UEGP: Unified Expert-Guided Pre-training for Knowledge Rekindle

Yutao Mou[1*], Kexiang Wang[2], Jianhe Lin[2], Dehong Ma[2], Jun Fan[2]
Daiting Shi[2], Zhicong Cheng[2], Simiu Gu[2], Weiran Xu[1†*]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Baidu Inc., Beijing, China
{myt,xuweiran}@bupt.edu.cn, {wangkexiang,madehong,fanjun}@baidu.com
{shidaiting01,chengzhicong01,gusimiu}@baidu.com
{linjianhe0309}@hotmail.com, {yindawei}@acm.org

Abstract

Pre-training and fine-tuning framework has become the standard training paradigm for NLP tasks and is also widely used in industrial-level applications. However, there are still a limitation with this paradigm: simply fine-tuning with task-specific objectives tends to converge to local minima, resulting in a sub-optimal performance. In this paper, we first propose a new paradigm: knowledge rekindle, which aims to re-incorporate the fine-tuned expert model into the training cycle and break through the performance upper bounds of experts without introducing additional annotated data. Then we further propose a unified expert-guided pre-training (UEGP) framework for knowledge rekindle. Specifically, we reuse fine-tuned expert models for various downstream tasks as knowledge sources and inject task-specific prior knowledge to pre-trained language models (PLMs) by means of knowledge distillation. In this process, we perform multi-task learning with knowledge distillation and masked language modeling (MLM) objectives. We also further explored whether mixture-of-expert guided pre-training (MoEGP) can further enhance the effect of knowledge rekindle. Experiments and analysis on eight datasets in GLUE benchmark and a industrial-level search re-ranking dataset show the effectiveness of our method.[1]

1 Introduction

In recent years, pre-trained language models (PLMs) have been widely used in various NLP tasks, such as sentiment classification, semantic matching, named entity recognition and etc., which generally adopt a two-stage training paradigm, i.e., pre-training and fine-tuning (Devlin et al., 2019; Radford et al., 2018). With powerful general language modeling capabilities, PLMs are also widely used as the backbones of search re-ranking, vector recall and other modules in information retrieval (Liu et al., 2021a; Zou et al., 2021), recommendation (Yao et al., 2021) and advertising systems (Qiao et al., 2019). In practical applications, we usually pre-train PLMs on a large-scale unlabeled general corpora, and then perform fine-tuning on a small-scale labeled dataset for downstream task to achieve the best performance. However, we find that simply fine-tuning PLMs with task-specific objectives is often sub-optimal, and the potential performance of PLMs remains to be exploited.

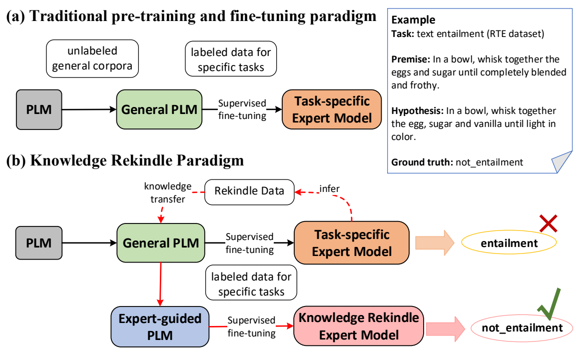


Figure 1: Comparison between traditional pre-training and fine-tuning paradigm and our proposed knowledge rekindle paradigm. We take a bad case of text entailment task as an example.

Recently, there is a main trend for the development of pre-trained language models: the scale of PLMs is increasing. The researchers find that the performance of PLMs could be further improved by simply scaling up the model capacity, training data size, and increasing the number of training steps (Kaplan et al., 2020). Representative works include GPT-3 (Brown et al., 2020), ERNIE3.0 (Sun et al., 2021) and etc. And Aghajanyan et al. (2020) also found that larger-scale PLMs have smaller intrinsic dimensions (Li et al., 2018), which means stronger generalization capabilities and higher per-

[*] This work was done during Yutao Mou's internship at Baidu Inc.

[1]We release our code at https://github.com/MurrayTom/UEGP

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2661–2673

formance on downstream tasks. As the capacity of PLMs continues to increase, they have stronger generalization ability and higher performance upper bound, but research shows that simply fine-tuning them with task-specific objectives such as cross-entropy, mean square error and etc., often makes the model converge to local minima, resulting in a sub-optimal performance (Mannor et al., 2005; Margolin, 2005).

To solve the problem, we first define a new paradigm, named "Knowledge Rekindle". The concept arises from the human learning process: after students have preliminary understanding of specific knowledge under the guidance of teachers, if they continue to learn independently, the students may eventually surpass the teacher. We find that the performance of fine-tuned expert models cannot be improved by further fine-tuning and the training cycle of the fine-tuned expert model is over, but according to previous research (Mannor et al., 2005; Margolin, 2005), the expert model is sub-optimal, so "Knowledge Rekindle" hopes to re-incorporate the expert model into the training cycle to further break through performance upper bounds rather than throwing it away, as shown in Fig 1. Next, we further propose a **U**nified **E**xpert-**G**uided **P**re-training (**UEGP**) framework for knowledge rekindle. Without loss of generality, we first collect a large amount of task-agnostic pre-training corpora ("rekindle data") from public websites such as Wikipedia; then reuse existing task-specific fine-tuned expert models as "teacher models", guiding general PLMs ("student model") to learn task-specific prior knowledge through knowledge distillation. In this process, we perform multi-task learning with masked language modeling (MLM) and knowledge distillation objectives, which aims to avoid PLMs from over-fitting expert knowledge in the expert-guided pre-training stage, resulting in the weakening of general language modeling capabilities. We find that MLM loss, as a regularization term can prevent the expert-guided PLMs from converging to the local minima (Section 5.3). Finally, we fine-tune the expert-guided PLM without introducing additional annotated data, and experimental results prove that the performance of the new fine-tuned expert is generally better than the original expert model (Section 4.4), which means the goal of knowledge rekindle is achieved. We also experimented with a mixture-of-expert guided pre-training (MoEGP) strategy that leverages multiple expert models for multi-task knowledge distillation. Experimental results demonstrate that this method consistently improves compared to the single expert guided pre-training strategy (Section 5.1). We leave more details in the following Section 3.

