# A Robustly Optimized BERT Pre-training Approach with Post-training

**Zhuang Liu**[1,†], **Wayne Lin**[2], **Ya Shi**[3], **Jun Zhao**[4]
[1]Dongbei University of Finance and Economics, Dalian, China
[2]University of Southern California, LA, USA
[3]Union Mobile Financial Technology, Beijing, China
[4]IBM Research, Beijing, China

## Abstract

In the paper, we present a '***pre-training***'+'***post-training***'+'***fine-tuning***' three-stage paradigm, which is a supplementary framework for the standard '***pre-training***'+'***fine-tuning***' language model approach. Furthermore, based on three-stage paradigm, we present a language model named PPBERT. Compared with original BERT architecture that is based on the standard two-stage paradigm, we do not fine-tune pre-trained model directly, but rather post-train it on the domain or task related dataset first, which helps to better incorporate task-awareness knowledge and domain-awareness knowledge within pre-trained model, also from the training dataset reduce bias. Extensive experimental results indicate that proposed model improves the performance of the baselines on 24 NLP tasks, which includes eight GLUE benchmarks, eight SuperGLUE benchmarks, six extractive question answering benchmarks. More remarkably, our proposed model is a more flexible and pluggable model, where post-training approach is able to be plugged into other PLMs that are based on BERT. Extensive ablations further validate the effectiveness and its state-of-the-art (SOTA) performance. The open source code, pre-trained models and post-trained models are available publicly.

## 1 Introduction

Recently, the introduction of pre-trained language models (PLMs), including GPT (Radford et al., 2018), BERT (Devlin et al., 2019), and ELMo (Peters et al., 2018), among many others, has achieved tremendous success to the natural language processing (NLP) research. Typically, the basic structure of such a model consists of two successive stages, one step during the pre-training phase and another step during the fine-tuning phase. During the pre-training phase it pre-trains on unsupervised dataset firstly, then during the fine-tuning phase it fine-tunes on downstream supervised NLP tasks. Up to now, these models obtained the best performance on various NLP tasks. Some of the most prominent examples are BERT, and BERT based SpanBERT (Joshi et al., 2019), ALBERT (Lan et al., 2020). These PLMs are trained on the large unsupervised corpus through some unsupervised training objectives. However, it is not obvious that the model parameters which is obtained during unsupervised pre-training phase can be well-suited to support the this kind of transfer learning. Especially during the fine-tuning phase, for the target NLP task only a small amount of supervised text data is available, fine-tuning the pre-trained model are potentially brittle. And for the pre-trained model, supervised fine-tuning requires substantial amounts of task-specific supervised training dataset, not always available. For example, in GLUE benchmark (Wang et al., 2019b), Winograd Schema dataset (Levesque et al., 2012) have only 634 training data, too small for fine-tuning natural language inference (NLI) task. Moreover, although PLMs, such BERT, can learn contextualized representations across many NLP tasks (to be task-agnostic), which leverages PLMs alone still leaves the domain-specific challenges unresolved (BERT are trained on general domain corpora only, and capture a general language knowledge from training dataset, but lack domain or task-specific data severely). For example, in financial domain, they often contain unique vocabulary information, such as stock, bond type, and the sizes of labeled data are also very small (even only few hundreds of samples).

† Corresponding Author: liuzhuang@dufe.edu.cn

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, August 13 - 15, 2021.

In the paper, to overcome the aforementioned issues, we proposed a novel three-stage BERT (called PPBERT) architecture, in which we add a second stage of training, that is '***post-training***', to improving the original BERT architecture model.

