal generalization. From Table 4, we see even Sequential Pretraining could improve over the model pretrained merely on the year 2020 data (Task-Specific (2020)) consistently. We find performance further improves with CL algorithms applied. SEED-Logit-KD performs best in general on crossyear hashtag prediction tasks. In crossyear emoji prediction, we find Contrast-KD and SEED-KD perform best. We also find that SEED-Logit-KD could slightly outperform Logit-KD.

5 Related Works

Domain and Temporal Adaptation of Language Models. Gururangan et al. (2020) study adaptation of PTLMs to domain-specific corpora. Arumae et al. (2020) study algorithms to mitigate forgetting in original PTLMs, but does not investigate forgetting that happens over a sequence of domains. Maronikolakis and Schütze (2021); Röttger and Pierrehumbert (2021); Luu et al. (2021) proposes sequential pretraining over domains or emerging data, but did not investigate CL algorithms. Several recent studies have demonstrated the necessity of adapting LMs over time (Lazaridou et al., 2021) while specifically focusing on factual knowledge (Dhingra et al., 2021; Jang et al., 2021).

Continual Learning Algorithms in NLP. Continual learning in NLP has mainly been studied for classification tasks. An effective approach is to utilize a number of stored past examples (de Masson d'Autume et al., 2019; Wang et al., 2020), or pseudo examples (*e.g.,* the ones generated with a PTLM (Sun et al., 2020; Kanwatchara et al., 2021)). Recent extensions of the algorithm (Chuang et al., 2020) perform knowledge distillation with generated pseudo examples. Other lines of works focus on regularization over the sentence representations (Wang et al., 2019; Huang et al., 2021; Liu et al., 2019a) or directly merging models in the parameter space (Matena and Raffel, 2021). Model expansion-based approaches (Liu et al., 2019a; Pfeiffer et al., 2021), including learning domain specific expert models (Gururangan et al., 2021), are also actively studied.

6 Conclusion

In this paper, we formulated the lifelong language model pretraining problem and constructed two data streams associated with downstream datasets. We evaluated knowledge retention, adaptation to the latest data, and temporal generalization ability of continually pretrained language models. Our experiments show distillation-based approaches being most effective in these evaluation setups. A limitation of the work is that it has not been fully addressed whether there exists a variant of distillation-based CL approach that consistently outperforms Logit-KD. Based on the current observation, we conclude the performance of different KD approaches for CL is highly task-dependent. It asks for more future works into continual learning algorithms within the proposed problem setup.

References

Kristjan Arumae, Qing Sun, and Parminder Bhatia. 2020. An empirical investigation towards efficient multi-domain language model pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4854–4864, Online. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Abdul Hameed Azeemi and Adeel Waheed. 2021. Covid-19 tweets analysis through transformer language models. *ArXiv*, abs/2103.00199.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. 2021. Co2l: Contrastive continual learning. *ArXiv*, abs/2106.14413.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2914–2924, Online. Association for Computational Linguistics.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, D. Gillick, Jacob Eisenstein, and William W. Cohen. 2021. Time-aware language models as temporal knowledge bases. *ArXiv*, abs/2106.15110.

Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, L. Zhang, Yezhou Yang, and Zicheng Liu. 2021. Seed: Self-supervised distillation for visual representation. *ArXiv*, abs/2101.04731.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821.

Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2782–2788. IJCAI/AAAI Press.

Suchin Gururangan, Michael Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2021. Demix layers: Disentangling domains for modular language modeling. *ArXiv*, abs/2108.05036.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages

8342–8360, Online. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc-Alexander Najork. 2021. Dynamic language models for continuously evolving content. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2018. Lifelong learning via progressive distillation and retrospection. In *ECCV*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Xiaolei Huang and Michael J. Paul. 2018. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699, Melbourne, Australia. Association for Computational Linguistics.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2736–2746, Online. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2110.03215*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul, and Peerapon Vateekul. 2021. Rational LAMOL: A rationale-based lifelong learning framework. 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)*, pages 2942–2953, Online. Association for Computational Linguistics.

Angeliki Lazaridou, A. Kuncoro, E. Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, Kris Cao, Tomás Kociský, Susannah Young, and P. Blunsom. 2021. Assessing temporal generalization in neural language models. *NeurIPS*.

Zhizhong Li and Derek Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947.

Tianlin Liu, Lyle Ungar, and João Sedoc. 2019a. Continual learning for sentence representations using conceptors. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3274–3279, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment.

