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]Amazon Inc.
{xisenjin,xiangren}@usc.edu
{dejiaoz, henghui, xiaoweiw, 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 from a new domain that deviates from what the PTLM was initially trained on, or newly emerged data that contains out-of-distribution information. 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) to analyze its ability of acquiring new knowledge and preserving learned knowledge. Our experiments show continual learning algorithms improve knowledge preservation, with logit distillation being the most effective approach. We further show that continual pretraining improves generalization when training and testing data of downstream tasks are drawn from different time steps, but do not improve when they are from the same time steps. 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 over a range of natural language processing tasks (Liu et al., 2019b; Brown et al., 2020). However, in an open-ended real-world setup, the distribution of data may constantly shift from that of pretraining corpus, because of new data domains being introduced (Gururangan et al., 2020), or the evolving nature of language itself over time (Lazaridou et al., 2021). For example, one may extend LMs pretrained over certain science domains (Beltagy et al., 2019) to new science domains. As another example, one may expect PTLMs over Tweets (Nguyen et al., 2020) to continually improve so that it solves tasks over up-to-date tweets. These practical scenarios ask whether we can continuously update PTLMs whenever new corpora become available efficiently. Towards this goal, we propose to study lifelong (continual) pretraining of language models: the language model is sequentially pretrained over several corpora without (or with only a little) re-training over previously seen corpora.

A number of recent works study LM adaptation to a new domain that is different from the original pretraining domain. They show performance gains on downstream tasks in the new domain (Gururangan et al., 2020; Yao et al., 2021), as well as retaining knowledge learned in general domain when certain regularizations are applied (Arumae et al., 2020). However, continual (sequential) pretraining over multiple distinct corpora is less studied — it is unclear how the LM can continually acquire and accumulate knowledge from different corpora to benefit downstream tasks in a new domain or on more recent data, and whether the LM can retain the knowledge learned from earlier corpora to preserve decent performance on seen domains.

Furthermore, lifelong LM pretraining poses unique challenges to existing continual learning (CL) methods. For memory-based CL approaches (Wang et al., 2019; Chaudhry et al., 2019), a small number of training examples may not be sufficient to represent past knowledge that should be retained, as the size of the pretraining corpus is typically huge. Moreover, there is no guarantee that a CL approach that retains pretraining performance also retains fine-tuning performance well, as the latter solely depends on the learned representations instead of the masked language modeling

†Work partially done while Xisen Jin was interning at Amazon Inc.

‡Work done while Shang-Wen Li was working at Amazon Inc. The current affiliation is Facebook AI.

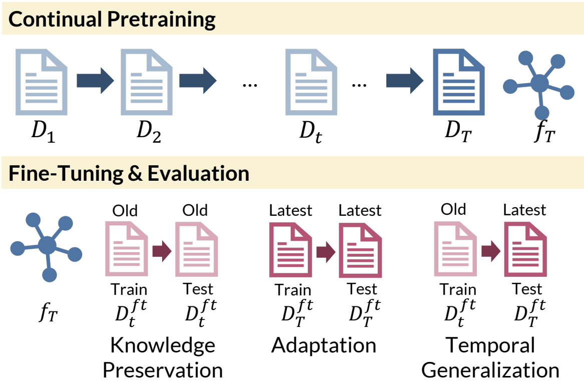

Figure 1: Training, evaluation setups, and metrics of lifelong language model pretraining. The model sequentially visits each corpus, and is fine-tuned on downstream datasets related to pretraining. We evaluate knowledge preservation and adaptation to new data with downstream task performance on old and latest domains respectively. Besides, we evaluate temporal generalization where training/test examples are drawn from different time steps.

prediction head.

In this paper, we formulate a Lifelong Language Model Pretraining (LPT) task to simulate the aforementioned challenges and to provide a testbed for studying them. The setup is illustrated in Figure 1. We construct two text data streams to simulate two common scenarios: 1) a domain-incremental text stream consists of academic papers published in three research fields, where corpus of each domain arrives sequentially; 2) a temporal tweet stream consists of tweets collected from four different years, where corpora of tweets arrives in chronological order. We keep track of the downstream task performance of fine-tuned models and focus on the evaluation of knowledge preservation on the domain-incremental text stream; while for the chronologically ordered stream, we focus on adaptation to latest data and temporal generalization where training and testing distributions are from different time steps in downstream tasks.

