NLoPT: N-gram Enhanced Low-Rank Task Adaptive Pre-training for Efficient Language Model Adaption

Hao Gu[1,2],Jiangyan Yi[1,2,†], Zheng Lian[1,2], Jianhua Tao[3,4], Xinrui Yan[1,2]
[1]State Key Laboratory of Multimodal Artificial Intelligence Systems,Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Department of Automation, Tsinghua University, China
[4]Beijing National Research Center for Information Science and Technology, Tsinghua University
guhao2022@ia.ac.cn, jiangyan.yi@nlpr.ia.ac.cn
lianzheng2016@ia.ac.cn, jhtao@tsinghua.edu.cn, yanxinrui2021@ia.ac.cn

Abstract

Pre-trained Language Models (PLMs) like BERT have achieved superior performance on different downstream tasks, even when such a model is trained on a general domain. Moreover, recent studies have shown that continued pre-training on task-specific data, known as task adaptive pre-training (TAPT), can further improve downstream task performance. However, conventional TAPT adjusts all the parameters of the PLMs, which distorts the learned generic knowledge embedded in the original PLMs weights, and it is expensive to store a whole model copy for each downstream task. In this paper, we propose NLoPT, a two-step n-gram enhanced low-rank task adaptive pre-training method, to effectively and efficiently customize a PLM to the downstream task. Specifically, we first apply Low-Rank Adaption (LoRA), a prevalent parameter-efficient technique, for efficient TAPT. We further explicitly incorporate the task-specific multi-granularity n-gram information via the cross-attention mechanism. Experimental results on six datasets from four domains illustrate the effectiveness of NLoPT, demonstrating the superiority of LoRA based TAPT and the necessity of incorporating task-specific n-gram information.

Keywords: Task adaptive Pre-training (TAPT), Low-Rank Adaption (LoRA), N-gram

1. Introduction

Large Language Models (LLMs) (Touvron et al., 2023; Chowdhery et al., 2023) have achieved superior performance in a wide range of natural language processing tasks. When applied in the task of text classification, LLMs like ChatGPT (OpenAI, 2022), are prompted to generate results for a test sample by conditioning the model on a few in-context examplars or instructions describing the task. In spite of the success that LLMs have achieved superior performance comparable to supervised baselines or even state-of-the-art results in a variety of text classification benchmarks, these models can be costly in terms of token and time usage, especially when many LLM calls are needed. Thus, it is still prevalent to adopt *pre-training then fine-tuning* paradigm for encoder-only Pre-trained Language Model (PLMs) like BERT (Devlin et al., 2019), along with its variations such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020).

However, it is not optimal to directly fine-tune a general BERT-based pre-trained model on a domain-specific task because PLMs are often pre-trained in a general domain, which means there are domain gaps between the pre-training stage and fine-tuning stage (Zhang et al., 2021b). To mitigate the problem of domain shift, existing work SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2020) were trained from scratch with domain corpus. However, these methods heavily rely on a large-scale domain-specific corpus which is laborious and expensive to construct (Zhang et al., 2021a). Therefore, considerable efforts had been dedicated to adapting PLMs to a target task by conducting continuous pre-training, i.e., starting from a general model (e.g., BERT) and continuously pre-training with a similar objective on domain-specific corpus. Gururangan et al. (2020) proposed domain adaptive pre-training (DAPT) and task adaptive pre-training (TAPT). DAPT and TAPT primarily differ in the way they utilize data. DAPT employs a large corpus of unlabeled domain-specific text, making time and hardware accessibility still the major constraints for developing such systems. In contrast, TAPT solely leverages the training data of the downstream task for continuous pre-training. Moreover, numerous studies (Kim et al., 2021; Diao et al., 2021; Nishida et al., 2021) had demonstrated that DAPT and TAPT can improve the downstream task performance. In this work, we focus on TAPT as it is more resource-efficient and practical. Nevertheless, there are still two main challenges in TAPT that remain insufficiently addressed.

[†]Corresponding author.

LREC-COLING 2024, pages 12259–12270

Firstly, conventional TAPT adjusts all parameters of the PLMs using task-specific data, which is susceptible to catastrophic forgetting. Furthermore, full-parameter TAPT can be costly, as it requires storing a model copy for each individual downstream task. Alternatives such as adapter (Houlsby et al., 2019; Kim et al., 2021) and sparse tuning (Guo et al., 2021) were proposed to overcome the first challenge. These methods, however, remain inefficient in terms of parameters and computation, as they either introduce inference latency or necessitate complex training steps. To mitigate these limitations, motivated by the recent successful application of Low-Rank Adaptation (LoRA) (Hu et al., 2022) in instruction tuning of LLMs such as LLaMA (Touvron et al., 2023), we employ the LoRA technique for efficient TAPT in this work.

As for the second challenge, conventional full-parameter TAPT ignores that many specialized domains contain specific words not included in the vocabulary of PLMs. These domain-specific vocabularies that provide a compact representation of the target domain play a vital role in the domain adaptation of PLMs. Yao et al. (2021) introduced a framework that expands task-specific vocabulary automatically by augmenting task-specific subword units. Nishida et al. (2021) was designed to align the static word embedding of a PLM with the word embedding derived in the target domain with FastText (Bojanowski et al., 2017). Nevertheless, all these works overlook the multi-grained domain information carried by n-grams and fail to utilize it effectively. Diao et al. (2021) utilized an unlearnable n-gram matching matrix to bridge the domain gap between source and target vocabulary. In contrast, we incorporate n-gram information via the cross-attention mechanism.