Antonis Maronikolakis and Hinrich Schütze. 2021. Multidomain pretrained language models for green NLP. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 1–8, Kyiv, Ukraine. Association for Computational Linguistics.

Michael Matena and Colin Raffel. 2021. Merging models with fisher-weighted averaging. *ArXiv*, abs/2111.09832.

Martin Müller, Marcel Salathé, and Per Egil Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *ArXiv*, abs/2005.07503.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Elsa A Olivetti, Jacqueline M Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M Hiszpanski. 2020. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4):041317.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542. IEEE Computer Society.

Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7605–7617, Online. Association for Computational Linguistics.

Anthony V. Robins. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.*, 7:123–146.

Paul Röttger and J. Pierrehumbert. 2021. Temporal adaptation of bert and performance on downstream document classification: Insights from social media. *Findings of EMNLP*.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4535–4544. PMLR.

Yangyang Shi, Martha Larson, and Catholijn M. Jonker. 2015. Recurrent neural network language model adaptation with curriculum learning. *Comput. Speech Lang.*, 33:136–154.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. LAMOL: language modeling for lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

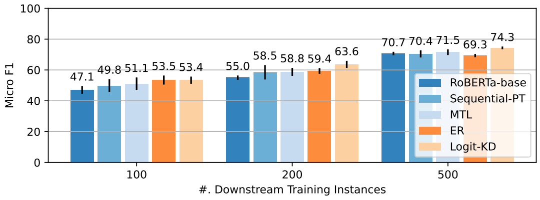
Jens Vindahl. 2016. Chemprot-3.0: a global chemical biology diseases mapping.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806, Minneapolis, Minnesota. Association for Computational Linguistics.

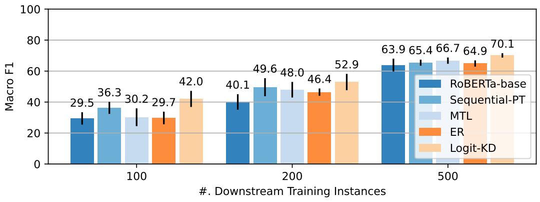
Zirui Wang, Sanket Vaibhav Mehta, Barnabas Poczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 535–548, Online. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 460–470, Online. Association for Computational Linguistics.



(a) Chemprot



(b) SciERC

Figure 6: Performance of downstream models with various number of training examples. The models are fine-tuned from the final pretrained model (f_4).

A Detailed Experiment Settings

We use a linearly decreasing learning rate initialized with 5e-4 on the research paper stream and 3e-4 on the tweet stream. On the research paper stream, we train the model for 8,000 steps in the first task, and 4,000 steps in the subsequent tasks. On the tweet stream, we train the model for 8,000 steps in all tasks. We hold out 128,000 sentences from each corpus to evaluate MLM performance. As the size of pretraining corpora is large, during training, each training example is visited only once. We use the masked language modeling perplexity over held-out validation sets of the pretraining corpora as the metrics for hyperparameter tuning. Common hyperparameters such as learning rate and batch sizes are tuned with Task-specific models with the first task. Hyperparameters that are specific to continual learning algorithms, such as the scale of the distillation loss, is tuned using the first two domains in the stream according to the MLM performance over validation sets. The weight of the distillation term α is set as 1.0 for logit distillation and 0.1 for other distillation algorithms. By default, we replay or perform distillation with a mini-batch of examples from the replay memory every 10 training steps in ER and Distillation-based CL approaches. We use the huggingface transformers library `https://github.com/huggingface/transformers` for implementation.

B Low-Resource Fine-Tuning

Figure 6 summarizes the performance of fine-tuned models from the final model checkpoint ($t = 4$) using different amount of downstream training examples. We see on Chemprot and SciERC, the benefit of Sequential Pretraining over RoBERTa-base is more significant in low-resource fine-tuning setups. Whenever Seqential Pretraining outperforms RoBERTa-base, we notice Logit-KD could further improve over Sequential Pretraining.