We try to figure out the best working approach to continually pretrain language models. We conduct systematic evaluation of existing CL algorithms, spanning over memory-based, distillation-based and adapter-based approaches, to establish strong baselines. We further provide extensive analysis on distillation based approaches, integrating various knowledge distillation techniques to continual learning, to dissect the research question - which "dark knowledge" should be retained to best improve overall pretraining performance.

Our results show that, in our continual pretraining setup, (1) continual learning algorithms are effective for knowledge preservation, where distillation over output logits performs the best; (2) continual pretraining improves temporal generalization, but does not improve adaptation to latest data on the tweet stream. We expect our problem formulation, evaluation setup, methods and analysis could inspire more future study towards continual pretraining of language models.

2 Related Works

2.1 Domain and Temporal Adaptation of Language Models

Language models may require a round of pretraining over domain-specific corpus before being applied to domain-specific downstream tasks (Gururangan et al., 2020). Within this intermediate pretraining process, Arumae et al. (2020) study algorithms to mitigate forgetting in original pretraining language model weights. However, they do not investigate forgetting that happens over a sequence of intermediate domains. To our best knowledge, Maronikolakis and Schütze (2021) is the only work that proposes to study sequential pretraining over a number of domains, but the work did not investigate continual learning algorithms.

Several recent studies have demonstrated the necessity of learning language models over real-world dynamically evolving data streams (Lazaridou et al., 2021), where some of them specifically focus on methods to accumulate and update factual knowledge (Dhingra et al., 2021; Jang et al., 2021). A notable work by Röttger and Pierrehumbert (2021) studies whether continual pretraining over social media streams improves its performance on downstream stream tasks that requires up-to-date knowledge, but did not investigate into continual learning algorithms.

2.2 Continual Learning Algorithms in NLP

Continual learning in NLP has mainly been studied for classification tasks. The algorithms span over data-based, model regularization-based, and model expansion-based approaches. Data-based approaches involve memory-based approaches, which utilize a small number of stored past examples, or pseudo examples (*e.g.,* the ones generated with a pretrained language model) to alleviate forgetting. Among this line of work, MbPA (de Masson d'Autume et al., 2019), and meta-MbPA (Wang et al., 2020) maintain an episodic memory during training and regularly retrain on stored examples. LAMOL (Sun et al., 2020) jointly train a language model to generate past training examples for future replay. There have been recent extensions of the algorithm, such as L2KD (Chuang et al., 2020), which performs knowledge distillation from the past saved model checkpoint with generated examples, instead of simply retraining over them like MbPA; Rational LAMOL (Kanwatchara et al., 2021) applies critical freezing according to human provided rationals or unsuperivsingly generated ones. Sentence Embedding Alignment (Wang et al., 2019) stores sentences and their representations and tries to ensure a simple linear mapping that maps from old representations to new representations given a batch of sentences. Huang et al. (2021) propose an information disentanglement and regularization-based approach which disentangles and applies separate regularization to task-agnostic and task-specific representations. Model regularization-based approaches perform regularization directly in the weight space. The algorithms are not intensively studied for NLP tasks. EWC (Kirkpatrick et al., 2017) and Online-EWC (Schwarz et al., 2018) are regularization approaches broadly studied as baselines. Liu et al. (2019a) apply conceptor-based continual learning algorithms to sentence representation learning. Model expansion-based approaches separate task-specific parameters from irrelevant ones and freeze shared parameters to prevent catastrophic forgetting. While sometimes not explicitly studied, Adapter-based approaches (Wang et al., 2021) could be applied to continual learning. The algorithms learn a single adapter per task without interference with pretrained weights or other tasks; at the same time, knowledge captured in previous tasks can be effectively fused to new tasks (Pfeiffer et al., 2021).

3 Problem Formulation

In this section, we formulate the problem of continual language model pretraining, introduce the datasets creation process and set up evaluation protocols.