Typically there are two directions to pursue new state-of-art in the post pre-trained PLMs era. One is to construct novel neural network architecture model based on PLMs, like BERTserini (Yang et al., 2019a) and BERTCMC (Ohsugi et al., 2019). Other approach is to optimize pre-training, like GPT 2.0 (Radford et al., 2018), MT-DNN (Liu et al., 2020a), SpanBERT (Joshi et al., 2019), and ALBERT (Lan et al., 2020). In the paper, we present another novel method to improve the PLMs. We present a '***pre-training***'+'***post-training***'+'***fine-tuning***' three-stage paradigm and further present a language model named PPBERT. Compared with original BERT architecture that is based on the standard '***pre-training***'+'***fine-tuning***' PLMs approach, we do not fine-tune pre-trained models directly, but rather ***post-train*** them on the domain or task related training dataset first, which helps to better incorporate task-awareness knowledge and domain-awareness knowledge within pre-trained model, also in the training dataset can reduce bias. More specifically, our framework involves three sequential stages: pre-training stage using on large-scale corpora (see subsection 2.1), post-training stage using the task or domain related datasets via multi-task continual learning method (see subsection 2.2), and fine-tuning stage using target datasets, even with little labeled samples or without labeled samples (see subsection 2.3). Thus, PPBERT can benefits from the regularization effect since it leverages cross-domain or cross-task data, which helps model generalize better with limited data and adapt to new domains or tasks better.

Sum up, on a wide variety of tasks our proposed post-training process outperforms existing BERT benchmark, and achieved better performance on small dataset and domain-specific tasks in particular substantially. Specifically, we compared our model with BERT baselines on GLUE and SuperGLUE benchmark tasks and consistently significantly outperform BERT on all of 16 tasks (8 GLUE tasks and 8 SuperGLUE tasks), increasing by the GLUE average score of 87.02, showing an absolute improvement of 2.97 over BERT; showing an absolute improvement of 5.55, pushing the SuperGLUE to 74.55. More remarkably, our model is a more flexible and pluggable. The post-training appoach can be straight plugged into other PLMs based on BERT. In our ablation studies, we plug the post-training strategy into original BERT (i.e., PPBERT) and its variant, ALBERT (called PPALBERT), respectively. Our approaches advanced the SOTA results for five popular question answering datasets, surpassing the previous pre-trained models by at least 1 point in absolute accuracy. Moreover, through further ablation studies, the best model obtains SOTA results on small datasets (1/20 training set). All of these clearly demonstrate our proposed three-stage paradigms exceptional generalization capability via post-training learning.

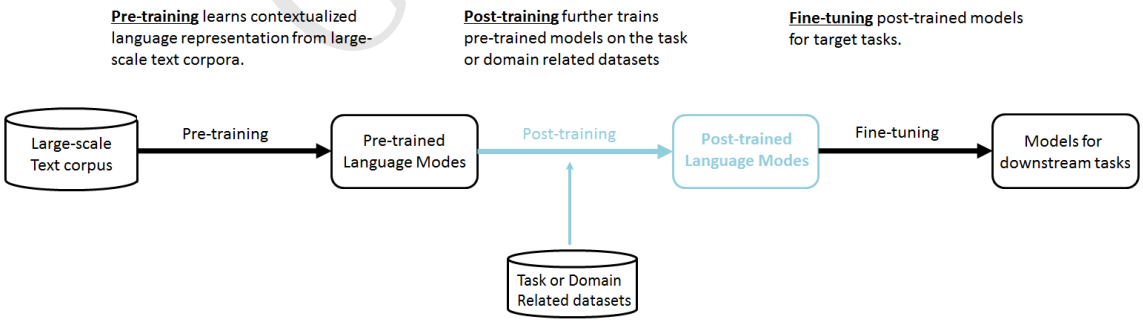


Figure 1: An illustration of the architecture for our PPBERT, which is a '*pre-training*'-'*post-training*'-then-'*fine-tuning*' three-stage BERT. Compared with standard BERT architecture that has the two-stage '*pre-training*'-then-'*fine-tuning*', we do not directly fine-tune pre-trained models, but rather add a second stage of training (called '*post-training*'). More specifically, during the pre-training stage, we first on the large-scale dataset conduct unsupervised pre-training, and then during the post-training stage post-train pre-trained models on the task or domain related dataset, and last during the fine-tuning stage conduct fine-tuning on downstream supervised NLP tasks.



Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, August 13 - 15, 2021.

## 2 The Proposed Model: PPBERT

As shown in Figure 1, the standard BERT is built based on two-stage paradigm architecture, '*pre-training*'+'*fine-tuning*'. Compared traditional pre-training methods, PPBERT does not fine-tune the pre-trained model directly after pre-training, but rather continues to post-train the pre-trained model on the task or domain related corpus, helping to reduce bias. During post-training processing our proposed PPBERT framework can continuously update pre-trained model. The architecture of our PPBERT architecture is shown in Figure 1.

