MAP: Memory-Augmented Pre-trained Language Model for Domain Tasks

Anonymous EMNLP submission

Abstract

Recently, domain-specific PLMs have been proposed to boost the task performance of specific domains (e.g., biomedical and computer science) by continuing to pre-train general PLMs with domain-specific corpora. However, this Domain-Adaptive Pre-Training (DAPT; Gururangan et al. (2020)) tends to forget the previous general knowledge acquired by general PLMs, which leads to a *catastrophic forgetting* phenomenon and sub-optimal performance. To alleviate this problem, we propose a new framework of **M**emory-**A**ugmented **P**re-trained Language Model (**MAP**), which augments the domain-specific PLM by a memory representation built from the frozen general PLM without losing any general knowledge. Specifically, we propose a new memory-augmented layer, and based on it, different augmented strategies are explored to build the memory representation and then adaptively fuse it into the domain-specific PLM. We demonstrate the effectiveness of MAP on various domains (biomedical and computer science publications, news, and reviews) and different kinds (text classification, QA, NER) of tasks, and the extensive results show that the proposed MAP can achieve SOTA results on all tasks. The source code will be publicly available upon publication.

1 Introduction

Pre-trained Language models (PLMs), such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b), have achieved promising performance on NLP tasks. Typically, these general models are firstly pre-trained on large unlabeled corpus and then directly fine-tuned on downstream tasks. However, there is an inherent gap in text distribution between unlabeled pre-training corpus and labeled task corpus, which leads to the distribution shift problem (Gururangan et al., 2020) and makes PLMs perform poorly on some domain tasks (Beltagy et al., 2019; Lee et al., 2020b). To address this shift problem, the domain-adaptive pre-training (DAPT) is proposed (Huang et al., 2019; Beltagy et al., 2019; Gururangan et al., 2020; Lee et al., 2020b) to further pretrain general PLMs on large-scale domain corpora, achieving better performance than general PLMs.

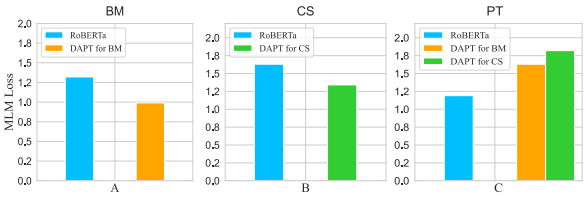

Figure 1: Masked LM (MLM) loss of RoBERTa on 50K randomly sampled documents from each domain before and after DAPT. Figure A and B denote the inference loss of general RoBERTa-base and domain-specific PLMs on the samples of biomedical (BM) and computer science (CS). Figure C means the loss of these models on the samples from the pre-training (PT) corpus of RoBERTa. We report the results of (Gururangan et al., 2020) and lower MLM loss is better.

Although DAPT can effectively learn the domain distribution of the target task, its continual pretraining process updates the parameters of general PLMs, which inevitably leads to partial general knowledge being forgotten. This *catastrophic forgetting* (Goodfellow et al., 2014; Li and Hoiem, 2016; Thompson et al., 2019) phenomenon is verified in Figure 1, where we observe that the domain-specific PLMs show better results than general PLMs on domain corpus, but perform worse on the general corpus. We argue that this forgotten knowledge is beneficial for domain-specific PLMs and should be used to improve their generalization ability on domain tasks.

To alleviate the *catastrophic forgetting*, we propose a simple yet effective memory-augmented framework named general **M**emory-**A**ugmented **P**re-trained model (**MAP**). In addition to the backbone domain-specific PLM, MAP introduces a new memory-augmented layer. It explicitly in-

corporates the representation built from a frozen general PLM as the memory to make the backbone model access the complete general knowledge. Then, a new proposed memory-attention within the memory-augmented layer enables the domain-specific PLM adaptively combine the memory representation and the domain-specific representation. Using the memory built from the frozen general PLM has two advantages: (1) frozen PLM never suffers from forgetfulness since the parameters remain unchanged (Levine et al., 2022); (2) it doesn't require additional training for the general PLM during fine-tuning. However, building and fusing memory into a backbone model is essentially a many-to-many scenario, where we need to choose which layer output of the general PLM as the memory representation, and which layer in the domain-specific PLM should be fused. Thus, we propose several memory-augmented strategies for better building then combining the memory representation into domain-specific PLM.

We evaluate our MAP on text classification, Question Answering (QA) and Name Entity Recognition (NER) tasks covering four domains: biomedical science, computer science, news, and reviews. Experimental results demonstrate that MAP outperforms existing baselines on all tasks. We compare different memory-augmented strategies, and the results show that the proposed chunk-based gated memory transfer strategy achieves the best results. In addition, for the memory representation building, we empirically find that the freezing way is better than the unfreezing one, which also has better training efficiency. Furthermore, we apply the proposed framework to a small-scale domain pretraining setting and find that MAP is also practical in achieving lower MLM loss. Our contributions are summarized below:

- We empirically find that forgotten general knowledge due to *catastrophic forgetting* can benefit the domain-specific downstream tasks since it can improve PLMs' generalization ability.
- We propose a novel MAP framework, which introduces several memory-augmented strategies to construct the memory representation from the frozen general PLM effectively and then fuse it into the domain-specific PLM by a new memory-augmented layer.
- We conduct extensive experiments on various domain-specific tasks including text classification, QA, and NER, the results demonstrating that our MAP outperforms existing baselines.

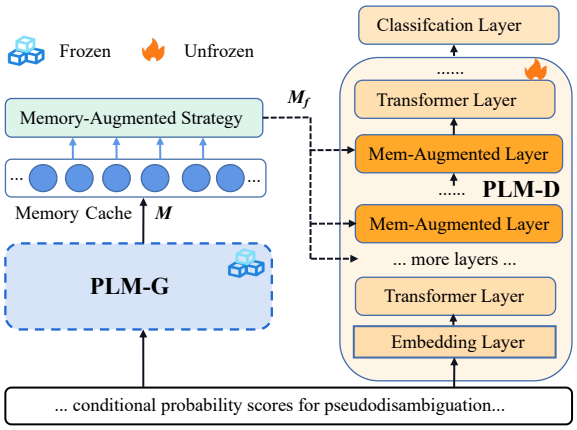

Figure 2: A framework of MAP with the cs-domain task input. PLM-G denotes the frozen general PLM, PLM-D denotes the domain-specific PLM.