3.1 Lifelong Pretraining of PTLMs

We assume a language model f visits a stream of total T unlabeled text corpus $D_{1..T} = \{D_1, D_2, ...D_T\}$, indexed by t. In our case, f is a

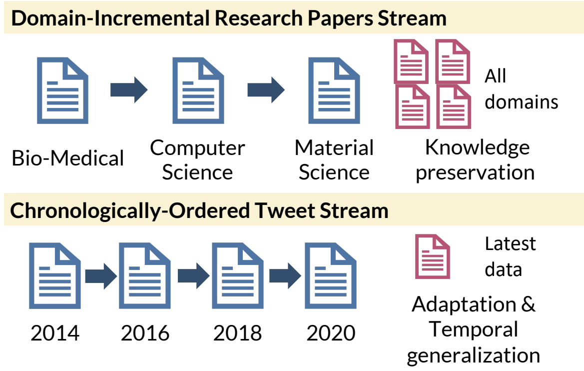

Figure 2: Two data streams created for studying lifelong language model pre-training. We focus on evaluating knowledge preservation on the domain-incremental research papers stream; we focus on adaptation to the latest data and temporal generalization on the chronologically ordered tweet stream.

RoBERTa-base model (Liu et al., 2019b), and is initialized with pretrained RoBERTa weights. Following prior continual learning literature, we refer to each corpus D_t as a pretraining "task", and t as the time step. The data distribution $P(D_t)$ evolves with the time step t. We assume a language model, noted as f, is sequentially trained over each task, while it is prohibited to access or retrain over the full corpora from earlier tasks. The constraint is common in practice, *e.g.*, because of privacy issues to store earlier datam and time/computation budget for retraining over all previous corpora. We use f_t to denote the model right after learning the task D_t. The model f is fine-tuned over downstream tasks $\{D^{\text{FT}}_{t,j}\}$, where t denotes that the downstream task is related to D_t and j is the index of the downstream tasks. In the fine-tuning process, the model has no access the pretraining corpus $D_{1..T}$.

3.2 Data Streams & Downstream Datasets

We create two data streams to simulate the practical use cases of continual pretraining, namely a domain-incremental research paper stream and a chronologically-ordered tweet stream. Figure 2 illustrates the created data streams and the evaluation focus of two data streams.

Domain Incremental Research Paper Stream. The research paper data stream consists of full text of research papers from bio-medical, computer science, and material science domains. The dataset is filtered from the S2ORC dataset[1] accord-

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

ing to the "majority fields of study" labels associated with each paper. The papers from three different domains are presented sequentially to the model. We evaluate downstream fine-tuning performance over two datasets for each paper domain: Chemprot relation exaction dataset (Vindahl, 2016) and sampled-RCT abstract sentence role labeling dataset (Dernoncourt and Lee, 2017) for the biomedical domain; ACL-ARC citation intent classification dataset (Jurgens et al., 2018) and SciERC relation extraction dataset (Luan et al., 2018) for the computer science domain; and relation extraction datasets over Synthesis procedures (Mysore et al., 2019) and named entity recognition over material science papers (MNER) (Olivetti et al., 2020). We report micro-averaged F1 on Chemprot and sample-RCT datasets following the original work, and report macro-averaged F1 on all other datasets.

Chronologically-Ordered Tweet Stream. The tweet data stream consists of tweets from years 2014, 2016, 2018, and 2020 respectively grabbed by the Archive Team[2]. The tweets from four years are presented sequentially to the language model. We pre-processed the tweets by converting user names and urls to the special USER token and URL token respectively. We hold out 1M tweets from each year to create downstream hashtag prediction datasets: for all tweets containing at least one hashtag, we remove all hashtags from the tweets and let the model predict the hashtags. Because multiple hashtags may exist in a tweet, we formulate the problem as a multi-label classification problem, and report the label ranking average precision (LRAP) score. We truncate the label space to 200 most frequent hashtags (after manual filtering out hashtags that are likely to be automatically generated, *e.g.*, #nowplaying), and independently sample up to 500 training examples per label for training, validation and test sets, so that the dataset is approximately balanced.

3.3 Evaluation Protocols

Our evaluation protocols are designed to benchmark practical usage of approaches in two axes: knowledge preservation of earlier data as well as adaptation and generalization ability to latest data.