### 2.1 Pre-training

The training procedure of our proposed PPBERT has 2 processing: pre-training stage and post-training stage. As BERT outperforms most existing models, we do not intend to re-implement it but focus on the second training stage: Post-training. The pre-training processing follows that of the BERT model. We first use original BERT and further adopt a joint post-training method to enhance BERT. Thus, our proposed PPBERT is more flexible and pluggable, where post-training approach is able to be plugged into other language models based on BERT, such as ALBERT (Lan et al., 2020), SpanBERT (Joshi et al., 2019), not only applied to original BERT.

### 2.2 Post-training

Compared with original BERT architecture that has two-stage paradigm, '*pre-training*'+'*fine-tuning*', we do not fine-tune pre-trained model, but rather first *post-train* the model on the task or domain related training dataset directly. We add a second training stage, that is '*post-training*' stage, on an intermediate task before target-task fine-tuning.

#### 2.2.1 Training Details

In the post-training stage, its aims to train the pre-trained model on the task or domain related annotated data continuously, to learn task knowledge or domain knowledge from different post-training tasks by keeping updating the pre-trained model. Thus, it brings a big challenge: How to train these post-training tasks in a continual way, and more efficiently post-train a new task without forgetting the knowledge that is learned before.

Inspired by (Chen and Liu, 2018; Sun et al., 2019) and (Parisi et al., 2019), which show Continual Learning can train the model with several tasks in sequence, but we find that, standard Continual Learning method trains the model with only one task at each time with the demerit that it is easy to forget the knowledge previously learned. Also concurrently, inspired by (Liu et al., 2020a; Liu et al., 2020b) and (Hou et al., 2020; Liu et al., 2020c), which show Multi-task Learning can allow the use of different training corpus to train sub-parts of neural networks, but we find that, although Multi-task Learning could train multiple tasks at the same time, it is necessary that all customized pre-training tasks are prepared before the training could proceed. So this method takes as much time as continual learning does, if not more. So we present a multi-task continual learning method to tackle with this problem. More specifically, whenever a new post-training task comes, the multi-task continual learning method first utilizes the parameters that is previously learned to initialize the model, and then simultaneously train the newly-introduced task together with the original tasks, which will make sure that the learned parameters can encode the knowledge that is previously learned. More crucially, during post-training we allocate each task $K$ training iterations, and then further assign these $K$ iterations for each task to different stages of training. Also concurrently, instead of updating parameters over a batch, we divide a batch into more sub-batches and accumulate gradients on those sub-batches before parameter updates, which allows for a smaller sub-batch to be consumed in each iteration, more conducive to iterating quickly by using distributed training. As a result, proposed PPBERT can continuously update pre-trained model using the multi-task continual learning method. So we can guarantee the efficiency of our post-training without forgetting the knowledge that is previously trained.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, Augest 13 - 15, 2021.

#### 2.2.2 Post-training Datasets

As discussed above, fine-tuning processing has main challenges, on the target task directly, as follows: **i)** during the fine-tuning phase, there is only a small amount of supervised training data, fine-tuning the pre-trained model are potentially brittle; **ii)** for the pre-trained model, its supervised fine-tuning requires substantial amounts of task-specific supervised training dataset, limited and indirect, not always available; **iii)** leveraging BERT alone leaves the domain or task-specific questions unresolved. To enhance the performance of pre-trained model, we need to effectively fuse task knowledge (from related NLP tasks supervised data) or domain knowledge (from related in-domain supervised data). As a common NLP task, Questions and Answers (QA) ,to get the answer based on a question, requires reasoning on facts relevant to the given question and deep semantic understanding of document. Thus, a large-scale QA supervised corpus can benefit most NLP tasks. Similarly, NLI task (a.k.a. RTE) and sentiment analysis (SA) are also two important and basic tasks for natural language understanding. Eventually, we use QA dataset (CoQA), NLI dataset (SNLI) and SA dataset (YELP) as post-training datasets. We post-train our model on CoQA, SNLI and YELP data simultaneously.