Our contributions are three-fold: (1) We are the first to define "knowledge rekindle" as an improved paradigm of pre-training and fine-tuning, which re-incorporates the fine-tuned expert model into the training cycle and effectively overcomes the sub-optimal problem of simply fine-tuning PLMs using task-specific objectives. (2) We propose a unified expert-guided pre-training framework for knowledge rekindle, in which knowledge distillation helps PLMs to gain prior knowledge of downstream task and masked language modeling objective prevents the expert-guided PLMs from converging to local minima. (3) Extensive experiments and analyses demonstrate that our method has achieved significant improvements.

2 Related Work

2.1 Pre-trained Language models

Pre-trained language models have been widely used in various NLP tasks. Many researchers are exploring how to break through the performance upper bounds of fine-tuned expert models for specific tasks. One of the mainstream technical routes is to scale up PLMs, and studies have shown that scaling up the capacity of PLMs, training data size and increasing the number of training steps is helpful for improving the general language modeling capabilities of the pre-trained language models. The most representative work is GPT-3, ERNIE3.0 and etc.. GPT-3 is a revolutionary model, which contains 175 billion parameters, shows strong capabilities for language understanding and generation (Qin et al., 2021). InstructGPT (Ouyang et al., 2022) is a supervised fine-tuned version of GPT-3, which aims to align with the real requirements of human beings, and has demonstrated strong capabilities in many downstream tasks. However, training a large-scale pre-trained language model requires a lot of training resources and a very high training cost, which limits the wide application of large language models in the industrial-level applications, such as information retrieval and recommendation system.

Another line of methods (Gururangan et al., 2020; Wu et al., 2020; Liu et al., 2021b; Gao et al., 2021) propose domain-specific pre-training for some small-scale general language models such

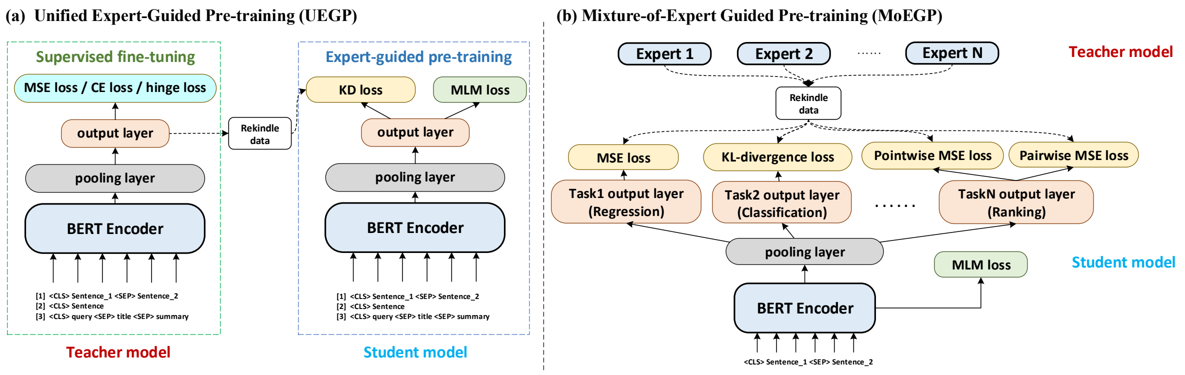


Figure 2: Illustration of our proposed unified expert-guided pre-training (UEGP) framerwork and mixture-of-expert guided pre-training (MoEGP) framework. We mainly discuss four types of NLU tasks: classification, semantic matching, textual entailment and ranking. Different types of tasks have different input and output formats, corresponding to different KD losses.

as BERT and ERNIE. Domain-specific pre-training can improve the performance of general PLMs on specific domains, so that better performance can be achieved after further fine-tuning. Currently, many industrial systems adopt domain-specific pre-training strategies in order to achieve optimal performance in specific scenarios. The expert-guided pre-training framework proposed in this paper is plug-and-play, and student models can be either general PLMs or domain-specific PLMs.

2.2 Knowledge Distillation

Initially knowledge distillation (KD) (Hinton et al., 2015) was designed to compress models for online deployment, and recently it has also been used as an important means of knowledge transfer. Researchers have explored to perform KD at different training stages, such as pre-trained models (Sanh et al., 2019), fine-tuned models (Krishna et al., 2019), and both (Jiao et al., 2019). They also explored different KD methods, such as distilling the output logits by teacher models (Sun et al., 2019), or distilling the intermediate hidden representations (Sun et al., 2020). Traditional KD usually distills the knowledge of a large-scale teacher model into a small-scale student model, aiming to match the student's performance to that of the teacher. However, quite a few recent studies have focused on two counter-intuitive settings: reversed-KD and defective-KD (Yuan et al., 2020). Reversed-KD selects a small-scale model with poor performance as a teacher model, a large-scale model with better performance as a student model, and defective-KD chooses a large-scale model with insufficient training as a teacher model. These two counter-intuitive settings give better results than fine-tuning the student model. Motivated by reversed-KD, Qin et al. (2021) proposed knowledge inheritance pre-training, which collects small-scale PLMs as knowledge sources, and adopts knowledge distillation method to train large-scale PLMs. The expert-guided pre-training framework proposed in this paper also adopts the KD objective, which aims to transfer task-specific prior knowledge into student PLMs so that fine-tuned student models can exceed the performance of teacher models and achieve knowledge rekindle.

3 Approach

3.1 Problem Formulation

In this paper, we are the first to propose the new paradigm "knowledge rekindle", which aims to reincorporate the expert model into the training cycle of pre-training and fine-tuning paradigm to further break through performance upper bounds. Next, we will briefly introduce the traditional pre-training and fine-tuning paradigm, and then dive into the definition of "knowledge rekindle".

Pre-training and Fine-tuning Paradigm. This is a one-way pipeline, as shown in fig 1(a): we first collect a amount of unlabeled general corpora D_u on Wikipedia or other public websites, and adopt the general language modeling objectives such as masked language modeling (MLM) (Devlin et al., 2019) or unidirectional language modeling (Radford et al., 2018) to train on D_u to obtain a well initialized general PLM θ_0. And then we fine-tune θ_0 on labeled data D_l to obtain an expert model θ_l for a specific downstream task.