2 The Method of MAP

In this section, we first overview the MAP framework. Then we detail the a new memory-augmented layer which fuses general knowledge into domain-specific PLMs. Finally, we propose different memory-augmented strategies including single-layer memory transfer, multiple-layer memory transfer, gated memory transfer and chunk-based gated memory transfer.

2.1 Overview

Our MAP framework aims to tackle the *catastrophic forgetting* of domain-specific PLMs by using the memory cache built from the frozen general PLMs, which is illustrated in Figure 2. Given an sequence $\boldsymbol{x} = [x_1, x_2, \ldots, x_t, \ldots, x_n]$ with $\boldsymbol{x_t}$ denoting the t-th token, general PLMs output the contextual representations of the input tokens as the memory cache, which is fed into the domain-specific PLMs to build final representation for domain tasks:

$$\boldsymbol{M} = \text{PLM-G}\left(\boldsymbol{x}; \theta_g\right) \quad (1)$$

$$\boldsymbol{H} = \text{PLM-D}\left(\boldsymbol{x}, \boldsymbol{M}; \theta_d\right) \quad (2)$$

where θ_g and θ_d are the parameters of general and domain-specific PLMs respectively. We only update the θ_d and the θ_g is frozen when funetuning. The general PLM could be a BERT or RoBERTa, which contains l layers of Transformer (Vaswani et al., 2017) encoder blocks and outputs a set of hidden states denoted as a memory cache $\boldsymbol{M} = \left\{\boldsymbol{M^1}, \boldsymbol{M^2}, \ldots, \boldsymbol{M^l}\right\}$. In the MAP

framework, the domain-specific PLM utilizes a new *memory-augmented layer* to adaptively incorporate the memory representation built from the memory cache $\boldsymbol{M}$ and enhance its generalization ability. Note that *memory-augmented layer* is built by only the parameters of original Transformer layer without adding new ones. Moreover, we explore different memory-augmented strategies and further propose the chunk-based memory transfer strategy, which fully uses the memories from different chunks of the general PLM.

2.2 Memory-Augmented Layer

Memory-augmented layer differs from traditional Transformer layer only in the multi-head self attention module. The new memory-augmented attention module is proposed to fuse the memory representation into the domain-specific PLM, denoted as memory-attention. Its main idea is to linearly transform the memory representation into new pairs of (keys, values) and concatenate them into the back of pairs produced by the domain-specific PLM. Then the multi-head self-attention is performed to adaptively fuse these new concatenated representation. The whole process reuses the parameters of the Transformer layer of domain-specific PLM and does not introduce any new parameters.

Specifically, if i-th Transformer layer is a memory-augmented one, it obtains the domain-specific representation $\boldsymbol{H}_{i-1}$ from previous layer and the memory representation $\boldsymbol{M}_f$ as the input and fuse them by the following way:

$$\begin{aligned}\text{Memory-Attention}(\boldsymbol{H}_{i-1}, \boldsymbol{M_f}) = \\ \text{Concat}\left(head_1, \ldots, head_k\right)\boldsymbol{W^o}\end{aligned} \tag{3}$$

where $\boldsymbol{M_f}$ is the memory representation directly extracted from the memory cache $\boldsymbol{M}$ or effectively constructed by some adaptive aggregation strategies, which has the same shape as the intermediate hidden state $\boldsymbol{H_i}$ of the domain-specific PLM, k means the number of heads and $\boldsymbol{W^o}$ is a trainable parameter matrix. Then, $\boldsymbol{M_f}$ is linearly transformed into new pairs of (keys, values) which were appended to the last of domain-specific ones:

$$\tilde{\boldsymbol{K}}_{i,j} = \text{Concat}(\boldsymbol{K}_{i,j}, \boldsymbol{M}_f^{\boldsymbol{k}}) \tag{4}$$

$$\tilde{\boldsymbol{V}}_{i,j} = \text{Concat}(\boldsymbol{V}_{i,j}, \boldsymbol{M}_f^{\boldsymbol{v}}) \tag{5}$$

$$\begin{aligned}\boldsymbol{Q}_{i,j}, \boldsymbol{K}_{i,j}, \boldsymbol{V}_{i,j} = \\ \boldsymbol{H}_{i-1}\boldsymbol{W}_{i,j}^{\boldsymbol{q}}, \boldsymbol{H}_{i-1}\boldsymbol{W}_{i,j}^{\boldsymbol{k}}, \boldsymbol{H}_{i-1}\boldsymbol{W}_{i,j}^{\boldsymbol{v}}\end{aligned} \tag{6}$$

$$\boldsymbol{M}_f^{\boldsymbol{k}}, \boldsymbol{M}_f^{\boldsymbol{v}} = \boldsymbol{M_f}\boldsymbol{W}_{i,j}^{\boldsymbol{k}}, \boldsymbol{M_f}\boldsymbol{W}_{i,j}^{\boldsymbol{v}} \tag{7}$$

where $\boldsymbol{W}_{i,j}^{\boldsymbol{q}}$, $\boldsymbol{W}_{i,j}^{\boldsymbol{k}}$ and $\boldsymbol{W}_{i,j}^{\boldsymbol{v}}$ are trainable parameters to generate queries, keys, values respectively, and j refers to j-th attention head. Then the self-attention is performed on the queries and merged pairs of (keys, values) as follow:

$$head_j = \text{Softmax}\left(\frac{\boldsymbol{Q}_{i,j}\tilde{\boldsymbol{K}}_{i,j}^{\boldsymbol{T}}}{\sqrt{d_k}}\right)\tilde{\boldsymbol{V}}_{\boldsymbol{i,j}} \tag{8}$$

where d_k is the head dimension acting as a scaling factor. Firstly, a unified attention matrix is computed by the standard scaled dot-product of each query against the keys of general memory and the domain-specific keys. Then, a softmax operation gets the normalized scores that weigh and sum these concatenated values. Without additional parameter update for the general PLM, domain-specific PLM can dynamically capture useful general knowledge and ignore noisy information through the memory-augmented layer.

2.3 Memory-Augmented Strategies

The remaining problem is how to build the memory representation $\boldsymbol{M_f}$ from the memory cache $\boldsymbol{M}$ and which layer of the domain-specific PLM should be the memory-augmented layer to fuse $\boldsymbol{M_f}$. Essentially, it is a many-to-many layer assignment problem between the general PLM and the domain-specific PLM. To study the effect of layer assignment, we propose and compare different strategies, as shown in Figure 3.