In this work, for generality and wide applicability of our proposed PPBERT, we use only CoQA, SNLI and YELP as post-training datasets. Note that, because PPBERT adopts the effective multi-task continual learning training method (subsubsection 2.2.1), its post-training datasets are easily scalable, which is meant to be combined further with other datasets, including domain specific data.

### 2.3 Fine-tuning

In fine-tuning processing, we first initialize PPBERT model with the post-trained parameters, and then use supervised dataset from specific tasks to further fine-tune. In general, for each downstream task, after being fine-tuned it has its own fine-tuned models.

## 3 Experiments

### 3.1 Tasks

To evaluate our proposed approach, we use a comprehensive experiment tasks, as follows:

**i)** in section 3, eight tasks in the GLUE benchmark (Wang et al., 2019b) and eight tasks in the SuperGLUE benchmark (Wang et al., 2019a);

**ii)** in section 4, five question answering tasks, two natural language inference tasks and two tasks in domain adaptation, financial sentiment analysis and financial question answering.

We expect that these NLP tasks will benefit from proposed '***pre-training***'+'***post-training***'+'***fine-tuning***' three-stage paradigm particularly.

### 3.2 Datasets

This subsection briefly describes the datasets.

#### 3.2.1 GLUE

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019b) is a collection of eight datasets to evaluate NLU tasks. GLUE[0] consists of a series of NLP task datasets (See Table 1), including: Corpus of Linguistic Acceptability (CoLA), Multi-genre Natural Language Inference (MNLI), Recognizing Textual Entailment (RTE), Quora Question Pairs (QQP), Semantic Textual Similarity Benchmark (STS-B), Stanford Sentiment Treebank (SST-2), Question Natural Language Inference (QNLI), Microsoft Research Paraphrase Corpus (MRPC).

#### 3.2.2 SuperGLUE

Similar to GLUE, the SuperGLUE benchmark (Wang et al., 2019a) is a new benchmark that is more difficult language understanding task datasets[1], including : BoolQ, CommitmentBank (CB), Choice of

[0]https://gluebenchmark.com/

[1]https://super.gluebenchmark.com/

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, August 13 - 15, 2021.

Table 1: Summary of the GLUE benchmark.

| Corpus | Task | #Train | #Dev | #Test | Metrics |
|---|---|---|---|---|---|
| CoLA | Acceptability | 8.5k | 1k | 1k | Matthews corr |
| STS-B | Similarity | 7k | 1.5k | 1.4k | Pearson/Spearman corr |
| QQP | Paraphrase | 364k | 40k | 391k | Accuracy/F1 |
| MRPC | Paraphrase | 3.7k | 408 | 1.7k | Accuracy/F1 |
| SST-2 | Sentiment | 67k | 872 | 1.8k | Accuracy |
| QNLI | QA/NLI | 108k | 5.7k | 5.7k | Accuracy |
| MNLI | NLI | 393k | 20k | 20k | Accuracy |
| RTE | NLI | 2.5k | 276 | 3k | Accuracy |

***Notes:*** The details of GLUE benchmark. The #Train, #Dev and #Test denote the size of the training set, development set and test set of corresponding corpus respectively.

Plausible Alternatives (COPA), Multi-Sentence Reading Comprehension (MultiRC), Reading Comprehension with Commonsense Reasoning (ReCoRD), Recognizing Textual Entailment (RTE), Words in Context (WiC), Winograd Schema Challenge (WSC).

#### 3.2.3 SQuAD

The Stanford Question Answering Dataset (SQuAD) is one of the most popular machine reading comprehension challenges datasets. SQuAD is a typical extractive machine reading comprehension task, including a question and a paragraph of context. Its aim is to give a text span extracted from the document based on the given question. SQuAD consists of two versions: SQuAD (Rajpurkar et al., 2016) (in this version, the provided document always contains an final answer) and SQuAD v2.0 (Rajpurkar et al., 2018) (in this version, some questions are not answered from the provided document).

#### 3.2.4 Financial datasets

To better demonstrate the generality of our post-training approach, we further perform domain adaptation experiments on two financial tasks, FiQA sentiment analysis (SA) dataset and FiQA question answering (QA) dataset. As part of the companion proceedings for WWW'18 conference, (Maia et al., 2018) released two very small financial datasets (FiQA).