Knowledge Rekindle Paradigm. This is a cyclical process, as shown in fig 1(b): Assuming that we have a general pre-trained language model θ_0 and labeled data D_l for specific tasks. We firstly perform supervised fine-tuning of θ_0 to obtain the expert model θ_l. In the traditional pre-training and fine-tuning paradigm, the training cycle of the expert model has ended and the performance of the expert model has converged. However, this is a sub-optimal model. We propose the knowledge rekindle paradigm to re-incorporate expert models into the training cycle to guide the general PLM θ_0 to learn prior knowledge of downstream tasks and obtain an expert-guided PLM θ_r. Finally, we use D_l again to fine-tune θ_r, and obtain a new task-specific expert model θ_{lr}. We hope to obtain the effect of $S(\theta_{lr}) > S(\theta_l)$. Here $S()$ represents the evaluation metrics for downstream tasks. Knowledge Rekindle is a paradigm to solve the sub-optimal problem of simply fine-tuning PLMs, and we will present more specific solutions next.

3.2 Overall Architecture

Fig 2 displays the overall architecture of our proposed unified expert-guided pre-training (UEGP) framework for knowledge rekindle. Since our work is currently mainly applied to industrial-level applications such as information retrieval and recommendation systems, we mainly discuss natural language understanding tasks, including classification, semantic matching, textual entailment and ranking. We choose the pre-trained language model with self-encoder architecture represented by BERT as the backbone.

In the expert-guided pre-training stage, we use general pre-trained BERT as the "student model", and task-specific fine-tuned expert model as the "teacher model". We inject task-specific prior knowledge of expert model into the student model through knowledge distillation. However, we find that expert-guided pre-training with only the knowledge distillation objective will overfit the knowledge of expert model to a certain extent, resulting in the weakening of the general language modeling capabilities of the PLM itself. Therefore, we retain the traditional self-supervised language modeling objective MLM in the expert-guided pre-training stage. On the one hand, it ensures that the expert-guided PLM does not lose general language modeling capabilities, and on the other hand, it regularizes the model to prevent over-fitting (He et al., 2022). We will further explain the harmonic effect between KD loss and MLM loss in section 5.3. Regarding the collection of Pre-training data ("rekindle data"), we will introduce it in section 3.3. It is worth noting that rekindle data is domain/task-agnostic, which also makes the collections of rekindle data very convenient, and our method can be easily applied to various domains and tasks. The general formula for expert-guided pre-training for knowledge rekindle is as follows:

$$\mathcal{L}_{UEGP} = \mathcal{L}_{KD} + \mathcal{L}_{MLM} \tag{1}$$

where $\mathcal{L}_{KD}$ means knowledge distillation objective, and $\mathcal{L}_{MLM}$ means general masked language modeling objective. Finally, we fine-tuned the expert-guided PLMs to break through the performance upper bounds of the expert models.

3.3 Rekindle Data

In our unified expert-guided pre-training framework, pre-training data can be unlabeled corpus from any source such as wikipedia or domain-specific databases. In the experiment, we selected English Wikipedia and Chinese user search logs from search engines as pre-training data, which is also called “rekindle data”. The former aligns with the data used by BERT in the general pre-training stage, and we hope to prove that the improvement comes from the student gaining the prior knowledge of teacher model by learning to imitate the behavior of the expert model, rather than by learning domain or task-related knowledge from extra data; the latter is to demonstrate the effectiveness of our method in industrial-level information retrieval scenarios. For more details about pre-training data, please refer to Appendix A.

3.4 Expert-Guided Pre-training

For different tasks, we need to consider different input-output formats, different fine-tuning losses, and different knowledge distillation losses. In this section, we take the semantic matching task as an example. For more details about other tasks, please refer to Appendix D.

UEGP for semantic matching task. Semantic matching is a basic task in natural language understanding and is widely used in application scenarios such as information retrieval, recommendation and question answering systems. The semantic matching task aims to score and evaluate the similarity of two given sentences, and the model output is usually a floating point number range from 0 to 5, the higher the score means the more similar the two sentences are. The input format of the semantic matching task is usually to concatenate two sentences, connect them with a special <SEP> token, and add a special token <CLS> in front of the input text. In the output layer, we will take the embedding of the <CLS> token and forward it to a linear layer. The range of output value will be limited to 0-1 through the sigmoid (Finney, 1947) activation function, and then enlarged according to the range of the ground-truth labels. For objective functions, we use mean square error (MSE) loss for task-specific fine-tuning, and correspondingly, we also use the similar MSE loss as distillation loss for expert-guided pre-training. In addition, we perform multi-task learning with both KD loss and masked language modeling loss. In a word, the objective function of UEGP for semantic mathcing task is as follows:

$$\mathcal{L}_{KD\_MSE} = \frac{1}{|D_r|} \sum_{i=1}^{|D_r|} (\hat{y}_i - y_i)^2 \tag{2}$$

$$\mathcal{L}_{UEGP} = \mathcal{L}_{KD\_MSE} + \mathcal{L}_{MLM} \tag{3}$$

where D_r is the rekindle dataset, $\hat{y}_i$ is the semantic similarity score predicted by the student model, and y_i is the semantic similarity score predicted by the teacher model.

3.5 Mixture-of-Expert Guided Pre-training

In the previous discussion, we performed expert-guided pre-training on each task individually for knowledge rekindle. However, there are two problems with this training strategy: (1) When we only use the expert model of a single task for expert-guided pre-training, the expert-guided PLM can only achieve knowledge rekindle on a single task. We hope it will further benefit from more expert models. (2) When we need to process multiple downstream tasks, we need to perform knowledge rekindle for each task separately, and the training cost will increase exponentially.

In order to reduce the training cost and obtain a more powerful expert-guided PLMs, we extend the expert-guided pre-training framework and propose mixture-of-expert guided pre-training framework. Firstly, we perform supervised fine-tuning for each downstream task to obtain N task-specific expert models. Then we use these N expert models (teacher models) to perform task-specific inference on the collected unlabeled pre-training corpus respectively, and store the output logits. Next, we add N different output layers on BERT (student model) for N different tasks, and align the output logits of each output layer with the output of the corresponding expert model through the knowledge distillation objective. N different tasks have N different style of knowledge distillation losses, and we jointly optimize different loss functions. We still combine knowledge distillation and MLM objectives for joint optimization. The formula for mixture-of-expert guided pre-training is as follows:

$$\mathcal{L}_{MoEGP} = \sum_{i}^{N} \mathcal{L}_{KD}^{i} + \mathcal{L}_{MLM} \tag{4}$$

3.6 Compared with Continuous Pre-training

The further pre-training mentioned in (Gururangan et al., 2020) requires the collection of domain-specific or task-specific pre-training corpus, but in our knowledge rekindle setting, there is no need to especially collect domain-specific or task-specific corpus for expert-guided pre-training. In addition, for PLMs that are obtained from domain pre-training or task pre-training, we can also use the same method for knowledge rekindle.