Task	2014	2016	2018	2020
		Hashtag Prediction		
RoBERTa-base	$56.65_{\pm 0.6}$	$45.50_{\pm 2.1}$	$48.08_{\pm 1.0}$	$56.42_{\pm 0.2}$
Sequential PT	$59.00_{\pm 0.1}$	$54.28_{\pm 0.3}$	$56.79_{\pm 0.5}$	$59.85_{\pm 0.4}$
ER	$59.00_{\pm 0.1}$	$54.90_{\pm 0.2}$	$56.93_{\pm 0.1}$	$59.56_{\pm 1.7}$
Adapter	$58.76_{\pm 0.7}$	$52.55_{\pm 1.5}$	$54.34_{\pm 1.7}$	$59.01_{\pm 1.0}$
Logit-KD	$60.93_{\pm 0.5}$	$55.96_{\pm 0.2}$	$58.21_{\pm 0.5}$	$60.52_{\pm 0.2}$
Rep-KD	$60.47_{\pm 0.1}$	$51.77_{\pm 2.6}$	$55.79_{\pm 1.4}$	$59.80_{\pm 0.2}$
Contrast-KD	$60.72_{\pm 0.6}$	$55.85_{\pm 0.0}$	$57.94_{\pm 0.4}$	$59.54_{\pm 0.3}$
SEED-KD	$58.82_{\pm 0.4}$	$54.55_{\pm 0.5}$	$56.87_{\pm 0.2}$	$59.71_{\pm 0.2}$
SEED-Logit-KD	$61.28_{\pm 0.2}$	$55.59_{\pm 0.5}$	$57.75_{\pm 0.4}$	$60.74_{\pm 0.6}$
Task-Specific (2014)	$61.62_{\pm 0.3}$	$55.38_{\pm 0.6}$	$56.16_{\pm 0.6}$	$59.59_{\pm 0.3}$
Task-Specific (Latest)	$59.91_{\pm 0.3}$	$55.47_{\pm 1.0}$	$56.61_{\pm 0.4}$	$59.87_{\pm 0.6}$
MTL	$60.51_{\pm 0.3}$	$55.16_{\pm 1.6}$	$57.89_{\pm 0.4}$	$59.95_{\pm 0.3}$
		Emoji Prediction		
RoBERTa-base	$28.73_{\pm 0.2}$	$26.86_{\pm 0.2}$	$25.71_{\pm 0.1}$	$24.42_{\pm 0.2}$
Sequential PT	$32.69_{\pm 0.2}$	$30.55_{\pm 0.3}$	$29.30_{\pm 0.1}$	$27.69_{\pm 0.1}$
ER	$32.88_{\pm 0.2}$	$30.52_{\pm 0.2}$	$29.50_{\pm 0.1}$	$27.75_{\pm 0.1}$
Adapter	$32.15_{\pm 0.2}$	$29.85_{\pm 0.0}$	$28.72_{\pm 0.0}$	$26.80_{\pm 0.3}$
Logit-KD	$33.08_{\pm 0.3}$	$30.88_{\pm 0.1}$	$29.77_{\pm 0.1}$	$27.80_{\pm 0.1}$
Rep-KD	$32.71_{\pm 0.2}$	$30.51_{\pm 0.2}$	$29.45_{\pm 0.1}$	$27.27_{\pm 0.2}$
Contrast-KD	$32.90_{\pm 0.1}$	$31.01_{\pm 0.1}$	$29.48_{\pm 0.2}$	$27.72_{\pm 0.3}$
SEED-KD	$32.91_{\pm 0.1}$	$30.84_{\pm 0.3}$	$30.12_{\pm 0.1}$	$27.66_{\pm 0.1}$
SEED-Logit-KD	$33.28_{\pm 0.1}$	$31.17_{\pm 0.1}$	$29.98_{\pm 0.1}$	$27.84_{\pm 0.2}$
Task-Specific (2014)	$33.37_{\pm 0.2}$	$30.54_{\pm 0.3}$	$28.94_{\pm 0.0}$	$26.98_{\pm 0.2}$
Task-Specific (Latest)	$32.31_{\pm 0.0}$	$29.83_{\pm 0.5}$	$29.06_{\pm 0.2}$	$27.19_{\pm 0.1}$
MTL	$32.78_{\pm 0.1}$	$30.54_{\pm 0.0}$	$29.52_{\pm 0.2}$	$27.47_{\pm 0.0}$

Table 5: Full performance on Twitter Hashtag prediction and Emoji prediction, fine-tuned from the pre-trained model in the final time step.

C Full Results over the Tweet Stream

Tables 5 and 6 summarize full results over the Tweet stream. Compared to the table 4 in the main text, we add downstream performance over data from years 2014 and 2016 (D_1, D_2), and temporal generalization from year 2014 to 2020 ($D_1 \rightarrow D_4$).