Single-Layer Memory Transfer We first consider a single-layer memory transfer approach, where the last hidden state of the memory cache $\boldsymbol{M}$ is extracted as $\boldsymbol{M_f}$ and then it is fused into one layer of domain-specific PLM with memory-attention. We choose the layer near the top of the domain-specific PLM model as the memory-augmented layer which performs best in the experiment. This strategy does not require additional parameters.

Multiple-Layer Memory Transfer The single-layer memory transfer may ignore the knowledge learned from shallow layers of the general PLM. To perform layer-wise interaction between the general PLM and the domain-specific PLM, we propose a multiple-layer transfer strategy. This strategy leverages all hidden states from the memory cache $\boldsymbol{M}$ as the memory representations and then fuses them into the corresponding layers of the domain-specific PLM, which also does not introduce any new parameters.

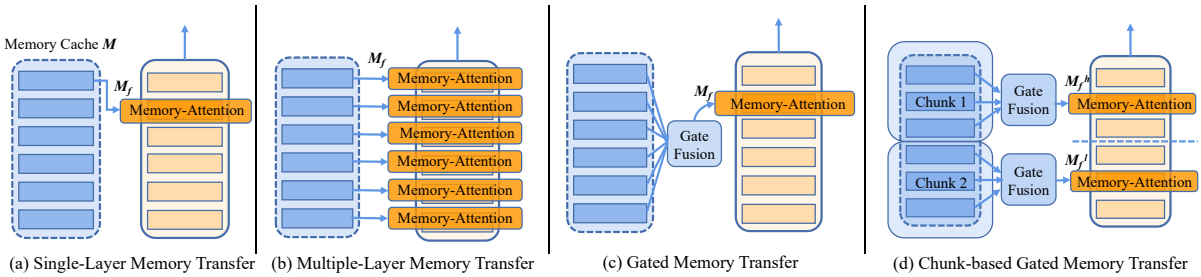

Figure 3: Memory-augmented strategies of the MAP framework. We take a 6-layer model as an example.

Gated Memory Transfer Multiple-layer memory transfer uses the hidden states output by all layers of the frozen general PLM as the memory representations, which inevitably introduces homogeneous and noise information. To avoid the problem, we further propose the strategy of gated memory transfer, which firstly exploits the token-level gate mechanism to adaptively weigh and sum representations of different layers into a memory representation, and then it will be fused into one layer of domain-specific PLM. We also choose the layer near the top of the domain-specific PLM as the memory-augmented one which achieves optimal performance in the experiment. The gate fusion mechanism is formulated as below:

$$\boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{t}} = \sum_{l=1}^{L} \alpha_l^t \boldsymbol{m}_{\boldsymbol{l}}^{\boldsymbol{t}} \tag{9}$$

$$\alpha_l^t = \frac{\exp\left(g\left(\boldsymbol{m}_{\boldsymbol{l}}^{\boldsymbol{t}}\right)\right)}{\sum_{i=1}^{L} \exp\left(g\left(\boldsymbol{m}_{\boldsymbol{i}}^{\boldsymbol{t}}\right)\right)} \tag{10}$$

$$\boldsymbol{M}_{\boldsymbol{f}} = \{\boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{1}}, \boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{2}}, \ldots, \boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{t}}, \ldots, \boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{n}}\} \tag{11}$$

where $\boldsymbol{m}_{\boldsymbol{l}}^{\boldsymbol{t}}$ is the token representation, t denotes the token index, l is the layer index, n is the length of tokens and g refers to a linear layer. We utilize a softmax function to calculate the importance of tokens in different layers. Therefore, the output token representation $\boldsymbol{m}_{\boldsymbol{f}}^{\boldsymbol{t}}$ is obtained by weighing the t-th token representations from different layers with their corresponding importance α_l^t. Finally, the built memory representation $\boldsymbol{M}_{\boldsymbol{f}}$ is fused to the memory-augmented layer.

Chunk-based Gated Memory Transfer The previous work (Liu et al., 2019a; Phang et al., 2021) have observed that the representations from upper layers and lower layers of pre-trained language models are significantly different. Motivated by this observation, based on the gated memory transfer, we further propose a chunk-based variant, which separates the layers of general PLM into a high-level chunk and a low-level chunk, and then apply the gate fusion strategy to get upper and lower-layer memory representations $\boldsymbol{M}_{\boldsymbol{f}}^{\boldsymbol{h}}$ and $\boldsymbol{M}_{\boldsymbol{f}}^{\boldsymbol{l}}$, respectively. Finally, we fuse them into two memory-agumented layers in the domain-specific PLM, and the details of different layer selections for this strategy is presented in Section 4.3.

3 Experiments

In this section, we first introduce the evaluation tasks and metrics. Then we illustrate the baseline methods, implementation settings. Finally, we conduct the experimental analysis of MAP.

3.1 Datasets and Metrics

Datasets We evaluate our model on three tasks: text classification, QA and NER. For text classification, we conduct experiments on eight tasks that cover four domains, including CHEMPROT (Kringelum et al., 2016) and RCT (Dernoncourt and Lee, 2017) in the biomedical domain, ACL-ARC (Jurgens et al., 2018) and SCIERC (Luan et al., 2018) in the computer science domain, HYPERPARTISAN (Kiesel et al., 2019) and AGNEWS (Zhang et al., 2015) in the news domain, HELPFULNESS (McAuley et al., 2015) and IMDB (Maas et al., 2011) in the reviews domain. In addition, we use micro-F1 as the metric for ChemProt and RCT, and use macro-F1 for the other datasets following (Gururangan et al., 2020). For NER, we use two datasets including NCBI-Disease (Dogan et al., 2014) in the biomedical domain, CoNNL-2003 (Sang and Meulder, 2003) in the news domain. We use the F1 score as the evaluation metric. For QA, we utilize two datasets including Medication (Pampari et al., 2018) in the biomedical domain, NewsQA (Trischler et al., 2017) in the news domain. We use the Exact-Match (EM) and the F1 score as the evaluation metrics. The detailed description and statistics of each tasks are shown in Appendix A.