In short, our method can improve the performance of fine-tuned expert models without introducing additional data and exploit the performance upper bounds of PLMs. This is the most important difference with further pre-training. Besides, our method is compatible with any PLMs and can be used sequentially.

4 Experiments

4.1 Datasets

We mainly conducted experiments on General Language Understanding Evaluation (GLUE) benchmark. GLUE covers a diverse range of NLP tasks, including classification (CoLA, SST-2), semantic matching (STS-B, MRPC, QQP) and textual entailment (QNLI, MNLI, RTE).

In addition, we also verified that knowledge rekindle is also applicable on a larger-scale industrial-level search re-ranking dataset (RE-RANK). For RE-RANK dataset, queries and documents are collected from the Chinese search engines and manually labeled on the crowd-sourcing platform, where a group of hired annotators assigned an integer label range from 0 to 4 to each query-document pair, representing their semantic relevance as $\{bad, fair, good, excellent, perfect\}$. We leave the detailed statistical information to Appendix A.

4.2 Baselines

In this work, we mainly compare our proposed knowledge rekindle paradigm with traditional pre-training and fine-tuning paradigm. Here, for the tasks on the GLUE benchmark, we chose the English BERT model with 12-layers and 24-layers as the PLMs backbone, and for the industrial-level Chinese search re-ranking dataset, we chose the Chinese ERNIE model with 12-layers and 48-layers as the PLMs backbone. We fine-tune these PLMs individually on the labeled dataset for each task as baselines. For our proposed knowledge rekindle paradigm, BERT and ERNIE with 12-layers are used as "teacher models" in our standard setting, and the general pre-trained BERT and ERNIE with different sizes as the "student model".

4.3 Evaluation Metrics

We adopt several widely used metrics to evaluate the performance of the fine-tuned expert model before and after knowledge rekindle: For STS-B, we choose spearman correlation coefficient as evaluation metric; for CoLA, we choose matthews correlation coefficient as evaluation metric; for SST-2, QNLI, MNLI, QQP, RTE and MRPC, accuracy is used as evaluation metric; for RE-RANK, we use positive-negative ratio (PNR) for evaluation.

4.4 Main Results

We validate the effectiveness and universality of knowledge rekindle paradigm on the GLUE benchmark. Table 1 shows the main results of our knowledge rekindle paradigm compared to traditional pre-training and fine-tuning baselines. We use the task-specific fine-tuned BERT-base model (BERT-base-FT) as "teacher model", and BERT-base and BERT-large as "student model" respectively for expert-guided pre-training. Finally, we fine-tune the expert-guided PLM to achieve knowledge rekindle. The experimental results show that knowledge rekindle paradigm significantly outperforms traditional pre-training and fine-tuning baselines on almost all 8 NLU tasks. Next, we analyze the results from two aspects:

(1) **The improvements are more significant when the student model capacity increases.** For example, on the STS-B dataset, UEGP-BERT-base-FT has improved by 0.57 compared to BERT-base-FT, UEGP-BERT-large-FT has improved by 1.24 compared to its teacher model BERT-base-FT, and is also superior to BERT-large-FT with the same capacity by 0.87. On the MRPC dataset, UEGP-BERT-base-FT has improved by 1.27 compared to BERT-base-FT, UEGP-BERT-large-FT has improved by 1.93 compared to its teacher model BERT-base-FT, and also outperform BERT-large-FT by 1.62. We argue that as the size of PLMs increases, the performance upper bounds on downstream tasks also increase. However, simply fine-tuning PLMs with task-specific objectives often leads to convergence to local minima, resulting in sub-optimal performance. The expert-guided pre-training framework can effectively break through the performance upper bounds of fine-tuned expert models without introducing additional labeling costs.

(2) **For data scarcity scenario, the knowledge rekindle paradigm improves more significantly.** For example, for the SST-2, QQP, and QNLI datasets, traditional pre-training and fine-tuning paradigm has achieved superior performance (accuracy over 90%), but for the STS-B, CoLA, and RTE datasets, the performance of baseline methods is relatively poor. We find that the amount of labeled data in STS-B, RTE, and CoLA is relatively scarce, which may be responsible for the poor performance of the pre-training and fine-tuning paradigms. Fine-tuning PLMs on scarce labeled data makes it easier to converge to local minima. Interestingly, we find that the performance improvement on the STS-B, CoLA, and RTE datasets is the most significant, which suggests that the knowledge rekindle paradigm is beneficial to improve capabilities of PLMs on data scarcity scenarios.

In addition, we also verify the effectiveness of the knowledge rekindle in industrial-level applications. We conduct experiments on an industrial-level search re-ranking dataset(RE-RANK). Specifically, we select task-specific fine-tuned ERNIE-12layers(ERNIE-12layer-FT) as the teacher model and ERNIE-48layers as the student model, perform expert-guided pre-training, and then perform fine-tuning to obtain a new expert model. We fine-tune the checkpoints obtained from different pre-training steps, and the experimental results are shown in table 2. The experimental results show that the knowledge rekindle paradigm is consistently better than traditional pre-training and fine-tuning baselines. We also find that as pre-training steps gradually increase, the performances of fine-tuned models are also gradually improved, but the improvement is not significant. Fewer pre-training steps mean using fewer pre-training data, which

Method	STS-B spearman corr.	CoLA matthews corr.	SST-2 accuracy	QQP accuracy	QNLI accuracy	MNLI accuracy	RTE accuracy	MRPC accuracy	Avg
BERT-base-FT(teacher)	89.34	56.23	92.08	90.61	91.10	83.54	68.59	84.17	81.96
BERT-large-FT	89.71	59.79	93.11	**91.19**	91.74	86.16	71.11	84.48	83.41
UEGP-BERT-base-FT(ours)	89.91	59.25	92.31	91.03	91.03	83.58	65.70	85.44	82.28
UEGP-BERT-large-FT(ours)	**90.58**	**62.02**	**93.46**	91.18	**92.54**	**86.38**	**73.28**	**86.10**	**84.50**