#### 3.2.5 Additional benchmarks

As shown in Table 6, we present additional datasets for extractive question answering tasks, including RACE (Lai et al., 2017), NewsQA (Trischler et al., 2017), TrivaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018). More details are provided in the supplementary materials.

### 3.3 Experimental Results

We evaluate the proposed PPBERT on two popular NLU benchmarks: GLUE and SuperGLUE . We compare PPBERT with standard BERT model and demonstrate the effectiveness of with '***post-training***'.

#### 3.3.1 GLUE Results

We evaluated performance on GLUE benchmark, with the large models and the base models of each approach. We reports the results of each method on the development dataset and test dataset. The detailed experimental results on GLUE are presented in Table 2. As illustrated in the BASE models columns of Table 2, $\text{PPBERT}_{\text{BASE}}$ achieves an average score of 81.53, and outperforms standard $\text{BERT}_{\text{BASE}}$ on all of the 8 tasks. As shown, in test dataset parts of LARGE models sections in Table 2, $\text{PPBERT}_{\text{LARGE}}$ outperform $\text{BERT}_{\text{LARGE}}$ on all of the 8 tasks and achieves an average score of 85.03. We also observe similar results in the dev set column, achieveing an average score of 87.02 on the dev set, a 2.97 improvement over $\text{BERT}_{\text{LARGE}}$. From this data we can see that $\text{PPBERT}_{\text{LARGE}}$ matched or even outperformed human level.

Table 2: The overall performance of PPBERT and the comparison against BERT models on GLUE benchmark.

| | | BASE model | | LARGE model | | | |
|---|---|---|---|---|---|---|---|
| | | Test Set | | Dev Set | | Test Set | |
| **Task** | **Human Perf.** | **BERT**$^{\dagger}$ | **PPBERT**$^{\ddagger}$ | **BERT**$^{\dagger}$ | **PPBERT**$^{\ddagger}$ | **BERT**$^{\dagger}$ | **PPBERT**$^{\ddagger}$ |
| CoLA | 66.4 | 52.1 | 52.3 | 60.6 | 61.3 | 60.5 | 61.1 |
| SST-2 | 97.8 | 93.5 | 94.6 | 93.2 | 95.7 | 94.9 | 95.7 |
| MRPC | 86.3/80.8 | 84.8/88.9 | 85.7/89.2 | 88.0 | 89.6 | 85.4/89.3 | 87.2/90.2 |
| STS-B | 92.7/92.6 | 87.1/85.8 | 87.6/86.5 | 90.0 | 91.3 | 87.6/86.5 | 90.5/89.8 |
| QQP | 59.5/80.4 | 89.2/71.2 | 88.8/73.0 | 91.3 | 92.2 | 89.3/72.1 | 90.6/73.9 |
| MNLI | 92.0/92.8 | 84.6/83.4 | 85.9/85.1 | 86.6 | 88.7 | 86.7/85.9 | 88.3/88.4 |
| QNLI | 91.2 | 90.5 | 92.2 | 92.3 | 93.8 | 92.7 | 93.7 |
| RTE | 93.6 | 66.4 | 72.3 | 70.4 | 84.2 | 70.1 | 80.3 |
| (Avg) | 85.94 | 80.00 | 81.53 (1.53 ↑) | 84.05 | 87.02 (2.97 ↑) | 82.45 | 85.03 (2.58 ↑) |

***Notes:*** The results on GLUE benchmark (Wang et al., 2019b), where the results on test set are scored by the GLUE evaluation server and the results on dev set are the median of three experimental results. The metrics for these tasks are shown in Table 1. Purple-colored texts indicate the results on par with or pass human performance. $^{\ddagger}$ indicates our proposed model. $^{\dagger}$ indicates original model BERT (Devlin et al., 2019).