Domain	BIOMED		CS		NEWS		REVIEWS	
Dataset	CP	RCT	CI	SE	HP	AG	AM	IMDB
Fine-Tuning	$81.9_{0.1}$	$87.2_{0.1}$	$63.0_{5.8}$	$77.3_{1.9}$	$86.6_{0.9}$	$93.9_{0.2}$	$65.1_{3.4}$	$95.0_{0.2}$
DAPT	$84.2_{0.2}$	$87.6_{0.1}$	$75.4_{2.5}$	$80.8_{1.5}$	$88.2_{5.9}$	$93.9_{0.2}$	$66.5_{1.4}$	$95.4_{0.1}$
Logits Fusion	$84.4_{0.3}$	$87.6_{0.1}$	$77.5_{2.5}$	$82.9_{1.3}$	$91.8_{2.5}$	$93.8_{0.1}$	$67.6_{0.2}$	$95.1_{0.2}$
Ensemble LMs	$84.6_{0.2}$	$87.4_{0.2}$	$76.2_{2.2}$	$82.6_{1.0}$	$91.4_{2.8}$	$93.9_{0.2}$	$67.5_{1.1}$	$95.1_{0.2}$
MAP (*Single-Layer Memory Transfer*)	$84.7_{0.3}$	$87.8_{0.1}$	$78.6_{2.3}$	$83.2_{0.6}$	$93.3_{1.6}$	$94.0_{0.1}$	$68.5_{0.6}$	$95.4_{0.1}$
MAP (*Multiple-Layer Memory Transfer*)	$84.7_{0.4}$	$87.7_{0.1}$	$74.9_{1.3}$	$82.0_{1.1}$	$92.2_{2.3}$	$93.8_{0.2}$	$67.6_{0.6}$	$95.2_{0.2}$
MAP (*Gated Memory Transfer*)	$84.8_{0.2}$	$87.8_{0.1}$	$77.9_{1.5}$	$83.2_{1.3}$	$93.2_{2.0}$	$94.0_{0.2}$	$68.1_{0.5}$	$95.4_{0.2}$
MAP (*Chunk-based Gated Memory Transfer*)	$\mathbf{85.0}_{0.3}$	$\mathbf{87.9}_{0.2}$	$\mathbf{79.4}_{1.0}$	$\mathbf{83.8}_{0.8}$	$\mathbf{95.2}_{0.0}$	$\mathbf{94.1}_{0.2}$	$\mathbf{69.0}_{0.8}$	$\mathbf{95.6}_{0.2}$

Table 1: The comparison against baselines on text classification tasks and performance of the proposed memory-augmented strategies. We report the averages across five random seeds, with standard deviations as subscripts. The best performance for each benchmark is marked in black bold. CP, CI, SE, HP, AG and AM denote CHEMPROT, ACL-ARC, SCIERC, HYPERPARTISAN, AGNEWS and AMAZON, respectively.

Domain	BIOMED		CS		NEWS		REVIEWS	
Dataset	CP	RCT	CI	SE	HP	AG	AM	IMDB
TAPT	$82.6_{0.4}$	$87.7_{0.1}$	$67.4_{1.8}$	$79.3_{1.5}$	$90.4_{5.2}$	$94.5_{0.1}$	$68.5_{1.9}$	$95.5_{0.1}$
DAPT+TAPT	$84.4_{0.4}$	$87.8_{0.1}$	$75.6_{3.8}$	$81.3_{1.7}$	$90.0_{6.6}$	$94.6_{0.2}$	$68.7_{1.8}$	$95.6_{0.1}$
MAP*	$\mathbf{85.1}_{0.3}$	$\mathbf{87.9}_{0.0}$	$\mathbf{79.6}_{2.2}$	$\mathbf{83.9}_{1.1}$	$\mathbf{95.2}_{0.0}$	$\mathbf{94.6}_{0.1}$	$\mathbf{69.9}_{0.3}$	$\mathbf{95.8}_{0.1}$

Table 2: The experimental results of MAP compared with TAPT and DAPT+TAPT on domain classification tasks. * means MAP framework built with the backbone PLM pre-trained with the process of DAPT then TAPT.

3.2 Baselines

In our experiment, all the baselines are built on the RoBERTa-base. The details baselines of text classification tasks are described as follows:

- **Fine-Tuning**: directly fine-tuning the general PLM for downstream tasks.
- **DAPT** (Gururangan et al., 2020): pre-training the general PLM with large-scale domain unlabeled corpora to get the domain-specific PLM then fine-tuning it.
- **Logits Fusion**: a straight-forward method combines the frozen general PLM and the domain-specific PLM by adding their logits. This method is optimized end-to-end and does not include any memory-augmented strategies in the model.
- **Ensemble LMs**: an ensemble method that adds the predicted probabilities of the fine-tuned general and the domain-specific PLMs for final prediction.
- **TAPT** (Gururangan et al., 2020): task-adaptive pretraining continues to pre-train the PLM on the training dataset, and then we fine-tune it for the downstream tasks.

For the NER and QA tasks, in addition to these above baselines, we also introduce KALA (Kang et al., 2022). Since KALA is only verified in QA and NER, we use it as a baseline for these two tasks.

- **KALA**: constructing an entity memory and knowledge graph on task-specific domain and then augmenting PLM by these additional knowledge.

3.3 Implementation

We implement the MAP framework based on RoBERTa-base. For the domain-specific PLM, we use the released pre-trained weights DAPT[1]. The more details on fine-tuning of the downstream tasks are shown in Appendix B.

3.4 Results and Analysis

Our experiment results on the domain-specific classification tasks are shown in Table 1 and 2, the results of QA and NER tasks are shown on Table 3.