Table 1: Performance comparison on eight GLUE tasks (dev set). We use the fine-tuned BERT-base as the teacher model, and the pre-trained BERT-base and BERT-large as student models. After expert-guided pre-training and further fine-tuning for specific tasks, UEGP-BERT-base-FT and UEGP-BERT-large-FT are obtained, respectively. Results are averaged over three random runs. ($p < 0.01$ under t-test)

Models	RE-RANK(eval)	RE-RANK(test)
ERNIE-12layer-FT(teacher)	3.457	3.328
ERNIR-48layer-FT	3.526	3.367
UEGP-ERNIE-48layer-FT(30k)	3.568	3.463
UEGP-ERNIE-48layer-FT(830k)	3.573	3.470
UEGP-ERNIE-48layer-FT(990k)	**3.584**	**3.498**

Table 2: Performance comparison on industrial-level search re-ranking task.

indicates that we can achieve lightweight expert-guided pre-training in practical applications, reducing training costs while maintaining performance.

5 Qualitative Analysis

5.1 MoE guided knowledge rekindle

We further extended the expert-guided pre-training framework and explored the feasibility of the mixture-of-expert guided pre-training. Specifically, we need to simultaneously perform knowledge distillation for the teacher models of 8 tasks on the GLUE benchmark in the expert-guided pre-training stage, and inject the expert knowledge of these 8 tasks into the student model. Here we still choose task-specific fine-tuned BERT-base (BERT-base-FT) as the "teacher model", and BERT-large as the "student model". Table 3 shows the comparison results of MoE guided pre-training and expert-guided pre-training. We can see that the former has achieved better or equal performances on the GLUE benchmark. We believe that the MoE guided pre-training distills the knowledge of multiple task-specific teacher models, and the knowledge for multiple tasks can complement each other, which helps to improve the performance of PLMs on specific tasks.

5.2 The effect of model size

Next, we will further discuss the impact of the sizes of teacher models and student models on the knowledge rekindle paradigm, respectively. We take QNLI and SST-2 as examples for experimental verification, and the results are shown in Table 4. Specifically, we compare four sets of teacher-student combinations[2] and find that the general trend is that the larger the size of teacher models, the greater the size of student models, and the more significant the performance of the knowledge rekindle paradigm. We believe that a larger teacher model means that the teacher model itself contains richer knowledge, and a larger student model indicates that the upper bound that the student model can reach is higher.

5.3 Explanation of the interaction between KD and MLM

In order to further explore why the interaction between knowledge distillation and MLM objectives helps to achieve knowledge rekindle, we analyzed it from two perspectives:

Ablation Study We first perform ablation analysis, and the results are shown in Table 5. Specifically, we use the fine-tuned BERT-base (BERT-base-FT) as the teacher model, and the pre-trained BERT-large as the student model. For UEGP-BERT-large-FT(KD+MLM), we adopt the multi-task learning objective of knowledge distillation and masked language modeling in the expert-guided pre-training stage; for UEGP-BERT-large-FT(KD), we only use the KD objective for expert-guided pre-training; for UEGP-BERT-large-FT(MLM), we just perform MLM on rekindle data, which is to explore whether the improvement of knowledge rekindle strategy comes from the guidance of expert knowledge, or from the introduction of additional unlabeled data. The experimental results show that the knowledge distillation objective enables PLMs to obtain prior knowledge for downstream tasks and promotes the performance improvement after further fine-tuning. In addition, adding MLM objective for multi-task learning can further improve the performance.

[2] 4-12 means that the size of teacher model is 4 layers and the size of student model is 12 layers.

Method	STS-B spearman corr.	CoLA matthews corr.	SST-2 accuracy	QQP accuracy	QNLI accuracy	MNLI accuracy	RTE accuracy	MRPC accuracy	Avg
BERT-base-FT(teacher)	89.34	56.23	92.08	90.61	91.10	83.54	68.59	84.17	81.96
BERT-large-FT	89.71	59.79	93.11	91.19	91.74	86.16	71.11	84.48	83.41
UEGP-BERT-large-FT	90.58	62.02	93.46	91.18	**92.54**	**86.38**	73.28	86.10	84.50
MoEGP-BERT-large-FT(step=58k)	90.34	**63.73**	92.88	91.24	92.44	85.59	74.36	85.73	84.54
MoEGP-BERT-large-FT(step=88k)	90.56	62.14	92.66	91.22	92.03	85.57	75.81	86.31	84.53
MoEGP-BERT-large-FT(step=128k)	**90.80**	63.11	**93.92**	**91.32**	92.11	86.28	**75.81**	**87.13**	**85.06**

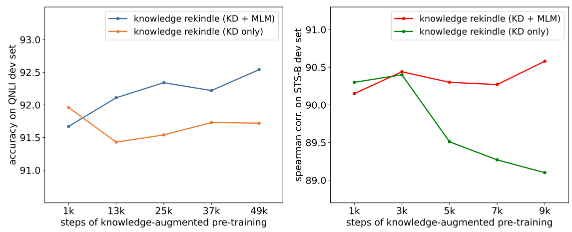
Table 3: Performance comparison of mixture-of-expert guided pre-training (MoEGP) and expert-guided pre-training for knowledge rekindle on eight GLUE tasks (dev set). We adopt eight task-specific fine-tuned BERT-base models as teacher models, and pre-trained BERT-large model as the student model. After expert-guided pre-training and fine-tuning, MoEGP-BERT-large-FT is obtained. We select checkpoints from three different pre-training steps and report their performances. Results are averaged over three random runs. ($p < 0.01$ under t-test)

Models	QNLI	SST-2	RTE	CoLA
BERT-tiny-FT(teacher)	82.29	87.15	-	-
BERT-base-FT(teacher)	91.10	92.08	68.59	56.23
BERT-large-FT(teacher)	91.74	93.11	71.11	59.79
UEGP-BERT-base-FT(4-12)	90.53	91.97	-	-
UEGP-BERT-base-FT(12-12)	91.03	92.31	65.70	59.25
UEGP-BERT-large-FT(12-24)	**92.54**	93.92	73.28	62.02
UEGP-BERT-large-FT(24-24)	92.10	**94.15**	**74.36**	**62.14**

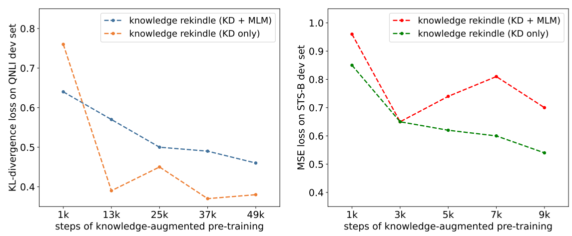
Table 4: The effect of model sizes of different teacher models and student models on knowledge rekindle. Among them, BERT-tiny, BERT-base, and BERT-large represent PLM capacity of 4 layers, 12 layers, and 24 layers respectively.