Table 3: Results on SuperGLUE benchmark.

| *Single Model* | **BoolQ** | **CB** | **COPA** | **MultiRC** | **ReCoRD** | **RTE** | **WiC** | **WSC** | (Avg) |
|---|---|---|---|---|---|---|---|---|---|
| Human Perf.$^{\S}$ | 89.0 | 95.8/98.9 | 100.0 | 81.8/51.9 | 91.7/91.3 | 93.6 | 80.0 | 100.0 | 89.79 |
| BERT$^{\S}$ | 77.4 | 75.7/83.6 | 70.6 | 70.0/24.1 | 72.0/71.3 | 71.7 | 69.6 | 64.4 | 69.00 |
| PPBERT (ours) | 80.3 | 81.4/86.9 | 74.2 | 76.5/40.7 | 78.7/77.5 | 77.4 | 72.9 | 68.7 | 74.55 |

***Notes:*** All results are based on a 24-layer architecture (LARGE model). PPBERT results on the development set are a median over three runs. Model references: $^{\S}$: ((Wang et al., 2019a)).

### 3.3.2 SuperGLUE Results

Table 3 shows the performances on 8 SuperGLUE tasks. As shown in Table 3, it is apparent that PPBERT outperforms BERT on 8 tasks significantly. The main gains from PPBERT are in the MultiRC (+6.5) and in ReCoRD (+6.7), both accounting for the rise in PPBERT's GLUE score. Also, as Table 3 shows, there is a huge gap between human performance (89.79) and the performance of PPBERT (74.55).

### 3.3.3 Overall Trends

Table 2 and Table 3 respectively show our results on GLUE and SuperGLUE with and without '***post-training***'. As shown, we compare proposed method to standard BERT benchmarks on 16 baseline tasks, and find on every task our proposed PPBERT outperforms BERT. Since in pre-training phase PPBERT has the same architecture and pre-training objective as standard BERT, the main gain is attributed to '***post-training***' in post-training phase. If we consider the gains, especially PPBERT is better at natural language inference and question answering tasks, and is not good at syntax-oriented task. In GLUE benchmark (we also observe similar results in SuperGLUE), for example, **i)** for the question answering tasks (QNLI, MultiRC, ReCoRD) and the natural language inference tasks (MNLI and RTE), we achieves significant accuracy gain of at least 1 point improvement. **ii)** for sentiment task (SST-2), although we observe a smaller gain (+0.8), it is mainly because the accuracy has been already high, a reasonable score (obtained a accuracy score of 95.7); **iii)** for simple sentence task, we observe the smallest gain (+0.2) on all tasks in the syntax-oriented (CoLA) task. Besides, this mirrors results also reported in (Bowman et al., 2018), who show that few pre-training tasks other than language modeling offer any advantage for CoLA. **iv)** for MRPC and RTE tasks, as shown in Table 2 and Table 3, what is interesting in the results is that we find consistent improvements after post-trainingThis reveals that the learned PPBERT representation by '***pre-training***'+'***post-training***' allows much more effective domain adaptation than the BERT representation by '***pre-training***' only.

## 4 Ablation Study and Analyses

### 4.1 Cooperation with other Pre-trained LMs

Our proposed PPBERT is a more flexible and pluggable, where post-training approach can be plugged into other PLMs based on BERT, not only applied to original BERT model. We further validate the performance of PPBERT when '***post-training***' appoach on different pre-trained LMs. We compare post-training by plugging it into original BERT (i.e., PPBERT) and and its variant, ALBERT (called PPALBERT) pre-trained LMs, respectively. Also, we further post-train the most recent proposed PPALBERT with one additional QA dataset (SearchQA), and call it PPALBERT$_{\text{LARGE}}$-$QA$.

#### 4.1.1 Comparisons to SOTA models

We evaluate our models on the popular SQuAD benchmark (subsubsection 3.2.3). Performance of each model is evaluated on the two standard metric values: F1 score and exact match (EM) score. F1 score measures the precision and recall, and less strict than then EM score. EM score measures whether the model output exactly matches the ground answers.