Knowledge Retention & Forgetting. The research papers data stream is representative of the use case where models should continually accumulate knowledge over all previously seen domains. One major challenge in such use case is the catastrophic forgetting, *i.e.,* significant performance degrade over seen domains while learning new domains. To evaluate knowledge preservation, we fine-tune a model f_t over downstream tasks $D^{\text{FT}}_{t',j}$ from all previously seen domains. We use $s(f_t, D^{\text{FT}}_{t',j})$ to denote the performance of the model fine-tuned from f_t on $D^{\text{FT}}_{t',j}$. We track the performance of the fine-tuned model on a dataset $D^{\text{FT}}_{t',j}$ over the pretraining time steps t ($t' \leq t \leq T$) and inspect the change of the performance. The forgetting is indicated by the performance degrade over the time step t.

Adaption & Generalization to New Data. Over chronologically ordered data streams such as the tweet stream, it is more crucial that the model performs well over the up-to-date data instead of outdated data. For this purpose, we focus on evaluating the performance on the downstream dataset $D^{\text{FT}}_{T,j}$ with the latest time step. We additionally evaluate the *temporal generalization* ability of f_T, where distributional gaps exist between the training and testing data in downstream tasks: the model is fine-tuned on outdated training examples, but evaluated on the latest test examples. Note that the label space of hashtag prediction can be different between years. In our experiments, for evaluation of temporal generalization, we use the label set of the earlier year, and collect examples examples from the held-out 1M tweets in the latest year that have at least one of the hashtags in the earlier year. We sub-sample 10,000 examples as the test set.

4 Methods

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

4.1 Task Specific Baselines

We consider several simple baselines before continual learning algorithms. `Roberta-base` cor-

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

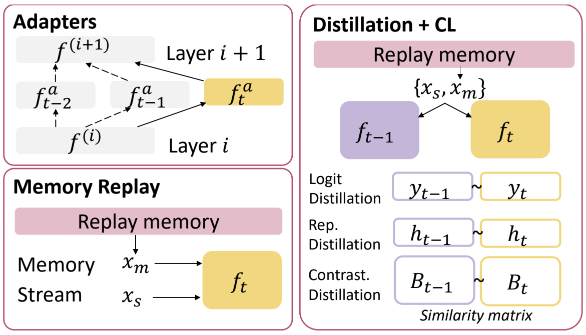

Figure 3: Comparison of adapter, memory replay, and distillation-based continual learning algorithms. We use f_t^a to represent adapters created for the task t. $\boldsymbol{x}_m$ and $\boldsymbol{x}_s$ note for the mini-batch of examples drawn from the memory and the data stream respectively.

responds to not pretraining on any of the domain-specific corpus, which indicates a lower bound of performance. We also train one task specific model for each domain, noted as `Task-Specific Models`. As there can be either positive or negative knowledge transfer from other tasks, there is chance that task specific models outperform continually pretrained models.

4.2 Adapter-based Approaches

Adapter-based approaches add small "adapter" layers between layers of transformers per task (Wang et al., 2021; Houlsby et al., 2019). We follow the adapter design by Pfeiffer et al. (2021). The adapters are implemented as Multi Layer Perceptron (MLP), so that output of each transformer layer is wrapped as $h_{i+1} = \texttt{ADAPTER}(\texttt{TRANS}(h_i))$, where h_i is the model input from the previous transformer layer. Here, the transformer model is frozen, and one adapter is trained per pretraining task. Because updates of adapters are not interfered by other tasks, the approach could perfectly mitigate forgetting.

During fine-tuning, we apply Adapter Fusion (Pfeiffer et al., 2021), which learns to weigh the outputs from all learned adapters, allowing knowledge transfer from other domains. In both cases, we unfreeze and fine-tune the base model weights of RoBERTa-base.

4.3 Memory Replay Approaches

We apply experience replay (ER) (Chaudhry et al., 2019), which alleviates forgetting by maintaining a fixed-size replay memory and regularly draw examples from the memory to replay. We maintain a memory of 100,000 examples, which are approximately 0.30% and 0.15% of training examples in the research paper stream and the tweet stream respectively. We populate the memory when the pretraining over the current task finishes, and randomly select examples from the current task to store in the memory. We ensure the memory always contains a balanced number of examples from all previously seen tasks. We sample a mini-batch from the memory to perform replay every 10 training steps.