Models	STS-B	QNLI	MSRP
BERT-base-FT(teacher)	89.34	91.10	84.17
BERT-large-FT	89.71	91.74	84.48
UEGP-BERT-large-FT(KD+MLM)	**90.58**	**92.54**	**86.10**
UEGP-BERT-large-FT(KD)	90.30	91.90	85.97
UEGP-BERT-large-FT(MLM)	89.65	91.59	83.93

Table 5: Ablation study of KD and MLM objectives.



(a) The performance trend of fine-tuned expert models



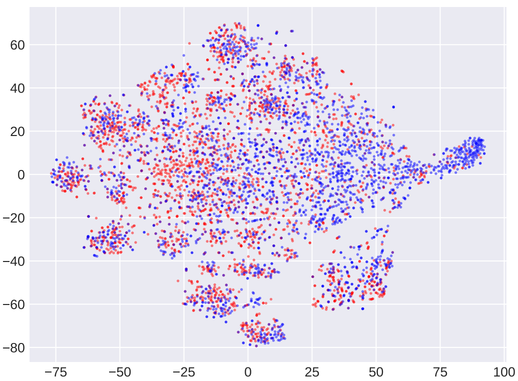
(b) The changes in task-specific loss value during expert-guided pre-training

Figure 3: The convergence of PLMs on downstream tasks during the expert-guided pre-training phase.

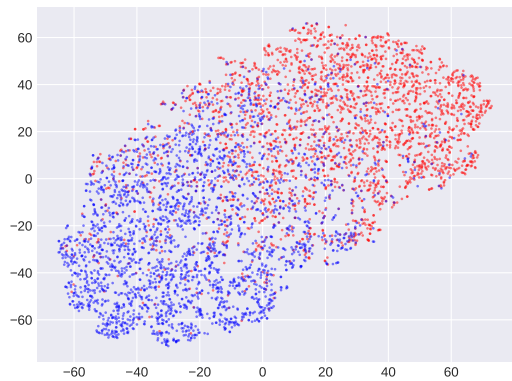
Observe the convergence during UEGP In order to gain a deeper understanding of why expert-guided pre-training can help improve the performance of fine-tuned expert models, we analyzed the convergence of PLMs on downstream tasks during the expert-guided pre-training phase and the results are shown in Figure 3. We can see that when we adopt multi-task learning with KD and MLM objective, as the expert-guided pre-training steps increase, the performance of the task-specific fine-tuned PLMs gradually increases. However, when we only use the KD objective, the performance of task-specific fine-tuned PLMs shows a downward trend as the number of training steps increases (see Figure 3(a)).

To explain this phenomenon, we analyze the changes in task-specific loss value during expert-guided pre-training process (see Figure 3(b)). We observe that, compared with the KD only method, the expert-guided PLMs trained by the KD and MLM combination objectives have relatively higher loss values on downstream tasks, and the convergence speed is relatively slow. We believe that the pre-training method with only KD objective easily causes the model to overfit the knowledge of the expert model in the expert-guided pre-training stage, so that the task-specific loss converges to the local minima in advanced, resulting in a sub-optimal results. The MLM objective, as a regularization term, can effectively prevent the expert-guided PLMs from overfitting expert models, slow down the occurrence of local minima, and ensure that task-specific fine-tuning can further improve performance.

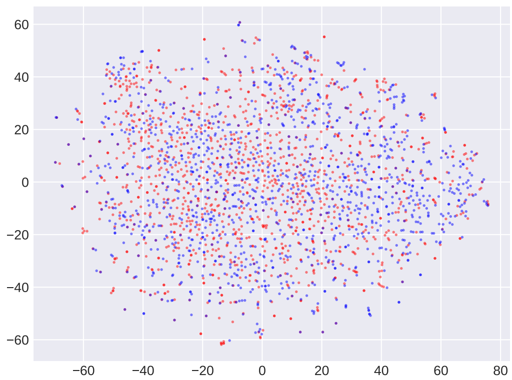
From this analysis, we can also explain why our proposed unified expert-guided pre-training framework for knowledge rekindle can effectively improve the performance upper bounds of fine-tuned expert models: On the one hand, the knowledge



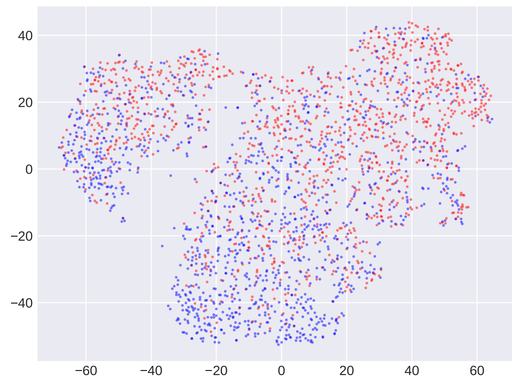
(a) QNLI general PLM



(b) QNLI expert-guided PLM



(c) RTE general PLM



(d) RTE expert-guided PLM

Figure 4: visualization for the performance of expert-guided PLMs and general PLMs on QNLI and RTE datasets. We use t-SNE (Van der Maaten and Hinton, 2008) to achieve dimensionality reduction

distillation objective enables the PLM to learn the prior knowledge of downstream tasks; on the other hand, the MLM objective, as a regularization term, effectively alleviates the over-fitting of PLMs to expert models and avoids the models from converging to local minima in advanced.