In this paper, we address the two aforementioned problems during conventional TAPT by proposing **NLoPT**: N-gram enhanced Low-Rank Task adaptive Pre-training, which consists of two stages: **LoRA TAPT** and **N-gram Fusion**. Specifically, in LoRA TAPT, we apply LoRA for task adaptive pre-training to capture task-specific knowledge efficiently. Several learnable LoRA modules are inserted in a PLM, then we train the LoRA modules and Masked Language Model (MLM) Head with the MLM loss on the task corpus. Note that parameters of the underlying PLM are frozen to prevent forgetting general knowledge stored in the original parameters. In N-gram Fusion, we inject task-specific multi-granularity n-gram information. In detail, we first merge the trained LoRA modules with the general model, resulting in a task-specific PLM. Then, we utilize a Pointwise Mutual Information (PMI) based method to extract task-specific n-grams from the downstream task. Subsequently, the task-specific n-gram information is injected through a cross-attention mechanism based fusion module. Afterwards, the n-gram fusion module and the classifier are supervised fine-tuned on the target task using cross-entropy loss. Experimental results demonstrate that NLoPT can effectively customize a PLM to a domain-specific downstream task.

In summary, the main contributions of this paper are as follows:

- We introduce NLoPT, a two-step process consisting of LoRA TAPT and N-gram Fusion, to customize a PLM to a domain-specific task effectively. Specifically, We first apply the LoRA technique for efficient TAPT, then we inject task-specific n-gram information via a cross-attention based module.
- Extensive experiments on six datasets across four domains show the effectiveness of our proposed method NLoPT, demonstrating the superiority of LoRA TAPT and the necessity of incorporating n-gram information.

2. Related Work

2.1. Unsupervised Domain Adaption

Existing unsupervised domain adaption methods can be broadly classified into two categories. (1) The *model-centric* approaches involve augmenting feature space (Ben-David et al., 2020) or designing new loss function (Ganin et al., 2016). (2) The *data-centric* approaches aim to develop better data selection schemes (Han and Eisenstein, 2019). A popular *model-centric* method (Gururangan et al., 2020) is to continually train a general PLM with task-relevant unlabeled data, leading to performance improvement of downstream tasks. However, these works require updating all parameters during adaption, which may distort the generic knowledge learned by the general PLM. Our work builds on these methods and makes them more parameter-efficient.

2.2. Parameter-Efficient Technique

As the size of pre-trained models continues to increase, storing a separate copy of the model for each downstream task becomes impractical. Consequently, a recent focus has been on the emerged parameter efficient techniques.

Adapter, introduced by (Houlsby et al., 2019), is the first to present the concept of parameter-efficient tuning, which involves inserting lightweight, task-specific layers or modules to a pre-trained model and only tuning their parameters. By enhancing the input sequence with continuous trainable tokens, the prompt based parameter-efficient methods (Lester et al., 2021; Liu et al., 2021a; Li and

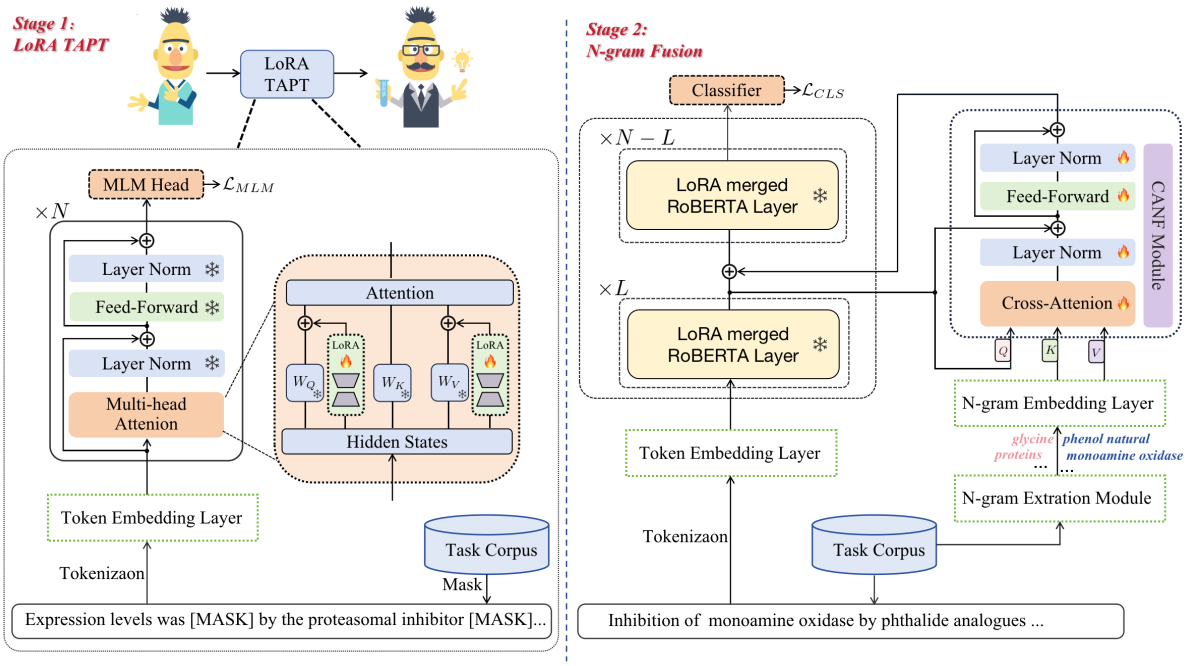


Figure 1: The overall structure of our method NLoPT. In this work, we use RoBERTa for experiment. Left: LoRA TAPT; right: N-gram Fusion.

