Fast Adaptation via Prompted Data: An Efficient Cross-Domain Fine-tuning Method for Large Language Models

Yiming Zhang*, Hantao Yang*, Haobo Wang, Junbo Zhao
Zhejiang University, Hangzhou Zhejiang, China
{yimingz, ht.yang, wanghaobo, j.zhao}@zju.edu.cn

Abstract

Large language models (LLMs) have achieved great success in a variety of natural language understanding tasks. However, domain discrepancies between the downstream task and the pre-training corpora may have hurdled LLMs to excel further in the vertical applications. Contrary to prior computational-heavy methods, we propose a lightweight solution to further bridge the gap in applying LLMs to diverse downstream tasks — a Fast Adaptation method for LLMs via Prompted Data, in short FAvPD. Notably, with FAvPD, we establish an additional adaptive tuning procedure, wherein we integrate downstream text corpora, gold labels as well as external knowledge sources and then envelop them into a form of highly controllable prompt. As a simple, easy-to-use, and versatile solution, FAvPD lies in the intersection of regimes like knowledge-augmented LLMs, fine-tuning, and adaptation techniques. With extensive experiments, we prove that FAvPD excels in both performance efficacy and training efficiency over related prior works. FAvPD is publicly available at https://github.com/Hyatio/FAvPD.

Keywords: Large Language Model, Knowledge Injection, Prompt

1. Introduction

The very notable emergence of the large language models (LLMs[1]) — such as BERT (Devlin et al., 2019), ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023) etc. — have greatly altered the terrain of natural language processing. By applying self-supervised learning on large-scale unlabeled corpora, LLMs are proven to capture rich lexical (Jawahar et al., 2019), syntactic (Hewitt and Manning, 2019) and semantic information that significantly benefits numerous downstream tasks. Simultaneously, many research studies have leveraged the strong language capabilities of these Language Models (LLMs) to address downstream tasks by employing fine-tuning methods (Hu et al., 2022a; Li and Liang, 2021). The paradigms based on pre-training and fine-tuning continue to be widely accepted as the standard workflow in various natural language understanding domains (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2018a).

Although LLMs have achieved great success in the most NLP fields, there still some works (JI et al., 2022; Bang et al., 2023) have also pointed out that when lacking domain-specific knowledge, LLMs are more prone to hallucinate in downstream tasks. Indeed, throughout the literature, the available methodological training-based techniques validated by the community may have included: knowledge injection (Sun et al., 2020b; Wang et al., 2021b; Meng et al., 2022), continued pre-training (Gururangan et al., 2020; Qin et al., 2023). Despite the merits, these lines' approaches often utilize distinct sources of information. For instance, the most common fine-tuning-based methods are widely applied for their lightweight nature of deployment, and they facilitate the additional tuning on a small portion of the ⟨**label**⟩ from the downstream dataset. Meanwhile, the knowledge injection scheme is instead dedicated to merging an external ⟨**knowledge**⟩ library or graph through manipulation in the representation space. And finally, the domain-adaptation methods are concerned with bringing the LLMs closer to the distribution drawn from the downstream ⟨**text**⟩. Against this background, our work explores how to uniformly apply the above various external information to continue training pre-trained language models (including large language models).

The major contribution of this work can be seen as two-fold. On one hand, we provide a unified scheme to feed a pre-trained LLM with downstream **text**, associated **labels**, and linked **knowledge** graph synergistically. Inspired by fields such as prompt-tuning (Ding et al., 2021) and in-context learning (Dong et al., 2022), we constructed a structure more suitable for training language models and proposed a knowledge reconstruction loss function to assist the model learning from examples. On the other, we provide an extremely lightning adaptation workflow upon the above data unification where we embed an additional adaptive tuning stage in between the pre-training and fine-tuning stages. Through empirical verification, we find that this additional stage of tuning requires fairly little data but sharply renders meritable performance gain. Our approach is dubbed as **F**ast **A**daptation **v**ia

* means Equal Contributors

[1] LLMs are commonly referred to as pre-trained language models (PLMs).