5.4 Visulization

In order to compare knowledge rekindle with traditional pre-trainng and fine-tuning paradigm more intuitively. We take two tasks of QNLI and RTE as examples to visualize the performance of expert-guided PLMs and general PLMs, see Figure 4. We did not perform task-specific fine-tuning. Since QNLI and RTE are both text entailment tasks, We can see that sentence-level representations obtained by general PLMs form a disorderly distribution and samples of different categories mix together. In contrast, expert-guided PLMs can form more discriminative distributions of different categories in the representation space, which indicates UEGP enables PLMs to learn the prior knowledge of expert models and have better initial parameter states, thereby further improving their performance after fine-tuning.

6 Conclusion

In this paper, we first propose knowledge rekindle paradigm as an improved paradigm of pre-training and fine-tuning, which aims to re-incorporate the fine-tuned expert model into the training cycle to further break through the performance upper bounds. We further propose a unified expert-guided pre-training method for knowledge rekindle, which adopts the combined objectives of knowledge distillation and masked language modeling. On the one hand, it enables PLMs to learn prior knowledge of downstream tasks, and on the other hand, it can avoid the model from converging to local minima in advance. In short, our method can exploit the performance upper bounds of PLMs without introducing additional data, and it is compatible with any PLMs and can be used sequentially. Extensive experiments on the GLUE benchmark and industrial search re-ranking dataset demonstrate the effectiveness of our method.

Limitations

This paper mainly focuses on the limitation with the traditional pre-training and fine-tuning paradigm: simply fine-tuning with task-specific objectives often converges to local minimum, leading to sub-optimal performance. Our proposed knowledge rekindle paradigm and unified expert-guided pre-training framework (UEGP) re-incorporate the fine-tuned expert model into the training cycle and break through the performance upper bounds of experts without introducing additional annotated data. However, our work also have several limitations: (1) Since our method is currently mainly used in application scenarios such as information retrieval and recommendation systems, we only conducted experiments on natural language understanding tasks. In the future, we will try to apply our method to generative models such as GPT and T5. (2) We mainly verify the effectiveness of knowledge rekindle as an improved paradigm of traditional pre-training and fine-tuning on the GLUE benchmark. Actually, our method is plug-and-play, and in the future we can also try to apply our method on conversation understanding, or other domain-specific PLMs such as TOD-BERT, FinBERT and etc..

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lianshang Cai, Linhao Zhang, Dehong Ma, Jun Fan, Daiting Shi, Yi Wu, Zhicong Cheng, Simiu Gu, and Dawei Yin. 2022. PILE: Pairwise iterative logits ensemble for multi-teacher labeled distillation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 587–595, Abu Dhabi, UAE. Association for Computational Linguistics.

David Cossock and Tong Zhang. 2006. Subset ranking using regression. In *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006. Proceedings 19*, pages 605–619. Springer.

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

David John Finney. 1947. Probit analysis; a statistical treatment of the sigmoid response curve.

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*.

Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021a. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3365–3375.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021b. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

Shie Mannor, Dori Peleg, and Reuven Rubinstein. 2005. The cross entropy method for classification. In *Proceedings of the 22nd international conference on Machine learning*, pages 561–568.

Leonid Margolin. 2005. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134:201–214.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

Yujia Qin, Yankai Lin, Jing Yi, Jiajie Zhang, Xu Han, Zhengyan Zhang, Yusheng Su, Zhiyuan Liu, Peng Li, Maosong Sun, et al. 2021. Knowledge inheritance for pre-trained language models. *arXiv preprint arXiv:2105.13880*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Siqi Sun, Zhe Gan, Yu Cheng, Yuwei Fang, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. *arXiv preprint arXiv:2009.14167*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. *arXiv preprint arXiv:2004.06871*.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4321–4330.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911.

Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294.

Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4014–4022.

A Details about datasets

Pre-training Data. In the expert-guided pre-training stage, we firstly need to collect some unlabeled corpus. Unlike TOD-BERT (Wu et al., 2020) and SimCSE (Gao et al., 2021), which need to specially collect domain/task-specific unlabeled pre-training data, our method is task-agnostic and we only need to collect corpus from public sources such as Wikipedia at a low cost. In this work, we collect a large amount of documents from Wikipedia. We pair the sentences in each document into sentence pairs. After deduplication and quality filtering, we obtain a total of 2 million sentence pairs as pre-training data for expert-guided pre-training. We name the task-agnostic rekindle dataset as "rekindle-NLU-EN", which is used for knowledge rekindle on 8 general English NLU tasks of GLUE benchmark. In addition, we also collected 90 million query-document pairs from Chinese search engines as rekindle data for industrial-level Chinese search re-ranking task. We name this rekindle dataset as "rekindle-NLU-CN". The detailed statistical information of rekindle data is shown in Table 6.

Evaluation Task and Data. The detailed statistical information of the 9 evaluation tasks and datasets are shown in Table 7.

Statistic	rekindle-NLU-EN	rekindle-NLU-CN
Number of samples	2,000,000	92,160,000
Max / Avg utterance length	1,721 / 56.01	1,325 / 122.17
Vocabulary size	27,345	48,345
Language	English	Chinese

Table 6: Statistics of two rekindle dataset.

Dataset	Training	Validation	Test	Vocabulary	Length (max / mean)
STS-B	5,749	1,379	1,377	10,794	125 / 27.81
CoLA	8,551	1,043	1,063	5,586	47 / 11.32
SST-2	67,350	873	1,821	11,572	66 / 13.31
QNLI	104,743	5,463	5,461	26,239	550 / 49.54
MNLI	392,702	9,815	9,796	25,648	330 / 30.58
QQP	363,870	40,431	390,965	25,821	444 / 39.91
MRPC	4,076	1,725	-	12,063	103 / 53.24
RTE	2,491	277	3,000	13854	289 / 70.19
RE-RANK	11,085,989	84,722	320,317	48,914	141,007 / 123.93

Table 7: Statistics of 9 labeled datasets for specific tasks

B Implementation

For a fair comparison of various methods, we use the general pre-trained BERT [3] (bert-base-uncased with 12-layer, and bert-large-uncased with 24-layer transformer) as our network backbone. For the traditional pre-training and fine-tuning paradigm, we fine-tune all parameters of BERT model using task-specific labeled datasets. The specific hyper-parameter settings are shown in Table 8. For the knowledge rekindle paradigm, the training process is divided into two stages: expert-guided pre-training and task-specific fine-tuning. In the expert-guided pre-training stage, we use the task-specific fine-tuned model as the teacher model, and the pre-trained language model BERT before fine-tuning as the student model. We perform multi-task learning with knowledge distillation and masked language modeling objectives for training. Specifically, we set the learning rate to 5e-5 and the batch size to 200, and we use Adam (Kingma and Ba, 2015) as the optimizer with linear warm-up and polynomial decay (warmup steps = 1000, lr_{min} = 0 and lr_{max} = 5e-5) In the task-specific fine-tuning phase, we adopted the same hyper-parameter setting as Table 8, which aims to eliminate the interference of other factors besides the expert-guided pre-training itself, and more accurately verify the effectiveness of our method. All experiments are done in the same computation environment with 8 NVIDIA 40GB A100 GPUs. As for the specific computational overhead metrics, taking the BERT-12layers as an example, the batch size of the expert-guided pre-training phase is set to 200, which is about 5min/1000 iterations. For ERNIE-48layer, the batch size of the expert-guided pre-training phase is set to 100, which is approximately 10.4min/1000 iterations.