Liang, 2021; Liu et al., 2021b), which are comparable to a continuous and differentiable extension of prompt engineering (Hambardzumyan et al., 2021), selectively update these new tokens while keeping the remaining parameters fixed. The observation (Aghajanyan et al., 2021) that PLMs inherently exhibit a low-rank property has sparked its broad application in various parameter-efficient tuning methods, such as LoRA (Hu et al., 2022), Compacter (Mahabadi et al., 2021). In addition, numerous studies have demonstrated that these parameter-efficient methods can achieve performance on par with full-parameter fine-tuning.

3. Preliminary

In this section, we give a brief description of Low-Rank Adaption (LoRA). LoRA is an effective parameter-efficient technique tailored for Transformer architectures given its extraordinary performance as reported in (Hu et al., 2022). LoRA reduces the number of trainable parameters by incorporating rank decomposition matrices into linear operation. Specifically, for a pre-trained weight W_0, LoRA composes its update ΔW_0 into two low-rank matrices W_A and W_B as shown in equation 1, where $W_A \in \mathbb{R}^{m\times r}$,$W_B \in \mathbb{R}^{r\times n}$,and the rank $r \ll min(m, n)$.

$$\Delta W_0 = \alpha W_A W_B \quad (1)$$

For any linear projection that takes a specific input x, LoRA modifies the output h as:

$$h = W_0 x + \Delta W_0 x = W_0 x + \alpha W_A W_B x \quad (2)$$

Keep in mind that during adaption, W_0 is frozen while $\theta_{lora} = \{W_A, W_B\}$ forms the trainable module. As recommended by (Hu et al., 2022), W_A is initialized as a random Gaussian initialization and W_B is initialized to all zeros at the beginning of training. α serves as a hyperparameter that modulates the effect of the adaption process. Despite LoRA's flexibility in being applied to any linear operation, we choose, for simplicity, to follow the practice established in (Hu et al., 2022), where it's only utilized in the query and value projection matrices of the attention module.

4. Method

Figure 1 shows the overall framework of NLoPT which consists of two stages, LoRA TAPT and N-gram Fusion. In LoRA TAPT, we conduct continuous pre-training for a general PLM with LoRA. Specifically, we train the learnable LoRA module and Masked Language Model (MLM) Head with MLM loss to facilitate more effective domain knowledge transferring. In N-gram Fusion, we merge the trained LoRA module with the original PLM to get the task-specific PLM. We then enhance the task-specific PLM with multi-granularity n-gram information from the task domain through the Cross-Attention based N-gram Fusion (**CANF**) module.

Finally, we fine-tune the CANF module and the classifier on the task dataset supervised while keeping the task-specific PLM frozen.

In this section, we first introduce LoRA TAPT. Subsequently, we introduce the n-gram extraction module and CANF module in detail.

4.1. LoRA TAPT

To help a general model to acquire task-specific knowledge, we explore a LoRA based TAPT approach to adapt the model to the target domain. Initially, we select a Pre-trained Masked Language Model $\mathcal{F}$. We then train the LoRA module and MLM Head with the Masked language model loss on the task corpus. We inherit from the conventional masking strategy (Devlin et al., 2019). Specifically, assume we have an input sequence X from task corpus $\mathcal{D}$, denoted as $X = (x_1, x_2, ...x_n)$. A subset of tokens $Y \in X$ are sampled for substitution, accounting for 15% of the tokens in X. Subsequently, 80% Y are replaced with a special token [MASK], 10% are replaced with random tokens, and 10% are left unchanged. The task is to recover the original tokens in Y. The objective function can be formulated as follow:

$$\mathcal{L}_{MLM} = \mathbb{E}_{X\sim\mathcal{D}}(\mathbb{E}_Y \sum_{x_i \in Y} (-\log P(x_i|X_{/Y}; \Theta)) \quad (3)$$

where $X_{/Y}$ represents the masked sequence of X with tokens specified in Y and Θ represents the trainable parameters. Note that we only make the injected LoRA module and MLM head trainable while keeping the parameters of the original PLM frozen, which makes our LoRA TAPT can capture transferable features for task domain and preserve the general knowledge embedded in the original PLM weights.

Furthermore, compared to full-parameter TAPT, LoRA TAPT does not require any modifications to the model structure or the training process. Consequently, our LoRA TAPT method can be seamlessly extended to scenarios where model architectures or objectives differ.

4.2. N-gram Fusion

The N-gram Fusion step comprises two key components: the N-gram Extraction Module, which aims to extract task-specific n-grams from a given downstream task dataset, and the CANF module, which injects task-specific n-gram information into the model.