D Dataset Details

The research paper stream consists of full text of 6.6M, 12.1M, 7.8M, and 7.5M research papers from the S2ORC (Lo et al., 2020) dataset. We evaluate downstream fine-tuning performance on two in-domain datasets for each research area: Chemprot relation exaction dataset (Vindahl, 2016) and RCT abstract sentence role labeling dataset (Dernoncourt and Lee, 2017) for the bio-medical domain; ACL-ARC citation intent classification dataset (Jurgens et al., 2018) and SciERC relation extraction dataset (Luan et al., 2018) for the

Task	2014 → 2020	2016 → 2020	2018 → 2020
Crossyear Hashtag Prediction			
RoBERTa-base	$39.31_{\pm 2.7}$	$42.23_{\pm 2.7}$	$37.19_{\pm 2.1}$
Sequential PT	$44.00_{\pm 1.1}$	$49.87_{\pm 1.8}$	$46.63_{\pm 0.9}$
ER	$43.31_{\pm 0.2}$	$50.72_{\pm 0.6}$	$46.27_{\pm 0.4}$
Adapter	$42.61_{\pm 0.5}$	$48.00_{\pm 1.6}$	$42.63_{\pm 0.9}$
Logit-KD	$44.26_{\pm 0.9}$	$50.92_{\pm 0.8}$	$46.84_{\pm 1.0}$
Rep-KD	$42.48_{\pm 0.2}$	$50.38_{\pm 1.5}$	$42.23_{\pm 0.2}$
Contrast-KD	$45.22_{\pm 0.1}$	$52.14_{\pm 1.1}$	$47.47_{\pm 0.8}$
SEED-KD	$43.39_{\pm 0.4}$	$49.62_{\pm 1.0}$	$46.37_{\pm 0.8}$
SEED-Logit-KD	$45.35_{\pm 0.6}$	$51.56_{\pm 0.7}$	$47.74_{\pm 0.3}$
Task-Specific (2014)	$44.34_{\pm 0.6}$	$49.26_{\pm 0.7}$	$45.09_{\pm 0.7}$
Task-Specific (2020)	$43.44_{\pm 0.5}$	$49.41_{\pm 1.1}$	$44.34_{\pm 0.4}$
- *4x steps*	$44.34_{\pm 0.6}$	$51.78_{\pm 0.7}$	$44.69_{\pm 0.7}$
MTL	$44.04_{\pm 0.3}$	$50.37_{\pm 0.3}$	$44.31_{\pm 0.0}$
Crossyear Emoji Prediction			
RoBERTa-base	$12.02_{\pm 0.4}$	$13.24_{\pm 0.2}$	$18.67_{\pm 0.1}$
Sequential PT	$14.20_{\pm 0.2}$	$16.08_{\pm 1.4}$	$21.06_{\pm 0.9}$
ER	$14.36_{\pm 0.4}$	$16.82_{\pm 0.3}$	$21.57_{\pm 0.1}$
Adapter	$13.53_{\pm 0.2}$	$15.68_{\pm 0.3}$	$20.64_{\pm 0.1}$
Logit-KD	$14.20_{\pm 0.3}$	$16.28_{\pm 1.1}$	$21.29_{\pm 1.0}$
Rep-KD	$13.89_{\pm 0.1}$	$16.03_{\pm 0.3}$	$20.86_{\pm 0.2}$
Contrast-KD	$14.42_{\pm 0.3}$	$17.52_{\pm 0.1}$	$21.43_{\pm 0.1}$
SEED-KD	$14.36_{\pm 0.1}$	$16.97_{\pm 0.4}$	$21.88_{\pm 0.3}$
SEED-Logit-KD	$14.36_{\pm 0.1}$	$16.97_{\pm 0.3}$	$21.62_{\pm 0.1}$
Task-Specific (2014)	$13.39_{\pm 0.2}$	$15.14_{\pm 0.2}$	$20.79_{\pm 0.3}$
Task-Specific (2020)	$13.00_{\pm 0.2}$	$14.48_{\pm 0.3}$	$19.30_{\pm 0.2}$
- *4x steps*	$12.90_{\pm 0.4}$	$14.85_{\pm 0.3}$	$19.83_{\pm 0.2}$
MTL	$14.07_{\pm 0.2}$	$16.64_{\pm 0.2}$	$20.94_{\pm 0.7}$