Prompted **D**ata (**FAvPD**). We provide dense experimental results to prove: (i)-FAvPD's extreme training efficiency and its advantageous efficacy; (ii)-FAvPD is highly versatile and can be easily combined with most recent white-box LLMs such as LLaMA (Touvron et al., 2023) and (iii)-FAvPD works with many other workflows such as LoRA (Hu et al., 2022a).

Let us give a concrete example. When utilizing an LLM for a downstream medical NER task, the standardized workflow usually is conducted by fine-tuning the pre-trained LLM directly by the gold text-label pair in a supervised manner. Despite that, FAvPD makes changes in two aspects. FAvPD exclusively establishes another round of the data-efficient tuning procedure, before the task-wise fine-tuning launches. This procedure manifests FAvPD's highly lightweight tuning nature because it only requires a draft of a few data points from the downstream corpora. Besides the gold labels exploited, FAvPD additionally compacts an external knowledge graph and the gold text altogether as a synergistic system.

We validate FavPD by extensive experiments. In terms of efficacy, our method outperforms most of the previous work and achieves state-of-the-art results on Open Entity and FIGER while exceeding the rivals by up to **10.1%** by F1. Furthermore, the combination of FAvPD and the PEFT method of LLMs can significantly improve the accuracy of domain question answering tasks. For training efficiency, FAvPD manifests in an extremely lightweight fashion. Notably, the corpora size and the training time consumption are roughly less than **1%** of the general knowledge injection framework. An exemplary code is provided as a part of the supplementary material.

2. Related Work

2.1. Knowledge-Enhanced LLMs

Knowledge-Enhanced LLMs mean the enhance the performance of LLM by Knowledge Injection method (Zhang et al., 2019). This method is a well-known practical approach to solving the lack of factual knowledge caused by domain discrepancies and aims to integrate entity information, relation triplets, or knowledge graph into the language model and further helps LLMs improve downstream tasks' performance (Peters et al., 2019; Wang et al., 2021c; Liu et al., 2020a; Yamada et al., 2020; Zhang et al., 2019; Wang et al., 2021b; Meng et al., 2022; Dong et al., 2023). Among those knowledge-enhanced models, many works use knowledge representation-based methods to incorporate factual knowledge (Zhang et al., 2019; Su et al., 2021a; Ye et al., 2022; Peters et al., 2019; Wang et al., 2021d; Yamada et al., 2020; Sun et al., 2020a). Other models use other forms to integrate knowledge into the model (Liu et al., 2020b; Meng et al., 2021; Hosseini et al., 2022,?; Lu et al., 2022).

Notably, the above methods are all based on the idea of directly integrating knowledge into the model, while the goal of FAvPD is to construct knowledge into samples and guide the models to learn the implicit knowledge patterns.

2.2. Continual Pretraining

The continual pretraining of LLMs, also called adaptation in some cases, can significantly improve the performance of some downstream tasks (Gururangan et al., 2020). A common solution is to conduct a further pre-training (Gururangan et al., 2020; Howard and Ruder, 2018; Phang et al., 2018; Gururangan et al., 2020; Qin et al., 2023) on the available domain- or task-specific training data. Additionally, there exist more advanced and sophisticated methods that yield superior results in proprietary domains (Lee et al., 2019; Beltagy et al., 2019; Araci and Genc, 2019; Huang et al., 2019). For example, PMC-LLaMA further finetuning LLaMA on medical papers (Wu et al., 2023). Such methods achieve a significant gain in performance in their respective domains but are expensive and time-consuming as they require a large number of domain corpora (usually several GBs).

We argue that these tasks require large amounts of training data and high computing power costs. However, the additional tuning in the FAvPD can be much faster and more lightweight with sampling part of the downstream task training set data (usually within 1 MB).