N-gram Extraction Module N-grams basically denote a sequence of consecutive words within a given window. Here, we extract task-specific n-grams using an unsupervised method. Suppose a sentence $\mathcal{S}$ from task dataset $\mathcal{D}$ can be formulated as $\mathcal{S} = w_1 w_2 \dots w_n$. For any two adjacent words x_{pre} and x_{next}, we compute their Pointwise Mutual Information (PMI) as:

$$PMI(x_{pre}, x_{next}) = \log \frac{P(x_{pre}, x_{next})}{P(x_{pre}) \cdot P(x_{next})} \quad (4)$$

where $P(x_{pre})$ and $P(x_{next})$ denote the probability of words x_{pre} and x_{next} respectively. $P(x_{pre}, x_{next})$ stands for the probability that x_{pre} and x_{next} will co-occur. The main intuition is if a high $PMI(x_{pre}, x_{next})$ score is observed, then x_{pre} and x_{next} are more likely to occur simultaneously rather than independently, thus forming an n-gram. Consequently, we place a delimiter when the PMI of two adjacent words is less than a specified threshold. This allows us to regard those successive words that are not separated by a delimiter as potential task-specific n-grams. We segment each sentence in the training set of the target task using the aforementioned method. Subsequently, by extracting the most frequently appeared n-grams, we form a task-specific n-gram set, denoted as $\mathcal{M}$, in which each n-gram has a minimum frequency of f. In our experiment, we extract task-specific unigrams (task-specific vocabulary, glycine, for example), bigrams (phenol natural, for example) and trigrams.

CANF Module First, we merge the trained LoRA module during LoRA TAPT with the general PLM to form the task-specific PLM, denoted as $\mathcal{F}'$. Then we frozen the parameters of $\mathcal{F}'$. To make $\mathcal{F}'$ aware of the task-specific multi-granularity information, we incorporate n-gram representation at the L-th layer of $\mathcal{F}'$. We can obtain a task-specific n-gram set $\mathcal{M}$ with the n-gram extraction module, then the embedding of task-specific n-grams can be derived via an n-gram embedding layer. We design CANF module, an n-gram fusion module based on the cross-attention mechanism, denoted as $\mathcal{F}_{ngram}$, aiming to inject the task-specific n-gram representation into the model. The input Q of CANF module comes from the L-th layer hidden states of $\mathcal{F}'$ corresponding to the input sequence, denoted as H_L. Both K and V are from the output of the n-gram embedding layer, denoted as G. Afterwards, I_{L+1}, which is the input of $(L+1)$-th layer of $\mathcal{F}'$, can be formulated as:

$$I_{L+1} = H_L + \mathcal{F}_{ngram}(Q = H_L, K = V = G) \quad (5)$$

Finally, we feed the hidden state corresponding to the [CLS] token into the classifier for classification. Here, we simply operate a linear layer to obtain the probabilities. The CANF module and the classifier are fine-tuned on the labeled task dataset with the cross-entropy loss.

5. Experiment Settings

In this section, we first introduce six datasets. Following that, the comparison methods, evaluation metrics and implementation details are represented, respectively.

5.1. Datasets

Following standard practice (Gururangan et al., 2020), we conduct our experiments across six text classification tasks spanning four domains, including biomedical(BIOMED), computer science(CS), NEWS, and REVIEWS. The datasets are described as follows.

- CHEMPROT (Kringelum et al., 2016) is a dataset manually annotated from 5,031 abstracts, with the aim of identifying potential relations between chemicals and proteins.
- RCT (Dernoncourt and Lee, 2017), a dataset based on PubMed for sequential sentence classification, consists of around 200,000 abstracts, totaling 2.3 million sentences, each labeled with their role in the abstract.
- ACL-ARC (Jurgens et al., 2018), a dataset sampled from the ACL Anthology Reference Corpus, contains approximately 2,000 citations annotated for their function.
- SCIERC (Luan et al., 2018), a dataset that comprises 500 scientific abstracts, is annotated specifically for relation classification.
- HYPERPARTISAN (Kiesel et al., 2019), designed for partisanship classification, incorporates 645 articles from hyperpartisan news sources that exhibit extreme left-wing or right-wing biases.
- IMDB (Maas et al., 2011) is a binary sentiment analysis dataset, conprised of 50,000 balanced reviews gathered from the Internet.

More details about the statistics of the datasets can be found in Table 1.

5.2. All Comparison Methods

In this subsection, we will give a introduction of the comparison methods in experiments conducted in Section 6.

We compare different combinations of n-gram fusion approaches and TAPT strategies. For n-gram fusion approaches, we consider no n-gram, TDNA, our proposed CANF module. TDNA is an n-gram incorporation method proposed in (Diao et al., 2021), designed to equip the model with n-gram information via an n-gram matching matrix. For TAPT methods, we compare LoRA TAPT with both Vanilla Fine-Tuning (i.e., without TAPT), Full-parameter TAPT, and Adapter TAPT. Details of different TAPT methods are described as follows:

- **Vanilla Fine-Tuning (FT)** directly fine-tunes the PLM on downstream tasks without conducting TAPT.
- **Full-parameter TAPT (F-TAPT)** (Gururangan et al., 2020) trains the entire parameters of the PLM during TAPT.
- **Adapter TAPT (A-TAPT)** (Kim et al., 2021) inserts random initialized MLP-like module between Transformer block. During training, all the pre-trained parameters are frozen and only the newly inserted adapter layers are trainable.

5.3. Evaluation

Following existing works (Diao et al., 2021), we use macro-F1 for ACl-arc, SciErc, Hyperpartisan, IMDB, and micro-F1 for ChemProt and RCT. Macro-F1 and Micro-F1 are both commonly used evaluation metrics in multi-class classification problems. Macro-F1 separately computes the F1 score for each class, and then takes the average. In contrast, Micro-F1 assigns equal weight to each individual instance or prediction, regardless of the class. This characteristic is beneficial in class-imbalanced scenarios, which is true for ChemProt and RCT.