4.4 Distillation-based Approaches

Distillation-based approaches store one previous model checkpoint of the model (noted as f_{t-1}) and apply knowledge distillation techniques to distill "dark knowledge" from f_{t-1} to the current model f_t (Li and Hoiem, 2018; Rebuffi et al., 2017; Hou et al., 2018). We explore various options of knowledge distillation algorithms. Not only do we expect to improve performance with these algorithms, we also hope to analyze which part of the "dark knowledge" in pretrained models are most important for knowledge preservation and adaptation.

Overall Workflow. We build distillation approaches on top of memory replay approaches. Each time the model receives a mini-batch of stream examples $\boldsymbol{x}_s$ or a mini-batch of memory examples $\boldsymbol{x}_m$, we obtain model outputs with f_{t-1} and f_t. We compute a distillation loss that penalizes the differences between the model outputs, and jointly optimize it with the masked language modeling 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 the KL divergence between $\boldsymbol{y}_t$ and $\boldsymbol{y}_{t-1}$.

Representation Distillation. We also consider minimizing the representational deviation of sentences between previous and current models. 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. We compute the ℓ^2 distance between $\boldsymbol{h}_{t-1}^{1:N}$ and $\boldsymbol{h}_t^{1:N}$ as the distillation loss.

Contrastive Distillation. We further consider intra-batch representational similarity as additional "dark knowledge" for distillation. The approach is

Task	Task 1 - Biomedical						Task 2 - Computer Science				Task 3 - Materials Science	
Dataset	Chemprot			RCT-Sample			ACL-ARC		SciERC		MNER	Synthesis
Evaluated After	Task 1	Task 2	Task 3	Task 1	Task 2	Task 3	Task 2	Task 3	Task 2	Task 3	Task 3	Task 3
Roberta-base	$82.03_{\pm 0.7}$	$82.03_{\pm 0.7}$	$82.03_{\pm 0.7}$	$78.07_{\pm 0.7}$	$78.07_{\pm 0.7}$	$78.07_{\pm 0.7}$	$64.32_{\pm 2.8}$	$64.32_{\pm 2.8}$	$79.07_{\pm 1.6}$	$79.07_{\pm 1.6}$	$83.15_{\pm 0.3}$	$91.25_{\pm 0.6}$
Naive	$\mathbf{83.74}_{\pm 0.3}$	$82.60_{\pm 0.3}$	$83.37_{\pm 0.5}$	$\mathbf{81.10}_{\pm 0.5}$	$80.49_{\pm 0.3}$	$80.63_{\pm 0.3}$	$\mathbf{73.71}_{\pm 2.8}$	$69.93_{\pm 2.2}$	$82.14_{\pm 1.1}$	$80.70_{\pm 0.8}$	$83.34_{\pm 0.3}$	$92.72_{\pm 1.0}$
ER	$\mathbf{83.74}_{\pm 0.3}$	$82.96_{\pm 0.3}$	$83.50_{\pm 0.6}$	$\mathbf{81.10}_{\pm 0.5}$	$80.79_{\pm 0.4}$	$81.04_{\pm 0.1}$	$69.92_{\pm 1.6}$	$69.09_{\pm 2.1}$	$82.01_{\pm 0.9}$	$80.59_{\pm 0.2}$	$83.79_{\pm 0.4}$	$\mathbf{93.20}_{\pm 0.2}$
Adapter	$83.68_{\pm 0.3}$	$83.03_{\pm 0.4}$	$83.19_{\pm 0.6}$	$80.72_{\pm 0.7}$	$80.63_{\pm 0.7}$	$80.64_{\pm 0.5}$	$73.28_{\pm 3.9}$	$67.60_{\pm 5.7}$	$79.79_{\pm 1.5}$	$80.10_{\pm 1.0}$	$\mathbf{83.94}_{\pm 0.4}$	$90.82_{\pm 3.3}$
Rep-KD	$\mathbf{83.74}_{\pm 0.3}$	$82.24_{\pm 1.0}$	$82.90_{\pm 0.3}$	$\mathbf{81.10}_{\pm 0.5}$	$80.60_{\pm 0.2}$	$80.51_{\pm 0.3}$	$70.68_{\pm 2.1}$	$69.93_{\pm 2.6}$	$80.45_{\pm 1.4}$	$79.58_{\pm 0.7}$	$83.89_{\pm 0.4}$	$92.16_{\pm 0.6}$
Contrast-KD	$83.38_{\pm 0.3}$	$82.39_{\pm 0.4}$	$83.06_{\pm 0.2}$	$81.00_{\pm 0.3}$	$80.39_{\pm 0.4}$	$80.53_{\pm 0.4}$	$75.34_{\pm 2.1}$	$69.94_{\pm 1.9}$	$80.85_{\pm 1.1}$	$\mathbf{82.45}_{\pm 0.9}$	$83.21_{\pm 0.3}$	$92.05_{\pm 0.4}$
Logit-KD	$\mathbf{83.74}_{\pm 0.3}$	$\mathbf{83.04}_{\pm 0.2}$	$\mathbf{84.12}_{\pm 0.4}$	$\mathbf{81.10}_{\pm 0.5}$	$\mathbf{81.26}_{\pm 0.4}$	$\mathbf{81.19}_{\pm 0.2}$	$70.72_{\pm 2.7}$	$\mathbf{71.38}_{\pm 1.8}$	$\mathbf{82.74}_{\pm 0.5}$	$80.93_{\pm 0.8}$	$83.54_{\pm 0.2}$	$92.73_{\pm 1.0}$
Task-Specific LM		$83.74_{\pm 0.3}$			$81.10_{\pm 0.5}$		$72.20_{\pm 2.6}$		$81.24_{\pm 1.7}$		$84.02_{\pm 0.2}$	$91.56_{\pm 0.4}$