Table 4 details performance gains when exploiting each of the three post-trained LMs on SQuAD datasets (two versions, respectively). As shown in Table 4, on the SQuAD dev dataset (version 1.1), compared with BERT baseline, adding post-training stage improves the EM by 1.1 points (84.1→85.2), and F1 1.2 points (90.9→92.1). Similarly, PPALBERT$_{\text{LARGE}}$ also outperforms ALBERT$_{\text{LARGE}}$ baseline, by 0.3 EM and 0.2 F1. Especially, PPALBERT$_{\text{LARGE}}$-$QA$ using further post-training relatively improves 0.1 EM and 0.1 F1 over PPALBERT$_{\text{LARGE}}$, respectively. We also observe similar results on SQuAD v2.0 development set. The most recent proposed PPALBERT sets a new state-of-the-art, achieving 87.7 EM and 90.5 F1.

Table 4: Comparison with state-of-the-art results on the Dev set of SQuAD.

| | **SQuAD1.1** | **SQuAD2.0** |
|---|---|---|
| *Single Model* | EM/F1 | EM/F1 |
| Human Perf. | 82.3/91.2 | 86.8/89.5 |
| ALBERT$_{\text{BASE}}$ (Lan et al., 2020) | 82.1/89.3 | 76.1/79.1 |
| BERT$_{\text{LARGE}}$ (Devlin et al., 2019) | 84.1/90.9 | 79.0/81.8 |
| XLNet$_{\text{LARGE}}$ (Yang et al., 2019b) | 89.0/94.5 | 86.1/88.8 |
| RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019) | 88.9/94.6 | 86.5/89.4 |
| ALBERT$_{\text{LARGE}}$ (Lan et al., 2020) | 89.3/94.8 | 87.4/90.2 |
| PPBERT$_{\text{LARGE}}$ (ours) | 85.2/92.1 | 82.2/84.8 |
| PPALBERT$_{\text{LARGE}}$ (ours) | 89.6/95.0 | 87.6/90.4 |
| PPALBERT$_{\text{LARGE}}$-$QA$ (ours) | **89.7/95.1** | **87.7/90.5** |

***Notes:*** Results on SQuAD 1.1/2.0 development dataset. Best scores are in bold texts, and the previous best scores are underlined.

#### 4.1.2 Performance on other QA and NLI tasks

Furthermore, extensive experiments on six NLP tasks about semantic relationship are conducted, including two natural language inference benchmarks (QNLI and MNLI-m, both from GLUE), and four extractive question answering benchmarks (TriviaQA, RACE, HotpotQA and NewsQA). All benchmarks except RACE, we use the same fine-tuning method as SQuAD. Different from others, RACE is a multiple-choice QA dataset. The experimental results for PPALBERT are shown in Table 5. As depicted in Table 5, both PPALBERT$_{\text{LARGE}}$ and PPALBERT$_{\text{LARGE}}$-$QA$ achieve state-of-the-art accuracy across all settings. Overall, as expected, only utilizing '***pre-training***' is inferior to our proposed '***pre-training***'-then-'***post-training***' method. The experimental results (subsubsection 4.1.1 and subsubsection 4.1.2) described above, indicate that our two stage training paradigm is very flexible, and proposed post-training appoach could be easily plugged into other PLMs. More remarkably, we achieve new SOTA performances on existing baselines.



Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, August 13 - 15, 2021.

Table 5: Performance on six QA and NLI tasks.

| *Single Model* | **NewsQA** | **TrivaQA** | **HotpotQA** | **RACE** | **QNLI** | **MNLI-m** |
|---|---|---|---|---|---|---|
| $\text{BERT}_{\text{LARGE}}$ † | 68.8 | 77.5 | 78.3 | 72.0 | 92.3 | 86.6 |
| $\text{SpanBERT}_{\text{LARGE}}$ † | 73.6 | 83.6 | 83.0 | - | 93.3 | 87.0 |
| $\text{RoBERTa}_{\text{LARGE}}$ ‡ | - | - | - | 83.2 | 94.7 | 90.2 |
| $\text{ALBERT}_{\text{LARGE}}$ § | - | - | - | 86.5 | 95.2 | 90.4 |
| $\text{PPALBERT}_{\text{LARGE}}$ (ours) | 74.6 | 84.3 | 83.4 | 86.7 | 95.6 | 90.7 |
| $\text{PPALBERT}_{\text{LARGE}}$-$QA$ (ours) | **74.8** | **84.5** | **83.5** | **86.8** | **95.9** | **90.9** |

***Notes:*** The details of NewsQA, TrivaQA, HotpotQA and RACE are shown in Table 6. QNLI and MNLI-m are from GLUE. Model references: †: ((Joshi et al., 2019)), ‡: ((Liu et al., 2019)), §: ((Lan et al., 2020)).