5.4. Implementation Details

All models are implemented in Pytorch. We leverage the pre-trained RoBERTa-base model and checkpoint from Huggingface's Transformer library (Wolf et al., 2020). We utilize FastText (Bojanowski et al., 2017) for a warm start for n-gram embeddings. We use AdamW (Loshchilov and Hutter, 2019) optimizer with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-6$. We set the learning rate following the previous work (Diao et al., 2021). When conducting TAPT, the settings are as follows: the bottleneck is set to 64 for Adapter TAPT, and the default rank r is 8 for our LoRA TAPT. The CANF module position (L in Figure 1) is set to 11 by default.

6. Experimental Results

In this section, to verify the effectiveness and efficiency of our proposed method NLoPT, we extensively experiment on six datasets across four domains. The experimental results are summarized in Table 2.

6.1. Effectiveness of LoRA TAPT

Below we delve deeper into two primary discoveries to verify the effectiveness of LoRA TAPT.

Domain	Dataset	Task Type	# of Class	# of Sentences		
				Train	Dev	Test
BioMed	ChemProt (CP)	relation classification	13	4,169	2,427	3,469
	RCT	abstract sent. roles	5	180,400	30,212	30,135
CS	ACl-arc (AC)	citation intent	6	1,688	114	139
	SciErc (SE)	relation classification	7	3,219	455	974
News	HyperPartisan (HP)	partisanship	2	515	65	65
Reviews	IMDB	review sentiment	2	20,000	50,000	25,000

Table 1: Statistics of the six task datasets in four target domains.

	N-gram			BioMed		CS		News	Reviews
	No	TDNA	CANF	CP	RCT	AC	SE	HP	IMDB
FT	✓			$81.25_{0.40}$	$87.06_{0.01}$	$73.38_{1.03}$	$82.64_{0.31}$	$88.30_{2.47}$	$94.26_{0.04}$
		✓		$82.05_{0.11}$	$87.11_{0.02}$	$73.90_{2.79}$	$83.80_{0.15}$	$88.92_{1.32}$	$94.03_{0.29}$
			✓	$82.60_{0.32}$	$87.21_{0.01}$	$74.39_{0.53}$	$84.54_{0.28}$	$89.91_{1.61}$	$94.43_{0.17}$
F-TAPT	✓			$83.17_{0.29}$	$87.49_{0.02}$	$75.25_{2.20}$	$84.66_{0.08}$	$91.08_{1.32}$	$95.28_{0.07}$
		✓		$83.29_{0.13}$	$87.47_{0.03}$	$76.29_{0.95}$	$85.14_{0.21}$	$90.15_{2.47}$	$95.18_{0.27}$
			✓	$\mathbf{83.99_{0.09}}$	$\mathbf{87.81_{0.01}}$	$\mathbf{77.14_{1.31}}$	$\mathbf{86.00_{0.13}}$	$\mathbf{92.42_{1.42}}$	$\mathbf{95.96_{0.29}}$
A-TAPT	✓			$82.76_{0.15}$	$87.43_{0.01}$	$74.12_{1.92}$	$85.06_{0.40}$	$88.96_{1.07}$	$95.02_{0.22}$
		✓		$82.84_{0.07}$	$87.27_{0.04}$	$74.24_{0.57}$	$85.11_{0.49}$	$89.23_{1.41}$	$95.08_{0.38}$
			✓	$83.52_{0.21}$	$87.55_{0.08}$	$75.34_{2.32}$	$85.21_{1.04}$	$90.22_{0.24}$	$95.42_{0.08}$
L-TAPT	✓			$82.98_{0.13}$	$87.41_{0.06}$	$74.37_{0.54}$	$84.59_{0.03}$	$90.28_{0.29}$	$95.26_{0.10}$
		✓		$83.07_{0.08}$	$87.45_{0.03}$	$75.02_{0.56}$	$85.11_{0.16}$	$89.61_{0.56}$	$95.42_{0.14}$
			✓	$83.75_{0.04}$	$87.60_{0.05}$	$75.73_{0.41}$	$85.58_{0.06}$	$91.89_{1.30}$	$95.81_{0.19}$
F-TAPT†	✓			$82.6_{0.4}$	$87.7_{0.4}$	$67.4_{1.8}$	$79.7_{1.5}$	$68.5_{1.9}$	$95.5_{0.1}$
A-TAPT†	✓			$82.7_{0.4}$	$87.4_{0.1}$	$69.3_{2.5}$	$82.4_{1.0}$	$70.8_{0.8}$	$95.6_{0.1}$

Table 2: Perfomances comparison on different combinations of TAPT methods and n-gram fusion strategies. For the sake of simplicity, we denote the Fine-Tuning, full-parameter TAPT, Adapter TAPT, LoRA TAPT as FT, F-TAPT, A-TAPT, L-TAPT respectively. We report average scores across five random seeds, with standard deviations as subscripts. F-TAPT†, A-TAPT† are taken from Gururangan et al. (2020) and Kim et al. (2021) for comparison, respectively. The highest result of each dataset is bolded and performance of our NLoPT is highlighted in grey.

	# of Params	Storage Size
F-TAPT	$115.26M$	$478MB$
A-TAPT	$3.02M$	$9.2MB$
L-TAPT	$0.94M$	$1.2MB$

Table 3: Here, we provide a comparison of various TAPT methods. # of params refers to the number of trainable parameters during TAPT and storage size indicates the disk space required for different TAPT methods.