Table 1: Performance of downstream models fine-tuned from continually pre-trained language models on the multi-domain research paper stream. Performance of the best performing CL algorithm is marked bold.

Task	Task 1 Biomedical			Task 2 Computer Science		Task 3 Mat. Science
Evaluated After	Task1	Task 2	Task 3	Task 2	Task 3	Task 3
Roberta-base	1.993	1.993	1.993	2.153	2.153	2.117
Naive	**1.210**	1.548	1.359	1.604	1.853	1.355
ER	**1.210**	1.514	1.356	1.607	1.907	1.361
Adapter	1.437	1.437	1.437	1.728	1.728	1.641
Logit-KD	**1.210**	**1.311**	**1.274**	1.666	**1.707**	1.380
Rep-KD	**1.210**	1.489	1.345	**1.601**	1.868	**1.352**
Contrast-KD	1.216	1.535	1.372	1.624	1.888	1.363
Task-Specific LM		1.210		1.629		1.418

Table 2: Log-perplexity of masked language modeling on the multi-domain research paper data stream.

modified from (Cha et al., 2021), which is originally studied for supervised image classification tasks. The approach consists of two components: learning a representation space with unsupervised contrastive learning, and regularize the change of representation similarity between examples.

During training, in addition to the language model pretraining objective, we also learn a representation space with SimCSE (Gao et al., 2021), so that the similarity in the representation better reflects the semantic similarity in the sentence. In SimCSE, the positive pair is defined as the representations of the same sentence using different dropout masks, while the negative pairs in the mini-batch consist of representations of different sentences within the mini-batch. Let z, z' be two random dropout masks. The SimCSE loss is written as,

$$\ell_{\text{con}} = -\alpha \log \frac{e^{\text{cos-sim}(\boldsymbol{h}_i^{z_i}, \boldsymbol{h}_i^{z_j})/\tau}}{\sum_{j=1}^{N} e^{\text{cos-sim}(\boldsymbol{h}_i^{z_i}, \boldsymbol{h}_j^{z_j'})/\tau}} \quad (1)$$

where N is the number of examples in the mini-batch, $\boldsymbol{h}_i^{z_i}$ is the sentence representation of an instance i after dropout, τ is a temperature hyperparameter, α is weighting hyperparameter to the masked language modeling loss. Following the original work, we use the representation of the start-of-sequence (<s>) token as the sentence representation.