Performance on Domain Classification Tasks From Table 1, we can observe that MAP with the proposed chuck-based gated memory transfer can achieve better results than all the baselines, which proves that incorporating memory from the general frozen PLM is beneficial for the domain-specific PLM. Specifically, the strategy of chunk-based gate

[1]https://github.com/allenai/dont-stop-pretraining

Domain	BIOMED		NEWS	
Dataset	Medication (QA)	NCBI-Disease (NER)	NewsQA (QA)	CoNNL-2003 (NER)
Fine-Tuning	$26.9_{0.2}$ | $71.5_{0.5}$	$86.9_{1.0}$	$57.2_{0.6}$ | $71.9_{0.4}$	$95.6_{0.2}$
TAPT	$27.0_{0.2}$ | $71.2_{0.3}$	$86.2_{0.8}$	$57.2_{0.5}$ | $71.8_{0.3}$	$95.6_{0.3}$
DAPT	$27.2_{0.3}$ | $71.4_{0.4}$	$87.2_{0.4}$	$58.7_{0.6}$ | $72.4_{0.4}$	$95.8_{0.2}$
KALA	$27.3_{0.4}$ | $71.1_{0.5}$	$87.7_{0.3}$	$58.0_{0.6}$ | $72.7_{0.3}$	$95.4_{0.3}$
MAP	$\mathbf{29.1}_{0.3}$ | $\mathbf{72.2}_{0.4}$	$\mathbf{88.7}_{0.2}$	$\mathbf{59.9}_{0.8}$ | $\mathbf{73.3}_{0.4}$	$\mathbf{96.2}_{0.2}$

Table 3: The experimental results of extractive QA and NER tasks in biomedical and news domains. We use Exact Match and F1 score as the metrics for the QA tasks: Medication and NewsQA, and F1 score for NER tasks: NCBI-Disease and CoNNL-2003.

memory transfer outperforms other strategies, we conjecture that it adaptively selects the token-level information across different layers and adequately utilizes the general knowledge from both the high-level and low-level chunks. However, we also observe that multiple-layer memory transfer has little improvement compared with the baselines, because it incorporates excessive redundant and noisy information from the general PLM without the proposed gated fusion. Besides, single-layer memory transfer is a simple yet effective strategy achieving better than the baselines and the non-gated fusion strategy of multiple-layer memory transfer.

Since the chunk-based gate memory transfer strategy achieves the best performance compared with the baselines, we use it as the default memory-augmented strategy within MAP in the following experiments.

Comparison with Further TAPT Further task-adaptive pre-training (TAPT) has been proven to improve the domain-adaptive pre-training (DAPT) (Gururangan et al., 2020). To demonstrate the effectiveness of MAP on TAPT, we build a MAP framework that replaces the domain-specific pre-trained PLM with the task-adaptive pre-trained PLM. From Table 2, we find that our MAP also outperforms DAPT+TAPT on all datasets, indicating that the proposed framework is general for different backbone models, including the domain-adaptive and the task-adaptive PLMs.

Effectiveness for QA and NER We also evaluate MAP on the tasks of QA and NER, and the experiment results are shown in Table 3. From the results, we see that our method achieves better results than the baselines on all datasets, especially KALA (Kang et al., 2022), which spends a considerable effort to construct entity memory and knowledge graph from the contexts. These results further demonstrate the effectiveness of MAP.

Domain	BIOMED		CS	
Dataset	CP	RCT	CI	SE
DAPT	$84.2_{0.2}$	$87.6_{0.1}$	$75.4_{2.5}$	$80.8_{1.5}$
MAP w/o Frozen	$84.3_{0.5}$	$87.3_{0.3}$	$78.1_{2.2}$	$82.4_{1.1}$
MAP	$\mathbf{85.0}_{0.3}$	$\mathbf{87.9}_{0.2}$	$\mathbf{79.4}_{1.0}$	$\mathbf{83.8}_{0.8}$

Table 4: Results of utilizing the frozen and unfrozen general PLMs.

4 Further Discussion

In the following sections, we conduct some detailed analysis of MAP to demonstrate the effectiveness of the frozen general PLM and memory-attention. Moreover, we apply the proposed framework in the pre-training stage and study the effect of layer selection on the performance.

4.1 Effectiveness of Frozen Memory

We compare the frozen and unfrozen ways when building the general memory representation, and the results are shown in Table 4. From the table, we observe that the frozen method is better than the unfrozen model on all datasets, and both are better than the baseline DAPT. We argue that using the frozen memory has two advantages: (1) more efficient in model training without updating the parameters of the general PLM; (2) keeps general knowledge from PLM unchanged when fine-tuning, so it does not lead to a forgetting problem.

4.2 Effectiveness of Memory-Attention

To study the effectiveness of our proposed memory-attention module, we compare it with other attention-based variants within a memory-augmented layer. Specifically, cross-attention is an attention module widely-used in the multi-modal learning (Li et al., 2021; Zeng et al., 2021), we

Domain	BIOMED		CS	
Dataset	CP	RCT	CI	SE
Cross-Attention	$82.8_{0.4}$	$86.6_{0.1}$	$73.5_{2.0}$	$82.2_{1.0}$
Gate-Attention	$84.7_{0.3}$	$87.6_{0.1}$	$78.1_{1.4}$	$83.4_{0.7}$
Memory-Attention	$\mathbf{85.0}_{0.3}$	$\mathbf{87.9}_{0.2}$	$\mathbf{79.4}_{1.0}$	$\mathbf{83.8}_{0.8}$

Table 5: Performance of using the proposed memory-attention and other main-stream attention-based variants in MAP framework. We evaluate them on the datasets of biomedical and cs domains.

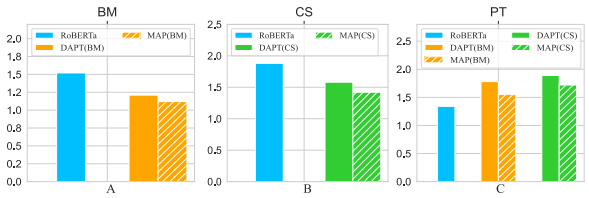

Figure 5: Maksed LM loss for the pre-training stage (lower value is better). PT denotes the samples similar to RoBERTa's pre-training corpus. DAPT(BM) denotes the domain-specific PLM for biomedical domain. MAP(BM) denotes the MAP framework with the biomedical-domain backbone. For instance, figure A represents further pre-training the models on the biomedical pre-training samples then inferring their MLM loss on the test samples.

apply it to adaptively fuse the memory representation M_f and the output representation from the self-attention module. We also include the gate-attention (Wu et al., 2022) as the fusion baseline, which utilizes a gate mechanism to weigh and sum the local and external-memory for long-sequence modeling. As shown in Table 5, our memory-attention module outperforms other variants without additional trainable parameters.