Table 6: The details of QA datasets.

| **Dataset** | **Lang.** | **#Que.** | **#Docs** | **Que.** | **Docs** | **Answer Type** |
|---|---|---|---|---|---|---|
| SQuAD 1.1† | EN | 100K | 536 | CS | Wiki. | Span of words |
| SQuAD 2.0‡ | EN | 150K | 500 | CS | Wiki. | Span of words |
| NewsQA (Trischler et al., 2017) | EN | 100K | 10K | CS | CNN | Span of words |
| HotpotQA (Yang et al., 2018) | EN | 78K | 113k | CS | Wiki. | Span/substring of words |
| TrivaQA (Joshi et al., 2017) | EN | 40K | 660K | TW | Wiki./Web doc. | Span/substring of words |
| RACE (Lai et al., 2017) | EN | 870K | 50K | EE | EE | Multiple-choice |
| CoQA (Reddy et al., 2019) | EN | 127K | 8K | CS | QA Dialog | Span/substring of words |

***Notes:*** CS denotes Crowdsourced. TW denotes Trivia websites. EE denotes English exam. Model references: †: ((Rajpurkar et al., 2016)), ‡: ((Rajpurkar et al., 2018)).

## 5 Conclusion

In the paper, we present a '***pre-training***'+'***post-training***'+'***fine-tuning***' three-stage paradigm and a language model named PPBERT based on the three-stage paradigm, which is a supplementary framework for the standard '***pre-training***'+'***fine-tuning***' two-stage architecture. Our proposed three-stage paradigm helps to incorporate task-awareness knowledge and domain knowledge within pre-trained model, also reduce the bias in the training corpus. PPBERT can benefits from the regularization effect since it leverages cross-domain or cross-task data, which helps model generalize better with limited data and adapt to new domains or tasks better. With the latest PLMs as baseline and encoder backbone, PPBERT is evaluated on 24 well-known benchmarks, which outperformS strong baseline models and obtains new SOTA results. We hope this work can encourage further research into the language models training, and the future works involve the choice of other transfer learning sources such as CV etc.

## References

Samuel R. Bowman, Ellie Pavlick, and Edouard Grave. 2018. Looking for elmo's friends: Sentence-level pretraining beyond language modeling. *CoRR*, abs/1812.10860.

Zhiyuan Chen and Bing Liu. 2018. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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.

Ming Hou, Xinqi Chen, Shifeng Huang, Shengli Xie, and Guoxu Zhou. 2020. Generalizing deep multi-task learning with heterogeneous structured networks. In *In Proceedings of ICLR*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings*

*of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. 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.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge.

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.

Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, and Jianfeng Gao. 2020a. The microsoft toolkit of multi-task deep neural networks for natural language understanding. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 118–126. Association for Computational Linguistics.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020b. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, January 5-10, 2021, Yokohama, Japan*, pages 4513–4519.

Zhuang Liu, Kaiyu Huang, Degen Huang, Zhuang Liu, and Jun Zhao. 2020c. Dual head-wise coattention network for machine comprehension with multiple-choice questions. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1015–1024. ACM.

Macedo Maia, Siegfried Handschuh, Andre Freitas, Brian Davis, Ross Mcdermott, Manel Zarrouk, Alexandra Balahur, and Ross Mc-Dermott, editors. 2018. *In Proceedings of WWW*. ACM.

Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2019. A simple but effective method to incorporate multi-turn context with BERT for conversational machine comprehension. *CoRR*, abs/1905.12848.

German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *In Proceedings of Technical report, OpenAI*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, Augest 13 - 15, 2021.

Yu Sun, Shuohuan Wang, and Yu-Kun Li. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. *Proc. of the 2ndWorkshop on Representation Learning for NLP*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with bertserini. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *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, Demonstrations*, pages 72–77. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.



Proceedings of the 20th China National Conference on Computational Linguistics, pages 1218-1227, Hohhot, China, Augest 13 - 15, 2021.