Table 6: Temporal generalization performance on Twitter Hashtag prediction datasets fine-tuned from the final pretrained model. `Year 1`→`Year 2` indicates the hashtag prediction model is fine-tuned on data in year `Year 1`, and evaluated on test data in `Year 2`.

computer science domain; relation extraction over Synthesis procedures (Mysore et al., 2019) and named entity recognition over material science papers (MNER) (Olivetti et al., 2020) for material science domain; keyphrase classification and hyponym classification after filtering out physics papers for the physics domain (Augenstein et al., 2017). We report micro-averaged F1 on Chemprot, RCT, MNER datasets following the evaluation metrics in the original work, and report macro-averaged F1 on all other datasets. We use the official data splits for all datasets except for RCT, where we employ a low-resource training setup following Gururangan et al. (2020).

The pretraining corpora for the tweet stream consist of 25M tweets in each year. For downstream tasks, we use a separate set of 1M tweets from each year to construct multi-label hashtag prediction (Gong and Zhang, 2016) datasets and single-label emoji prediction datasets (Barbieri et al., 2018). We replace user names to special tokens. For Hashtag prediction, the label space consists of tweets containing 200 most frequent hashtags in each year. We independently sample 500 tweets per label (hashtag) as training, validation and test sets, which results $10k$ examples in each of the data splits. For emoji prediction, we construct 20-way single-label emoji prediction datasets for each year following Barbieri et al. (2018) with the 1M held out tweets. We sample 5,000 tweets per emoji in each split, resulting in balanced datasets of the same size as the hashtag prediction datasets.

E Details of Continual Learning Algorithms

E.1 Contrastive Distillation

During continual pretraining, in addition to the language model pretraining objective, we add a unsupervised contrastive learning objective, namely the SimCSE (Gao et al., 2021) objective, so that the similarity in the sentence representation better reflects the semantic similarity in the sentence. We use the l^2-normalized representation of the start-of-sequence token at the final layer as the sentence representation, noted as $\boldsymbol{h}$. Then, we distill the intra-batch representational similarity from the previous model f_{t-1} to the current model f_t. Given a mini-batch of N examples $\boldsymbol{x}$, we compute the representational dot-product similarity matrix between normalized sentence representations $\boldsymbol{h}$ between each pair of examples with f_{t-1} and f_t, noted as $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$, where each element $\mathbf{B}_{ij}$ is,

$$\mathbf{B}_{ij} = \frac{\exp(\boldsymbol{h}_i \cdot \boldsymbol{h}_j / \tau)}{\sum_{k=1..N} \exp(\boldsymbol{h}_i \cdot \boldsymbol{h}_k / \tau)} \quad (1)$$

where τ is a temperature hyperparameter. We specify a temperature $\tau_t = 0.05$ for the teacher model f_{t-1} and a temperature τ_s for the student model $f_t = 0.01$. We compute the cross-entropy between $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$ as the distillation loss,

$$\ell_{\text{distill}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{B}_{ij}^{t-1} \log \mathbf{B}_{ij}^{t} \quad (2)$$

E.2 SEED Distillation

SEED distillation proposed by (Fang et al., 2021) has a similar spirit as the contrastive distillation with differences in the examples used for computing similarity matrices computes. The algorithm distills representational similarity *between the batch and a large set of other examples*, maintained in an example queue Q. As the number of target examples K can be much larger than the batch size, it allows distillation of richer information by regularizing similarities. During pretraining, the method maintains a fixed-size queue

Q to cache examples from the current domain D_t. Given a mini-batch of training examples $\boldsymbol{x}$, it computes cosine similarity between each pair of examples within the batch $\boldsymbol{x}$ and Q with f_{t-1} and f_t, resulting in two similarity matrices $\mathbf{B}^{t-1}$, $\mathbf{P}^y \in \mathbb{R}^{|B|\times|Q|}$. Similar to the contrastive distillation, the distillation loss is the cross-entropy between two similarity matrices $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$ computed in the same way as Eq. 2.

F Analysis and Controlled Experiments of Computational Costs

Computational cost is a crucial matter for online continual learning systems. In this section, we analyze the computational costs of continual learning algorithms and perform controlled experiments of computational costs.