4.3 Layer Selection for Memory-Attention

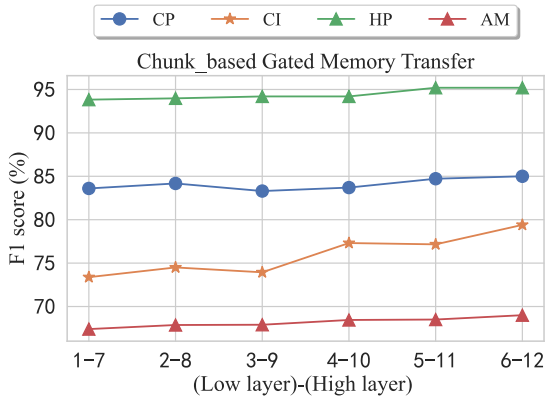

Figure 4: Performance of different layer selections in chunk-based gate memory transfer strategy.

Besides the strategy of multiple-layer memory transfer, other strategies need to do the layer selection. For the strategies of single and gated-memory transfer, we fuse the memory representation M_f into different layers $\{3, 6, 9, 12\}$ in the domain-specific PLM and find that the layer 9 as the memory-augmented layer can achieve the best performance in both strategies. We present more detailed results in Appendix C. For chunk-based gate memory transfer strategy, we experiment with transferring the memory representation of the high-level chunk into layer 7 to 12, and the other one of the low-level chunk into layer 1 to 6 , which keeps a same layer interval between the two memory-augmented layers in the domain-specific PLM. The experimental results are shown in Figure 4. The results show that there is an increasing tendency when placing memory-augmented layers to the top of the domain-specific PLM. Finally, we choose the layer 6 and 12 as the memory-augmented ones for the strategy of chunk-based gated memory transfer.

4.4 Apply MAP in the Pre-training Stage

In the previous experiment, we have incorporated the domain-specific PLM with the MAP framework in fine-tuning stage. In this section, we further study whether MAP is beneficial for pre-training stage. To this end, we randomly samples 50k documents from general[2], biomedical and computer science domains (Lo et al., 2020), respectively. In addition, we randomly split 70% of the data from each domain as the pre-training samples and the rest data as the test samples. More details about pre-processing the domain samples are shown in Appendix B. Then, we pre-train the models on the pre-training samples and calculate the masked LM loss on the test samples. From the experiment results shown in Figure 5, compared with baseline DAPT, we observe that utilizing MAP can reduce masked LM loss on the biomedical, cs and general domain. These results demonstrate that the proposed MAP also mitigates catastrophic forgetting during the adaptive pre-training.

5 Related work

Domain Adaptation for PLMs Recently, the domain shift problem of PLMs has attracted increasing research (Beltagy et al., 2019; Huang et al., 2019; Lee et al., 2020b; Gururangan et al., 2020) since the domain discrepancies between

[2]https://github.com/soskek/bookcorpus

the pre-training corpora and the downstream tasks can lead to a significant performance drop. To bridge the domain gap, SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2020b) further pre-train BERT with 1.14M scientific papers from Semantic Scholar corpus and biomedical documents, respectively, which can improve the performance of domain-specific NLU tasks compared with general BERT. Also, Gururangan et al. (2020) proposed doman-adaptive pre-training (DAPT) and task-adaptive pre-training (TAPT). Concretely, DAPT continues to pre-train the PLM with domain-specific corpora, while TAPT directly pre-trains the PLM on the task dataset. Moreover, BT-TAPT (Lee et al., 2021) inherits the crucial step of TAPT and leverages the back-translated strategy to augment the task data to improve the performance of PLM. TAPTER (Nishida et al., 2021) equips TAPT with domain-specific word embedding regularization to improve fine-tuning performance. However, above approaches suffer from catastrophic forgetting of general domain knowledge after adaptive pre-training, which leads to sub-optimal performance on downstream tasks.

Catastrophic Forgetting *Catastrophic forgetting* is a common phenomenon for continual learning, and it occurs when a training model forgets previously learned knowledge and over-fits to new tasks (Mccloskey and Cohen, 1989). Typically, regularization-based methods (Goodfellow et al., 2014; Kirkpatrick et al., 2016; Serrà et al.) exploit regularization to constrain the parameter update to alleviate the forgetting problem, and the memory-based methods (Guo et al., 2020; Saha et al., 2021) mitigate forgetting by storing important samples from past tasks in the external memory and rehearsing them via some gradient transformation strategies. In addition, plenty of works have been proposed to address *catastrophic forgetting* for NLP tasks. Dakwale and Monz. (2017) subtly minimized KL-divergence of prediction losses as a regularization term between fine-tuning and general domain models. Lee et al. (2020a) introduced a new regularization technique to mix the PLM parameters with vanilla parameters instead of stochastical dropout. Chen et al. (2020) adopted multi-task learning to jointly learn pre-training and downstream tasks with less forgetting during fine-tuning. Xie et al. (2021) preserved the model neurons of general and language-specific parts during fine-tuning. However, our method is orthogonal to the above approaches since we aim to effectively incorporate the domain-specific PLM with the memory representation built from the frozen general PLM to solve the forgetting issue without adding additional regularization terms in the model or using external memory to preserve samples from the past tasks.

Knowledge-Enhanced PLMs Knowledge-enhanced methods have shown effectiveness for PLMs via introducing internal or external knowledge. To improve the performance of fine-tuning tasks, REINA (Wang et al., 2022) retrieves the labeled training instances most similar to the input data and concatenates them before feeding them into PLMs. Besides, RETRO (Borgeaud et al., 2021) enhances the auto-regressive language model via leveraging a pre-trained frozen BERT model to retrieve related texts and then use a chunked cross-attention module to incorporate them. Memorizing transformer (Wu et al., 2022) leverages a learned gate to combine the attention results of the local context and the external context retrieved from previously seen sub-sequences. KALA (Kang et al., 2022) is the approach most relevant to our work. It incorporates intermediate hidden representations with domain-specific entities and their relational facts during task-specific fine-tune for domain tasks. However, our method doesn't need to retrieve similar texts or construct additional knowledge graphs. We propose several memory-augmented strategies to build the memory representation and then transfer it into the domain-specific PLM to mitigate the forgetting of general knowledge.

6 Conclusion

In this work, we propose MAP, a novel framework that utilizes the memory-augmented layer to fuse the memory representation built from the frozen general PLM to mitigate *catastrophic forgetting* of general knowledge caused by domain-adaptive pre-training. We explore different memory-augmented strategies to construct the memory representation and empirically find that chunk-based gate memory transfer achieves the most optimal performance. We validate MAP on various domains of classification, QA, and NER tasks. The results show that our method consistently outperforms existing baselines on all datasets, implying that explicitly leveraging forgotten general knowledge is beneficial for domain-specific downstream tasks.