Comparison with full-parameter TAPT We find that LoRA TAPT outperforms models without TAPT and exhibits comparable performances to the full-parameter TAPT scenario. LoRA TAPT consistently exceeds models without TAPT (i.e., fine-tuning) by a substantial margin on all classification tasks, no matter whether task-specific n-gram knowledge is used. The experimental results align with previous studies (Gururangan et al., 2020; Diao et al., 2021; Yao et al., 2021), indicating that continuous pre-training of a PLM on a target corpus can facilitate the acquisition of target domain knowledge, leading to enhanced performance on downstream tasks. Moreover, although the overall performance of LoRA TAPT is inferior to full-parameter TAPT, the difference between LoRA TAPT and full-parameter TAPT is negligible. On average, the difference on BioMed, CS, News, and Reviews is -0.13%, -0.47%, -0.80%, -0.02% on no n-gram setting, and -0.22%, -0.91%, -0.53%, -0.15% when n-gram information is fused through CANF module, respectively. Crucially, as depicted in Table 3, our proposed LoRA TAPT only requires $1.2MB$ while the full-parameter TAPT requires $478MB$ of disk space to store the model for downstream task fine-tuning. This observation illustrates that LoRA TAPT can significantly reduce the required disk space without sacrificing downstream task performance.

Comparison with Adapter TAPT LoRA TAPT brings more significant performance improvements when compared with Adapter TAPT. As shown in Table 2, LoRA TAPT with either n-gram or not is

Method	BioMed		CS		News	Reviews
	CP	RCT	AC	SE	HP	IMDB
NLoPT (10%)	$68.48_{1.46}$	$77.60_{0.68}$	$46.13_{3.53}$	$52.92_{4.60}$	$66.98_{3.87}$	$89.86_{0.90}$
NLoPT (20%)	$75.57_{0.87}$	$79.83_{0.47}$	$50.71_{3.01}$	$67.09_{1.62}$	$71.63_{5.38}$	$91.30_{0.49}$
NLoPT (50%)	$81.25_{0.74}$	$82.92_{0.28}$	$66.21_{2.19}$	$81.23_{1.14}$	$80.95_{1.83}$	$92.73_{0.32}$
NLoPT (100%)	$\mathbf{83.75_{0.04}}$	$\mathbf{83.65_{0.22}}$	$\mathbf{75.73_{0.41}}$	$\mathbf{85.58_{0.06}}$	$\mathbf{91.89_{1.30}}$	$\mathbf{93.55_{0.09}}$
gpt-3.5-turbo	56.85	68.60	50.36	67.35	72.31	78.38

Table 4: Comparison performances between our proposed NLoPT under different data size ratio and zero-shot *gpt-3.5-turbo*.

	Rank r	$r=1$	$r=4$	$r=8$	$r=16$	$r=32$	$r=64$	$r=128$
	# of Params	$0.68M$	$0.79M$	$0.94M$	$1.24M$	$1.82M$	$3.01M$	$5.36M$
	Storage Size	$0.16MB$	$0.58MB$	$1.20MB$	$2.30MB$	$4.6MB$	$9.1MB$	$19MB$
RCT	w/o n-gram	$87.12_{0.01}$	$87.26_{0.03}$	$87.41_{0.06}$	$87.39_{0.04}$	$87.41_{0.03}$	$87.44_{0.02}$	$\mathbf{87.47_{0.02}}$
	w/ n-gram	$87.24_{0.03}$	$87.40_{0.02}$	$87.60_{0.05}$	$87.62_{0.02}$	$87.65_{0.04}$	$87.62_{0.04}$	$\mathbf{87.69_{0.03}}$
SE	w/o n-gram	$83.25_{0.08}$	$84.01_{0.20}$	$84.59_{0.03}$	$84.47_{0.09}$	$84.60_{0.16}$	$\mathbf{84.84_{0.14}}$	$84.70_{0.17}$
	w/ n-gram	$84.12_{0.18}$	$84.63_{0.39}$	$85.58_{0.06}$	$85.57_{0.27}$	$85.67_{0.24}$	$85.96_{0.12}$	$\mathbf{86.00_{0.09}}$
HP	w/o n-gram	$87.54_{1.42}$	$89.56_{0.91}$	$90.28_{1.29}$	$91.13_{0.60}$	$91.12_{1.74}$	$91.75_{2.57}$	$\mathbf{91.77_{2.45}}$
	w/ n-gram	$88.84_{0.98}$	$90.51_{0.95}$	$91.89_{1.30}$	$91.44_{0.64}$	$92.11_{0.96}$	$92.41_{1.51}$	$\mathbf{93.06_{0.59}}$
IMDB	w/o n-gram	$94.75_{0.19}$	$95.03_{0.09}$	$95.26_{0.10}$	$95.24_{0.06}$	$95.29_{0.05}$	$95.26_{0.12}$	$\mathbf{95.31_{0.04}}$
	w/ n-gram	$95.36_{0.15}$	$95.44_{0.14}$	$95.81_{0.19}$	$95.79_{0.04}$	$95.82_{0.08}$	$95.85_{0.16}$	$\mathbf{95.88_{0.14}}$

Table 5: Performances averaged over 5 independent runs on the RCT, SE, HP and IMDB dataset with different rank r on two settings, i.e. without n-gram and with CANF module. w/ and w/o indicate whether the model is equipped with n-gram or not.

more effective at improving downstream task performance than Adapter TAPT. Specially, our method outperforms Adapter TAPT on 7 out of 8 datasets. As suggested by the results in 3, although Adapter TAPT has more trainable parameters, it only surpasses LoRA TAPT by 0.05% on average on the SE dataset, while poorer performance on other datasets. We attribute these experimental results to the enhanced efficiency of LoRA TAPT in selecting useful knowledge for downstream tasks compared to Adapter TAPT.