To perform distillation, given a mini-batch of N examples $\boldsymbol{x}$, we compute the representational similarity matrix between each pair of examples with f_{t-1} and f_t, noted as $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$. We softmax-normalize the similarities in the second dimension. Then, we compute the cross-entropy between $\mathbf{B}^{t-1}$ and $\mathbf{B}^t$ as the distillation loss,

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

5 Experiments

In this section, we summarize our findings on the domain-incremental research paper stream and the chronologically-ordered tweet stream.

5.1 Experiment Settings.

We use the RoBERTa-base model throughout the experiments. All models are initialized with RoBERTa-base weights before continual pretraining. We set the maximal sequence length to 128, which is a result of a trade-off between computational overhead and performance. We use a batch size of 128 and accumulate gradients for 16 steps during continual pretraining, yielding an effective batch size of 2,048. 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. In contrastive distillation approaches, we set α=1 and τ=0.05.

5.2 Results on the Research Paper Stream

Table 1 summarizes the fine-tuning performance over the research paper data stream. For each downstream task related to a domain D_t, we fine-tune the language models at the time step t and afterward, as indicated in the header "evaluated after" in Table 1. We also report the performance of language modeling perplexity in Table 2.

Performance Without CL Algorithms. We first compare the performance of Naive CL, which performs online training without applying CL algorithms, with RoBERTa-base. We see, at the final task (Task 3), Naive CL could consistently outperform RoBERTa-base over all six downstream datasets, validating the benefit of continual pretraining despite the issue of forgetting. The improvement of language modeling perplexity is clearer.

Effect of Adapter Approaches. We compare the performance of Adapter with Naive CL. We notice that the approach does not consistently outperform Naive CL in downstream performance despite that the algorithms are immune to forgetting. The results imply the modeling capacity of adapter models is a bottleneck of performance.

Effect of Experience Replay. By comparing experience replay (ER) and Naive CL, we found surprisingly that ER could not consistently improve downstream performance, despite that it was highly effective for continual learning of classification tasks (Wang et al., 2019; Chaudhry et al., 2019). At the final task, there were minor improvements in the downstream performance over the Task 1 - Biomedical domain over Naive CL, but no improvements over the Task 2 - Computer Science domain. The similar conclusion holds for the language modeling perplexity. We hypothesis that the positive effect of example replay has diminished because of the overfitting to the memory examples.

Effect of Distillation Approaches. From the performance of Logit Distillation, Representation Distillation, and Contrastive Distillation, we find that only Logit Distillation could clearly reduce masked language modeling perplexity over the baselines. Representation and Contrastive Distillation do not reduce masked language modeling perplexity, which is understandable, because these two algorithms directly operate over the sentence representations, leaving the masked language model prediction head unregularized. However, we further find that only Logit distillation could consistently improve downstream task performance over Naive CL on downstream tasks in earlier domains when evaluated at the end of pretraining. The results indicate that Logit Distillation is a highly effective algorithm, and also implies that input logits are useful dark knowledge for knowledge preservation. We note the current results do not necessarily imply that the stability of representations and representational similarity is irrelevant to alleviating forgetting. In future works, we may further improve the loss term of distillation applied in Representation and Contrastive distillation.

Tweet year	**2014**	**2016**	**2018**	**2020**
Roberta-base	$56.65_{\pm 0.6}$	$45.50_{\pm 2.1}$	$48.08_{\pm 1.0}$	$56.42_{\pm 0.2}$
Naive CL	$59.00_{\pm 0.1}$	$54.28_{\pm 0.3}$	$56.79_{\pm 0.5}$	$\mathbf{59.85}_{\pm 0.4}$
ER	$59.00_{\pm 0.1}$	$54.90_{\pm 0.2}$	$56.93_{\pm 0.1}$	$59.56_{\pm 1.7}$
Logit-KD$_{\text{first}}$	$\mathbf{59.31}_{\pm 2.4}$	$\mathbf{55.12}_{\pm 0.5}$	$\mathbf{56.99}_{\pm 0.2}$	$59.77_{\pm 0.5}$
Task-Specific LM	$59.91_{\pm 0.3}$	$55.47_{\pm 1.0}$	$56.61_{\pm 0.4}$	$59.87_{\pm 0.6}$

Table 3: Performance on Twitter Hashtag prediction datasets, fine-tuned from the pre-trained model in the final time step.