We quantify computational costs with the total number of *forward (C_f) and backward (C_b)* computations ($C = C_f + C_b$) over the PTLMs, which is easy to control; in practice, we find the wall clock time of training was approximately linear to C. We summarize the number of forward and backward passes and the wall clock time of training in Table 7. In the visit of b batches from the training stream, Sequential PT performs b forward and backward passes respectively over the PTLM, resulting in $C = 2b$. Experience replay further replays 1 batch of examples every r steps over the training stream, which results in $C = (2 + 2/k)b$. In our main experiments, r is set to 10 (Sec. 3.3). Logit-Distill and Rep-Distill require one additional forward pass over a frozen PTLM to compute the target of distillation, resulting in $C = (3 + 3/k)b$. Distillation algorithms that perform contrastive learning with SimCSE (*i.e.* SEED-Distill and SEED-Logit-Distill) additionally require one forward and backward pass using the same batch of examples with different dropout masks. Therefore, for SEED-Logit-Distill, $C = (5 + 5/k)b$.

To control the number of forward and backward passes, we present approaches to compensate the lower computation costs compared to Distillation algorithms and one approach to shrink the computational cost of distillation algorithms: (1) for Sequential PT, we train the models for 1.2 times more steps so that $C = 2.4b$, noted as Sequential PT$_{b'=1.2b}$; (2) for ER, we increase the replay frequency k to 5 from the default setup 10, so that $C = 2.4b$. We also decrease the cost of Logit-KD and SEED-Logit-KD by reducing the frequency of distillation from every 1 batch to every r' =10 steps, while still replaying and distilling knowledge over 1 batch of memory examples every 10 training steps. This results in $C_f = (1 + 2/k + 1/k')b$ and $C_b = (1 + 1/k)b$, where $C = 2.4b$ when both r and r' are 10. The approach is referred to as Sparse Logit-KD. Finally, for SEED-Logit-KD, we remove the SimCSE loss from training and perform sparse distillation similar to Sparse-Logit-KD, which also results in $C = 2.4b$.

The performance of the models is presented in Table 8. We notice that at the end of pretraining, increasing the number of training steps in Sequential PT by 1.2 times does not lead to performance boost on the latest domain (D_4), while the performance over tasks from earlier domains (Chemprot, ACL-ARC, SciERC) slightly dropped, possibly due to increased forgetting. For ER, we notice replaying only slightly more frequently (ER$_{k=5}$) than the default setup (k=10) greatly increased the perplexity of MLM, implying significantly increased overfitting to the memory; while the performance differences of downstream tasks compared to the default ER is mixed. When we decrease the replay frequency of distillation, the performance on Logit-KD and SEED-KD also decreased and does not outperform ER.

The results show additional computation costs can be necessary for continual learning algorithms such as Logit-KD and SEED-Logit-KD. However, the results also show that there is no simple trade-off between computational cost and performance. We have seen that it is not always beneficial to increase the number of training steps over the emerging data, as it increases forgetting in earlier domains. Similarly, increasing the frequency of replay may lead to significant overfitting to the replay memory. Investigating into more effective continual learning algorithms, despite increased computation costs, allows us to obtain performance improvement that cannot be simply traded with more computation with arbitrary continual learning algorithms. We leave more thorough studies into this topic as future work.

G Experiments with BERT on Tweet Stream After 2019

In this section, we present an additional set of experiments on BERT-base (Devlin et al., 2019) model, which is originally pretrained with Wikipedia articles before 2019, with Tweets only after 2019. The

Method	#. of Forward	#. of Backward	#. Total	#. Total (k=10)	Wall Time$_{4k}$
Main results					
Sequential PT	b	b	$2b$	$2b$	4.0×10^4 sec.
ER	$(1+1/k)b$	$(1+1/k)b$	$(2+2/k)b$	$2.2b$	4.2×10^4 sec.
Logit-Distill	$(2+2/k)b$	$(1+1/k)b$	$(3+3/k)b$	$3.3b$	6.9×10^4 sec.
SEED-Logit-Distill	$(3+3/k)b$	$(2+2/k)b$	$(5+5/k)b$	$5.5b$	9.7×10^4 sec.
Additional Controlled Experiments					
Sequential PT$_{b'=1.2b}$	$1.2b$	$1.2b$	$2.4b$	$2.4b$	4.4×10^4 sec.
ER$_{k=5}$	$1.2b$	$1.2b$	$2.4b$	$2.4b$	4.4×10^4 sec.
Sparse Logit-KD	$1.3b$	$1.1b$	$2.4b$	$2.4b$	4.4×10^4 sec.
Sparse SEED-Logit-KD$_{\backslash contrast}$	$1.3b$	$1.1b$	$2.4b$	$2.4b$	4.5×10^4 sec.