6.2. Superiority of CANF Module

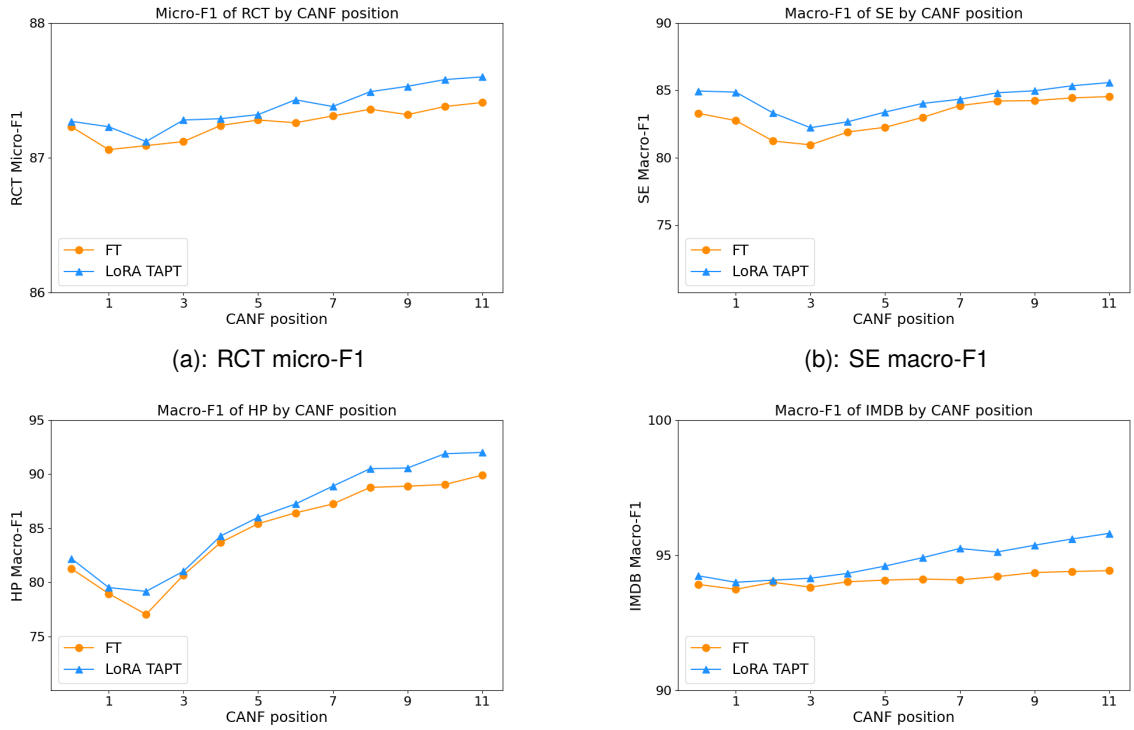
Table 2 illustrates that, regardless of the TAPT methods, the RoBERTa model enhanced with the CANF module consistently outperforms the one without it across all datasets. This clearly demonstrates the effectiveness of the CANF module. Notably, we find that the CANF module can yield considerable gains in certain datasets. For example, it can bring an average performance improvement of 1.37% on the AC dataset. A plausible explanation might be the significant domain discrepancy between the RoBERTa pre-training and the Computer Science (CS) domain. Therefore, the incorporation of multi-grained information from task-specific n-grams can yield substantial gains. Moreover, we compare CANF module with TDNA. As described in Table 2, CANF module demonstrates superior performance in comparison to TDNA on all settings. Specifically, when we compare the improvements over four domains, we observe gains of 0.43%, 0.67%, 1.63% and 0.48% in the BioMed, CS, News, and Reviews domains, respectively. Overall, our CANF module can consistently benefit downstream task performances and surpasses TDNA on all settings.

7. Analysis

In this section, we first compare NLoPT under different dataset size ratios with ChatGPT, then we analyze several aspects of NLoPT, including the effects of rank r during LoRA TAPT and effects of the position of CANF module. The details are shown as follows.

7.1. Comparison with ChatGPT

The recent remarkable success of LLMs has shifted the research paradigm of natural language processing, as exemplified by ChatGPT (OpenAI, 2022). In this subsection, we compare NLoPT under different data sizes with zero-shot ChatGPT. Specifically, we first randomly select 10% from IMDB dataset and 1% from RCT dataset to constrain all the datasets within the order of thousands. Then we sample different ratios (i.e., 10%, 20%, 50%, 100%) of the six datasets for supervised fine-tuning. Experimental results are reported in Table 4.



(a): RCT micro-F1

(b): SE macro-F1

(c): HP macro-F1

(d): IMDB macro-F1

Figure 2: Performances of RCT, SE, HP and IMDB datasets with regard to different CANF module positions, i.e., L in Figure 1, under both with FT (i.e., no TAPT) and LoRA TAPT settings.

Surprisingly, we find that NLoPT trained with a ratio of 10% can still outperform zero-shot *gpt-3.5-turbo* on three of six classification tasks. When the ratio comes to 20%, NLoPT can outperform zero-shot *gpt-3.5-turbo* on five of six classification tasks. A plausible explanation is that ChatGPT still lacks the reasoning ability in addressing tasks that require domain-specific knowledge.