Tweet years	**2014 → 2020**	**2016 → 2020**	**2018 → 2020**
Roberta-base	$39.31_{\pm 2.7}$	$42.23_{\pm 2.7}$	$37.19_{\pm 2.1}$
Naive CL	$44.00_{\pm 1.1}$	$49.87_{\pm 1.8}$	$\mathbf{46.63}_{\pm 0.9}$
ER	$43.31_{\pm 0.2}$	$\mathbf{50.72}_{\pm 0.6}$	$46.27_{\pm 0.4}$
Logit-KD$_{\text{first}}$	$\mathbf{44.37}_{\pm 1.3}$	$49.98_{\pm 0.7}$	$46.10_{\pm 0.7}$
Task-Specific LM (2020)	$43.44_{\pm 0.5}$	$49.41_{\pm 1.1}$	$44.34_{\pm 0.4}$

Table 4: Temporal generalization performance on Twitter Hashtag prediction datasets fine-tuned from the final pre-trained 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`.

Comparison to Task-Specific Models. We find that Task-Specific LMs, which are trained independently over each domain-specific corpus, achieve comparable or sometimes better downstream performance than the best continually pretrained models. However, we argue that a clear advantage of continually pretrained models is that a single model can be applied to multiple domains.

5.3 Results on the Chronologically Ordered Tweet Stream

Tables 3 and 4 summarizes the performance on the Twitter Hashtag prediction datasets from the year 2014 to year 2020, fine-tuned from the language model checkpoint at the end of pretraining. the We report the performance of a variant of Logit Distillation (which is the best performing approach on the research paper stream), namely the Logit-KD$_{\text{first}}$, which always uses the model checkpoint after the first task as the distillation teacher. Empirically, we find the performance outperforms standard logit distillation.

Effect of Continual Pretraining to Adaptation. As we mentioned in Sec. 3.3, over chronologically ordered data streams, the performance over the latest data is of higher importance. Therefore, in Table 3, we focus on the hashtag prediction performance in the year 2020. Unfortunately, we find continually pretrained models do not improve over task-specific models, which are only pretrained over the latest year. It may imply knowledge from earlier years is redundant for the hashtag prediction task given the latest data.

Effect of Continual Pretraining to Temporal Generalization. We further investigate whether continual learning algorithms improve temporal generalization, where the hashtags prediction models are fine-tuned on outdated training data (2014, 2016, 2018) but evaluated on the latest data (2020). From Table 4, we see continual pretraining almost always improve performance over Task-Specific LM. It implies pretraining data from earlier years are helpful for temporal generalization. However, we do not find a consistent trend within the performance of different continual learning algorithms: Logit-KD$_{\text{first}}$ performs the best in 2014→2020 generalization, while Naive CL performs the best in 2018→2020 generalization. The mixed results encourage future works to study more effective continual pretraining algorithms.

6 Conclusion

In this paper, we formulated the lifelong language model pretraining problem and constructed two data streams associated with downstream datasets. We constructed the domain-incremental research paper stream and the chronologically-ordered tweet stream, each of which is representative of a practical scenario of continual pretraining. We evaluate knowledge retention, adaptation to the latest data, and temporal generalization ability of continually pretrained language models. We evaluated a number of continual learning algorithms spanning over adapter-based approaches, memory replay- based approaches, and specifically focused on distillation-based approaches. Our experiments on the research paper stream demonstrate that continual learning algorithms are effective for persevering downstream task performance and language modeling perplexity in old domains, with Logit-Distillation being the single best working algorithm. On the tweet stream and hashtag prediction tasks, we find continual pretraining does not improve adaptation ability to the latest data, but brings moderate improvement to temporal generalization. Future works may continually improve 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.

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.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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 J. Y. Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *IJCNLP*.

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.

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

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.

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 *International Conference on Machine Learning*, pages 2790–2799. PMLR.

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*.

David Jurgens, Srijan Kumar, Raine Hoover, Daniel A. 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.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, Guillaume Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

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. Pitfalls of static language modelling. *ArXiv*, abs/2102.01951.

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.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*.

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.

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.

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.

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.

Paul Röttger and J. Pierrehumbert. 2021. Temporal adaptation of bert and performance on downstream document classification: Insights from social media. *ArXiv*, abs/2104.08116.

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.

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.

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.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. 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.

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 ACL*.