7 Limitations

Our MAP framework has been validated on domain-specific tasks and a small-scale domain pre-training experiment in Section 4.4. Due to the lack of large GPU resource, we have not validated our MAP framework in large-scale pre-training, a more challenging setting that we leave as future work. We also consider automatic layer selection to be an under-studied problem and believe that AutoML techniques (Pham et al., 2018; Tan and Le, 2019), such as evolutionary search (Deb et al., 2002; Chen et al., 2021), will be promising methods. Finally, the proposed framework is built on the encoder-only model, RoBERTa-base. In the future, we will apply our framework on the other types of architectures, such as decoder-only GPT (Radford et al., 2018) and encoder-decoder BART (Lewis et al., 2020).

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2021. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426.

Huangke Chen, Ran Cheng, Witold Pedrycz, and Yaochu Jin. 2021. Solving many-objective optimization problems via multistage evolutionary search. *IEEE Trans. Syst. Man Cybern. Syst.*

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *EMNLP*.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. In *Proceedings of the XVI Machine Translation Summit*.

Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *IJCNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *ICLR*.

Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. 2020. Improved schemes for episodic memory-based lifelong learning. In *NIPS*.

Suchin Gururangan, Ana Marasovic, 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 *ACL*.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342.

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*.

Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. KALA: knowledge-augmented language model adaptation. In *NAACL*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, D. Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In **SEMEVAL*.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Jens Kringelum, Sonny Kim Kjærulff, Søren Brunak, Ole Lund, Tudor I. Oprea, and Olivier Taboureau. 2016. Chemprot-3.0: a global chemical biology diseases mapping. *Database J. Biol. Databases Curation*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020a. Mixout: Effective regularization to fine-tune large-scale pretrained language models. In *ICLR*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020b. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*

Junghoon Lee, Jounghee Kim, and Pilsung Kang. 2021. Back-translated task adaptive pretraining: Improving accuracy and robustness on text classification. *CoRR*, abs/2107.10474.

Yoav Levine, Itay Dalmedigos, Ori Ram, Yoel Zeldes, Daniel Jannai, Dor Muhlgay, Yoni Osin, Opher Lieber, Barak Lenz, Shai Shalev-Shwartz, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Standing on the shoulders of giant frozen language models. volume abs/2204.10019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *NIPS*.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *ECCV*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*, pages 1073–1094.

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. *CoRR*, abs/1907.11692.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *ACL*.

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*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, A. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. *SIGIR*.

Michael Mccloskey and Neil J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. In *BioNLP@ACL*.

Kosuke Nishida, Kyosuke Nishida, and Sen Yoshida. 2021. Task-adaptive pre-training of language models with word embedding regularization. In *Findings of ACL*.

Anusri Pampari, Preethi Raghavan, Jennifer J. Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *EMNLP*.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. 2018. Efficient neural architecture search via parameter sharing. In *ICML*, Proceedings of Machine Learning Research.

Jason Phang, Haokun Liu, and Samuel R. Bowman. 2021. Fine-tuned transformers show clusters of similar representations across layers. In *BlackboxNLP@EMNLP*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Gobinda Saha, Isha Garg, and Kaushik Roy. 2021. Gradient projection memory for continual learning. In *ICLR*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*.

Mingxing Tan and Quoc V. Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, Proceedings of Machine Learning Research.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *NAACL-HLT*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *ACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara

Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *CoRR*, abs/2203.08913.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In *ACL*.

Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-grained vision language pre-training: Aligning texts with visual concepts. In *ICML*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *ArXiv*, abs/1509.01626.

A Dataset Descriptions and Statistics

This section describes the details and statistics of three tasks: domain classification, domain extractive question answering (QA), and named entity recognition (NER).

For the text classification tasks, we leverage the following datasets covering four domains, including biomedical science, computer science, news, and reviews. In the biomedical domain, CHEMPROT (Kringelum et al., 2016) is the relation classification dataset based on chemical-protein interaction. RCT (Dernoncourt and Lee, 2017) is the role classification task constructed from the abstract of the biomedical articles. In the computer science domain, ACL-ARC (Jurgens et al., 2018) is the task of annotated citations for articles' functions. SCIERC (Luan et al., 2018) is constructed from scientific abstracts annotated with relation. In the news domain, HYPERPARTISAN (Kiesel et al., 2019) is the news text classification for determining partisan leanings. AGNEWS (Zhang et al., 2015) is the topic classification for news. In the reviews domain, AMAZON (McAuley et al., 2015) is a binary classification task consisting of feedback on products. IMDB (Maas et al., 2011) consists of movies reviews, which is a binary sentiment classification dataset.

For the NER tasks, we use two datasets involving the news and biomedical domain. Concretely, CoNLL-2003 (Sang and Meulder, 2003) consists of news stories from the Reuters Corpus. NCBI-Disease (Dogan et al., 2014) is annotated with disease mentions.

For the QA tasks, we utilize two domain-specific datasets. Specifically, NewsQA (Trischler et al., 2017) is a machine comprehension dataset consisting of news articles. Medication (Pampari et al., 2018) is constructed by electronic medical records from clinical text.

The detailed statistics of text classification tasks and NER tasks are shown in Table 7, QA in Table 8.

B Implementation Details

We use the huggingface (Wolf et al., 2020) library to implement our MAP framework, which contains various transformer-based pre-trained language models (PLMs) and their saved checkpoints. We implement the DAPT, TAPT and DAPT+TAPT models of biomedical, cs, news and reviews do-

mains [3] from the library released by (Gururangan et al., 2020). All the experiments are implemented on Nvidia V100 GPUs with 32GB memory. We select the best checkpoint on the validation set during training to infer the test set.

Configurations for Classification Tasks In this section, we explain the setting of fine-tuning for domain-specific classification tasks. We fine-tune the domain-specific PLM with our MAP framework for 5 to 15 epochs, respectively, with the same learning rate of 4e-5 and the dropout rate of 0.5. The default classification layer of the model is 1 except for the IMDB dataset with 2, and the default maximum sequence length is 256 except for IMDB with 512. We leverage the Adam optimizer to schedule the learning rate, with the Adam epsilon of 1e-8, the Adam beta-1 of 0.9, and the Adam beta-2 of 0.999. We apply the grid-search method to find the optimal batch size and numbers of GPUs for all the classification datasets. The detailed settings are shown in Table 7.