[3]https://github.com/google-research/bert

C Positive-Negative Ratio (PNR) matric

The Positive-Negative Ratio (PNR) measures the consistency between the golden labels and the scores output by models (Cai et al., 2022). For a given query q and a list of N associated documents ranked by model, the PNR can be calculated by this formulation:

$$PNR = \frac{\sum_{i,j\in[1,N]} I\left\{y_i > y_j\right\} I\left\{f\left(q, d_i\right) > f\left(q, d_j\right)\right\}}{\sum_{i,j\in[1,N]} I\left\{y_i > y_j\right\} I\left\{f\left(q, d_i\right) < f\left(q, d_j\right)\right\}}, \tag{5}$$

where I is the indicator function, taking the value 1 if the internal statement is true or 0 otherwise.

D Details of UEGP on different task

UEGP for classification task. In natural language understanding, classification is the most common task, including intent recognition, sentiment classification and so on. In the GLUE benchmark, CoLA and SST-2 are two representative text classification datasets. For classification tasks, we usually concatenate a special token <CLS> in front of the input text, and at the output layer we will take the embedding of the <CLS> token and classify it through a softmax layer. For objective functions, we use cross-entropy (CE) loss as the objective function for fine-tuning. Since classifier essentially outputs a posterior probability distribution, we use KL-divergence loss as the knowledge distillation loss to let the student model simulate the output probability distribution of the teacher model. The objective function of expert-guided pre-training for classification task is as follows:

$$\begin{aligned}\mathcal{L}_{KD\_KL} &= \frac{1}{|D_r|}\sum_{i=1}^{|D_r|} KL(y_i||\hat{y}_i) \\ &= \frac{1}{|D_r|}\sum_{i=1}^{|D_r|}\sum_{c=1}^{C} y_i^c \log\frac{y_i^c}{\hat{y}_i^c}\end{aligned} \tag{6}$$

$$\mathcal{L}_{UEGP} = \mathcal{L}_{KD\_KL} + \mathcal{L}_{MLM} \tag{7}$$

where $\hat{y}_i$ is the output logits predicted by the student model, and y_i is the logits predicted by the teacher model.

UEGP for textual entailment task. Text entailment (also known as natural language inference) aims at given a premise sentence and a hypothesis sentence, we need to predict whether the premise

Datasets	BERT-base(12 layers)			BERT-large(24 layers)		
	epoch	learning rate	batch size	epoch	learning rate	batch size
STS-B	4	1e-4	64	4	5e-5	64
CoLA	3	3e-5	64	5	3e-5	32
SST-2	4	2e-5	256	4	2e-5	64
QNLI	4	1e-5	256	4	2e-5	256
MNLI	3	3e-5	256	3	3e-5	256
QQP	3	5e-5	256	4	2e-5	64
MRPC	4	3e-5	32	4	3e-5	64
RTE	4	2e-5	64	5	2e-5	64

Table 8: The hyper-parameter settings for English GLUE datasets.

sentence contains the hypothesis, contradicts with the hypothesis or neither. That is, we need to classify the sentence pairs containing premise and hypothesis into three categories: entailment, contradiction or neutral. In the GLUE benchmark, QNLI, RTE and MNLI are three widely used evaluation datasets for text entailment task. Similar to the semantic matching task, the input format of the text entailment task is also to concatenate two sentences, and connect them with a special <SEP> token. Text entailment is formally a sentence-pair classification task, so the cross-entropy loss is used in the fine-tuning stage, and the KL-divergence loss is used as the distillation loss in the expert-guided pre-training stage.

UEGP for ranking task. In addition to semantic matching, classification and textual entailment, ranking is also a very important natural language understanding task, and plays a pivotal role in information retrieval and recommendation system. In search engines, a common ranking scenario is to rank query-document pairs based on semantic relevance. For a query, we will retrieve K related documents, and then pair the query with the documents, that is, <query, doc1> <query, doc2>...<query, docN>. In the search re-ranking task, we hope that the model will score each query and document pair based on semantic relevance, and the relative order of the scores is proportional to the relevance degree of the query and documents.

The input form of the search re-ranking task is to concatenate the query, the title of the document and the summary of the document, and each part is separated with a special token <SEP>. The output form is similar to that of semantic matching task, which is a floating-point score range from 0 to 5. However, since the semantic matching task only needs to consider the similarity between the two input sentences, but the search re-ranking task needs to consider the relative order between different documents. Thus, we use hinge loss for task-specific fine-tuning. In the expert-guided pre-training stage, in order to better align with fine-tuning objectives, we adopt pointwise (Cossock and Zhang, 2006) and pairwise (Zheng et al., 2007) MSE loss as the knowledge distillation loss. Likewise, we add a masked language modeling loss for multi-task learning.

$$\mathcal{L}_{pointwise\_mse} = \sum_{i}^{|Q|} \sum_{j}^{|D_i|} \left(\hat{y}_i^{d_j} - y_i^{d_j} \right)^2 \tag{8}$$

$$\mathcal{L}_{pairwise\_mse} = \sum_{i}^{|Q|} \sum_{\substack{j,k \\ j \neq k}}^{|D_i|} \left[\left(\hat{y}_i^{d_j} - \hat{y}_i^{d_k} \right) - \left(y_i^{d_j} - y_i^{d_k} \right) \right]^2 . \tag{9}$$

where Q is the query set, and D_i is the document set retrieved by the i-th query.