7.2. LoRA Rank r

The LoRA rank r is the hyperparameter that allows a straightforward way to balance performance and parameter efficiency. Here, we turn our attention to the effect of rank r on model performance. We use RCT from BIOMED, SE from CS, HP from NEWS and IMDB from REVIEWS for our analysis. Table 5 reveals the effect of LoRA rank r on downstream task performance.

We find that increasing the rank r may not always lead to performance improvements. For example, the peak performance of SciErc under no n-gram setting is achieved when r equals 64. Moreover, the model exhibits significant performance improvement as the rank r increases, provided r is less than 8. Nevertheless, when r exceeds 8, the rate of increase in performance relative to r starts to plateau. This observation suggests that a low rank, such as $r = 8$, which has only $29.4k$ trainable parameters during LoRA TAPT, is sufficient for task adaptive pre-training.

7.3. CANF Module Position

In this subsection, We explore the impact of the position of the CANF module, i.e., the position where task-specific n-gram information is injected, on the performance of downstream tasks. Similar to subsection 7.2, we use RCT, SC, HP, and IMDB datasets for analysis. We conduct experiments on two settings, including FT (i.e., no TAPT) and LoRA TAPT.

As shown in Figure 2, We observe that as the position where CANF module is plugged increases, there is initially a decline in performance on downstream tasks, followed by an improvement on all datasets. Notably, the best performance is consistently obtained when we fuse n-gram information at the 11-th layer of the RoBERTa. This is not surprising, as BERT-based MLMs tend to capture high-level semantic features in higher layers (Jawahar et al., 2019). This analysis indicates that incorporating task-specific n-gram information at the final layer is the most effective method to equip RoBERTa with task-specific multi-granularity information.

8. Conclusion

In this work, we present NLoPT, a simple yet effective two-step process aiming to tailor a Pre-trained Masked Language Model to a domain-specific downstream task. We first apply LoRA for task-adaptive pre-training, and then inject task-specific multi-granularity n-gram information via a cross-attention based n-gram fusion module CANF. We verify NLoPT on six datasets from four domains. The experimental results show that explicitly incorporating task-specific n-grams can offer large gains. In addition, our LoRA TAPT performs competitively to conventional full-parameter TAPT and surpasses Adapter TAPT on downstream tasks. Further analysis reveals the effects of LoRA rank r and CANF module position on downstream tasks.

9. Acknowledgements

This work is supported by the Scientific and Technological Innovation Important Plan of China (No. 2021ZD0201502), the National Natural Science Foundation of China (NSFC) (No. 62322120, No. 62306316, No.61831022, No.U21B2010, No.62101553, No.61971419, No.62006223, No. 62206278, No.62201572).

10. Bibliographical References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7319–7328. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. PERL: pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Trans. Assoc. Comput. Linguistics*, 8:504–521.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. 2021. Taming pre-trained language models with n-gram representations for low-resource domain adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3336–3349. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4884–4896. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: word-level adversarial reprogramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4921–4933. Association for Computational Linguistics.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4237–4247. Association for Computational Linguistics.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Seungwon Kim, Alex Shum, Nathan Susanj, and Jonathan Hilgart. 2021. Revisiting pretraining with adapters. In *Proceedings of the 6th Workshop on Representation Learning for NLP, RepL4NLP@ACL-IJCNLP 2021, Online, August 6, 2021*, pages 90–99. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021,*

(Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 4582–4597. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1022–1035.

Kosuke Nishida, Kyosuke Nishida, and Sen Yoshida. 2021. Task-adaptive pre-training of language models with word embedding regularization. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4546–4553. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. OpenAI Blog.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 460–470. Association for Computational Linguistics.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y. S. Lam. 2021a. Effectiveness of pre-training for few-shot intent classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1114–1120. Association for Computational Linguistics.

Rongsheng Zhang, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. 2021b. Unsupervised domain adaptation with adapter. *CoRR*, abs/2111.00667.

11. Language Resource References

Franck Dernoncourt and Ji Young Lee. 2017. *PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts*. Asian Federation of Natural Language Processing. PID https://github.com/Franck-Dernoncourt/pubmed-rct.

David Jurgens and Srijan Kumar and Raine Hoover and Daniel A. McFarland and Dan Jurafsky. 2018. *Measuring the Evolution of a Scientific Field through Citation Frames*. PID http://jurgens.people.si.umich.edu/citation-function/.

Johannes Kiesel and Maria Mestre and Rishabh Shukla and Emmanuel Vincent and Payam Adineh and David P. A. Corney and Benno Stein and Martin Potthast. 2019. *SemEval-2019 Task 4: Hyperpartisan News Detection*. Association for Computational Linguistics. PID https://pan.webis.de/semeval19/semeval19-web/.

Jens Kringelum and Sonny Kim Kjærulff and Søren Brunak and Ole Lund and Tudor I. Oprea and Olivier Taboureau. 2016. *ChemProt-3.0: a global chemical biology diseases mapping*. PID https://biocreative.bioinformatics.udel.edu/news/corpora/chemprot-corpus-biocreative-vi/.

Yi Luan and Luheng He and Mari Ostendorf and Hannaneh Hajishirzi. 2018. *Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction*. Association for Computational Linguistics. PID http://nlp.cs.washington.edu/sciIE/.

Andrew L. Maas and Raymond E. Daly and Peter T. Pham and Dan Huang and Andrew Y. Ng and Christopher Potts. 2011. *Learning Word Vectors for Sentiment Analysis*. The Association for Computer Linguistics. PID https://ai.stanford.edu/ amaas/data/sentiment/.