Table 7: Number of forward and backward passes over PTLMs and wall clock time of different approaches. The number of forward and backwards passes are computed over visits of b batches from the training data stream, where k is the frequency of replay. The wall clock time is calculated over $4k$ steps of training (which is the number of training steps of a single domain in the Research Paper stream) excluding the first domain, as no replay or distillation happens while learning the first domain. We use 2 Quadro RTX 8000 GPUs for training each model. In the additional controlled experiments (described in Appendix. F), we control the total number of forward and backward passes of different approaches. This also yields approximately the same wall clock time for approaches.

Task	D_1 - Biomedical			D_2 - Computer Science			D_3 - Materials Science			D_4 - Physics		
Dataset	Chemprot	RCT-Sample	MLM	ACL-ARC	SciERC	MLM	MNER	Synthesis	MLM	Keyphrase	Hyponym	MLM
Sequential Pretraining	$82.09_{\pm 0.5}$	$79.60_{\pm 0.5}$	1.654	$72.73_{\pm 2.9}$	$81.43_{\pm 0.8}$	1.807	$83.99_{\pm 0.3}$	$92.10_{\pm 1.0}$	1.590	$67.57_{\pm 1.0}$	$74.68_{\pm 4.4}$	1.381
Sequential Pretraining$_{b'=1.2b}$	$81.68_{\pm 0.5}$	$79.80_{\pm 0.4}$	1.656	$70.57_{\pm 3.0}$	$80.89_{\pm 1.2}$	1.793	$83.65_{\pm 0.3}$	$92.16_{\pm 0.7}$	1.578	**67.61**$_{\pm 1.4}$	$75.03_{\pm 4.1}$	**1.379**
ER	$82.73_{\pm 0.3}$	**79.98**$_{\pm 0.3}$	1.737	$72.50_{\pm 1.0}$	$81.64_{\pm 1.1}$	1.857	$83.99_{\pm 0.4}$	**92.65**$_{\pm 0.4}$	1.621	$66.11_{\pm 1.1}$	$72.82_{\pm 4.3}$	1.391
ER$_{k=5}$	**83.00**$_{\pm 0.1}$	$79.79_{\pm 0.4}$	1.913	$69.85_{\pm 2.6}$	**82.30**$_{\pm 1.2}$	2.049	**84.03**$_{\pm 0.2}$	$91.60_{\pm 0.6}$	1.721	$65.55_{\pm 0.4}$	$75.64_{\pm 3.2}$	1.418
Logit-KD-Sparse	$82.80_{\pm 0.4}$	$79.80_{\pm 0.5}$	1.476	$73.31_{\pm 2.0}$	$81.19_{\pm 0.8}$	1.744	$83.84_{\pm 0.4}$	$92.29_{\pm 0.7}$	**1.472**	$66.65_{\pm 0.7}$	**77.27**$_{\pm 7.1}$	1.385
SEED-KD-Sparse	$82.51_{\pm 0.4}$	$79.52_{\pm 0.5}$	**1.474**	**73.70**$_{\pm 3.4}$	$81.92_{\pm 0.8}$	**1.741**	$83.96_{\pm 0.3}$	$92.20_{\pm 1.0}$	1.480	$64.75_{\pm 1.1}$	$71.29_{\pm 3.6}$	1.381

Table 8: Performance of distillation algorithms in the setup of controlled computational costs.

Task	2019-1	2019-2	2020-1	2020-2
	Hashtag Prediction			
BERT-base	$46.38_{\pm 0.4}$	$48.05_{\pm 0.8}$	$41.67_{\pm 1.0}$	$69.00_{\pm 0.5}$
Sequential PT	$50.46_{\pm 0.1}$	$52.70_{\pm 0.7}$	$46.49_{\pm 1.0}$	$71.63_{\pm 0.7}$
ER	$49.90_{\pm 0.4}$	$52.33_{\pm 0.6}$	$46.84_{\pm 0.3}$	$71.67_{\pm 0.4}$
Logit-KD	$50.19_{\pm 0.9}$	**53.70**$_{\pm 0.4}$	**47.64**$_{\pm 0.4}$	**72.44**$_{\pm 0.5}$
SEED-Logit-KD	**50.79**$_{\pm 0.8}$	$52.84_{\pm 0.5}$	$46.04_{\pm 0.4}$	$72.24_{\pm 0.6}$

Table 9: Hashtag prediction performance of continually pretrained BERT models over tweets after 2019.