Configurations for QA and NER For the extractive QA tasks, we fine-tune the domain-specific PLM with our MAP framework for 3 epochs, which can converge to optimal performance. Besides, we train the model with the maximum sequence length of 384 and the learning rate of 3e-5, the weight decay rate with 1e-2 and the warm-up rate of 6e-2. For the experiments on NER tasks, we fine-tune NCBI-Disease for 20 epochs and CoNNL-2003 for 15 epochs, with the same maximum sequence length of 128, the learning rate of 5e-5, and the weight decay rate and warm-up rate are set to 0. Different from domain classification tasks, we utilize AdamW as learning rate optimizer instead of Adam. In addition, we also adopt the grid-search method to find the optimal bach size and number of GPUs for all tasks. The detailed settings are shown in Table 10.

Configurations for small-scale Pre-training This part describes the experimental settings of adaptive pre-training with MAP framework. Our simple pre-training experiment needs some external domain-relative corpora of two domains: Biomedical and Computer Science. Following by (Gururangan et al., 2020), we adopt SCISPACY (Neumann et al.) as a sentence splitting tool to obtain abstract and body paragraphs from S2ORC (Lo et al., 2020). After pre-processing for the corpora, we randomly sample 50K data for each domain and split 70% of them as pre-training sets and 30% as test sets. For the general corpus similar to ROBERTA's pre-training corpus, we also randomly sample 50K data from BOOKCORPUS[4] and split them by using the method mentioned above. The detailed hyperparameter settings for this cross-domain adaptive pre-training are shown in Table 6.

Hyperparameters	Domain corpus
Training epochs	5 to 10
Batch size per GPU	32
Number of GPUs	4
Maximum Sequence Length	512
The number of text lines (pre-training)	35K
The number of text lines (inference)	15K
Learning Rate	4e-5
Learning Rate Optimizer	Adam
Adam epsilon	1e-8
Adam beta 1	0.9
Adam beta 2	0.999

Table 6: Hyperparameters for simple adaptive pre-training with MAP framework on biomedical and cs corpora.

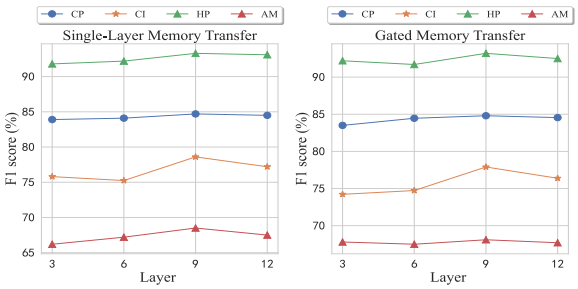

Figure 6: Performance of different layer-selection indexes of memory-attention for single-layer memory transfer and gated memory transfer strategies.

C Layer Selection Experiment

For single-layer memory transfer and gated memory transfer strategies, we experiment with adding the memory-attention to layer 3,6,9 and 12 in a 12-layer RoBERTa-base model, with result shown in Figure 6. We empirically find that adding memory-attention to the 9-th layer of the domain-specific model as the memory-augmented layer will obtain the best results for the two strategies. However, adding memory-attention to either too upper or too lower obtained fewer gains. Therefore, we adopt memory-attention on the 9-th layer as the default choice for the two strategies in the main experiment shown in Table 1.

[3] https://huggingface.co/allenai

[4] https://github.com/soskek/bookcorpu

Task	Domain	Dataset	Train	Dev.	Test	Classes
Classification	BIOMED	CHEMPROT	4169	2427	3469	13
	BIOMED	RCT†	180040	30,212	30135	5
	CS	ACL-ARC	1,688	114	139	6
	CS	SCIERC	3219	455	974	7
	NEWS	HYPERPARTISAN	515	65	65	2
	NEWS	AGNEWS†	115000	5000	7600	4
	REVIEWS	**Amazon**†	115251	5000	25000	2
	REVIEWS	**IMDB**†	20,000	5000	25000	2
Named Entity Recognition	News	CoNLL-2003	14,041	3,250	3,453	-
	Biomed	NCBI-Disease	5,433	924	941	-

Table 7: Statistics of Classification and NER tasks involving four domains including Biomedical, Computer Science, News and Reviews. † indicates high-resource settings.

		Train		Dev.		Test	
Domain	**Dataset**	**Context**	**Question**	**Context**	**Question**	**Context**	**Question**
Reviews	NewsQA	11428	74160	-	-	106	674
Biomed	Medication	182	7518	26	1858	53	4005

Table 8: Statistics of QA tasks including News and Biomedical domains. We report the number of contexts and questions of the two datasets.

	BIOMED		**CS**		**NEWS**		**REVIEWS**	
Hyperparameters	CP	RCT	CI	SE	HP	AG	AM	IMDB
Training Epochs	14	4	15	15	12	5	6	15
Batch Size per GPU	24	16	32	32	32	16	16	16
Number of GPUs	3	4	1	1	1	4	4	4
Maximum Sequence Length	256	256	256	256	256	384	512	512
Learning Rate	4e-5							
Dropout	0.5							
Classification Layer	1	1	1	1	1	2	1	2
Learning Rate Optimizer	Adam							
Adam Epsilon	1e-8							
Adam Beta 1	0.9							
Adam Beta 2	0.999							

Table 9: Hyperparameters for fine-tuning on eight classification tasks of four domains, we use these hyperparameters for reporting the performances of our proposed MAP framework in the main papers.

	BIOMED		**NEWS**	
Hyperparameters	Medication	NCBI-Disease	NewsQA	CoNNL-2003
Training epochs	3	20	3	15
Batch size per GPU	16	32	16	32
Number of GPUs	4	1	4	1
Maximum Sequence Length	384	128	384	128
Classification layer	1			
Learning rate optimizer	AdamW			
Learning rate	3e-5	5e-5	3e-5	5e-5
Weight Decay	1e-2	0	1e-2	0
LR decay Warm-up rate	6e-2	0	6e-2	0

Table 10: Hyperparameters for fine-tuning on QA and NER tasks of biomedical and news domains, we use them for reporting the performances of our proposed MAP framework in the main papers.