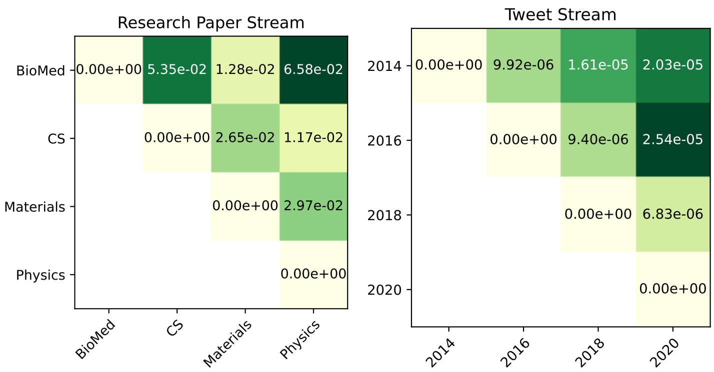
Task	2019-1→2019-2	2019-1→2020-1	2019-1→2020-2
	Hashtag Prediction		
BERT-base	$40.19_{\pm 0.3}$	$41.00_{\pm 0.6}$	$40.85_{\pm 0.8}$
Sequential PT	$43.30_{\pm 0.7}$	**48.60**$_{\pm 2.1}$	$44.07_{\pm 0.8}$
ER	$42.96_{\pm 0.9}$	$46.07_{\pm 1.6}$	$44.26_{\pm 0.7}$
Logit-KD	**43.35**$_{\pm 1.6}$	$46.91_{\pm 0.5}$	**45.03**$_{\pm 0.2}$
SEED-Logit-KD	$43.56_{\pm 0.4}$	$45.77_{\pm 0.7}$	$43.76_{\pm 0.5}$

Table 10: Temporal generalization performance of Hashtag prediction models fine-tuned from continually pretrained BERT models over tweets after 2019.

training corpora $D_{1..4}$ consist of tweets from the first half of 2019, the second half of 2019, the first half of 2020, and the second half of 2020 respectively. We accordingly construct hashtag prediction and cross-year hashtag prediction datasets. The performance of downstream tasks fine-tuned from the final pretrained model is presented in Table 9. We see Sequential PT clearly outperforms BERT-base which is not continually pretrained, and that Logit-KD generally improves hashtag prediction performance compared to Sequential PT except on the first half of 2019. We hypothesize the small temporal gap between $D_{1..4}$ makes improvements less significant than our main experiment setup. We present temporal generalization performance in cross-year hashtag prediction tasks in Table 10. Similarly, Logit-KD improves over Sequential PT in two out of three cross-year hashtag prediction setups.

H Analysis of Data Streams

In this section, we provide further analysis about the created research paper stream and the tweet stream. We measure cosine distances d_v of vocabulary distributions between each pair of different domains ($D_{1..4}$) and summarize the results in Figure 7. The results indicate that the Tweet stream has a magnitude smaller vocabulary distribution gap between domains, which is in the scale of $1e^{-5}$, compared to the research paper stream, which is in the scale of $1e^{-2}$. On the Tweet stream, we see the differences of vocabulary distributions align with the temporal gap between domains. On the



(a) Research Paper Stream

(b) Tweet Stream

Figure 7: Cosine distance of vocabulary distributions between each pair of datasets in two data streams.

research paper stream, we find some domains to be more similar than others. For example, Biomedical (D_1) and Material Science domains (D_3) have larger similarity in their vocabulary distributions, which explains general downstream performance increase on D_1 after the model is pretrained on D_3 (Fig. 4 (a,b)).

The differences in vocabulary distribution explain inconsistency in results between two data streams, specifically, whether lifelong pretraining improves downstream model performance on the latest domain, as we mentioned in Sec. 4.3. Other than this, our main findings, such as the effect of distillation-based CL algorithms on reducing forgetting, are consistent over two datasets with such significant differences in their changes of vocabulary distribution. We believe it implies the conclusions in this paper should be reliable in diverse data streams.

I Ethic Risks

We would like to note that, in practice, continually pretrained models over real-world data streams would require identification and removal of biased contents from pretraining corpora, which may affect the prediction of downstream models. As PTLMs are continuously updated, the bias in earlier pretraining may have a profound negative impact. In future works, it is preferable to develop algorithms to "forget" certain biased knowledge from language models. We further note that any data released in this paper, especially the tweet stream, should only be used for research purposes.