2.3. Prompt Learning

Since the emergence of GPT-3 (Brown et al., 2020), prompt-based learning has received considerable attention. GPT-3 (Brown et al., 2020) demonstrates that with prompt-tuning and in-context learning, large-scale language models can achieve superior performance in the low-data regime. The following works (Schick and Schütze, 2021a,b) argue that small-scale language models (Radford et al., 2018a; Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Hu et al., 2022b; Ding et al., 2021) can also achieve decent performance using prompt-tuning. Prompt-based learning works by utilizing the knowledge acquired by the pre-trained language models on a large amount of text data to solve various types of downstream tasks (Liu et al., 2021).

FAvPD uses prompts for different goals. The above work mainly uses labels as the target objective of prompt-tuning to directly train the model's downstream task capabilities. However, in FAvPD, the object that prompts needs to fill in the blanks is the information from distinct sources. We argue that this method can help the model to mine the implicit patterns between exogenous information and the text.

3. Method

As mentioned above, the unified FAvPD framework includes a special prompt structure for the information from distinct sources and an information reconstruction loss function. In this section, we will introduce both in detail.

3.1. Preliminary

Before we begin, we first explain a very important thing, that is, the stage in which FAvPD works. The goal of FAvPD is to enhance the Large Language Models (LLMs) by forming a synergy of the combination of three items the external knowledge, and the task-specific text associated with the corresponding ground-truth labels. In our proposal, this unified form of domain-specific data is encapsulated and then exploited in a new additional tuning stage. Put another way, FAvPD sheds some light on the following three-stage workflow: **Pre-Training -> Additional Tuning -> Fine-Tuning** . Noticed that the additional tuning is designed to be lightweight, data-efficient, and fast.

3.2. Notations

We denote the train set of downstream task as $\mathcal{D} = \{x_n, y_n\}_{n=1}^{N}$, where x_n denotes the input sentence, y_n the associated gold label and N the size of dataset $\mathcal{D}$. In particular, given a sentence $\boldsymbol{x} = [\ w_1, w_2,\ \dots\ ]$, it is an ordered sequence of tokens. We use $\mathcal{V}$ to denote the vocabulary set used in LLMs covering all the tokens inside both pre-training and downstream corpora. Noted, we ignore the data point index n for simplicity. We denote $\mathcal{K}$ as the set of external knowledge-base and $\{e, k\} \in \mathcal{K}$, where e is the entity and k is the corresponding factual knowledge.

3.3. Unified External Information Prompt

The core of FAvPD is to introduce an additional phase of pre-training, facilitated by a unified form of data envelope consisting of downstream text, labels, and an external knowledge component. In particular, unlike the usual fine-tuning, adaption, or knowledge-injection methods that focus on any element singly, FAvPD proposes to unify all three modalities.

Context with External Knowledge Source. To begin with, we add an external knowledge component to the envelope. We offer a simple example in Figure 2.

First, we gather the context information by inserting the input sentence x individually drawn from the downstream corpus $\mathcal{D}$. Afterwards we adopt a standardized entity-linking tool (Wu et al., 2020) that scans $\boldsymbol{x}$ together with searching $\mathcal{K}$. Noted, we use BLINK (Wu et al., 2020) toolkit made by Meta for entity linking. Feeding $\boldsymbol{x}$ into the entity-linking tool returns two variables: the entity-mention $w^{(e)}$ where $w^{(e)} \in \boldsymbol{x}$ denoting a sub-sequence related to the corresponded entity appearance on the knowledge base, together with its corresponding knowledge description, k. Formally, we write down the prompted knowledge: $\textbf{prompt}_k = [w^{(e)}, \text{is}, \text{a}, k]$. Notice that we simply adopt "`[] is a []`" as the standard template to fill in the prompt.

Context with Gold Labels Besides the external source of knowledge, we further incorporate the gold labels from the downstream tasks into the data envelope. Formally, for given input text x, we draft the corresponding label y and construct the prompted label as follows: $\textbf{prompt}_l = [w^{(e)}, \text{is}, \text{a}, y]$. Notice that in our targeted tasks, y is not only a quantized indexed number but also expresses entity information.

Final Assemble We use the lemmatization tool to process both constructed prompts. To process them, we concatenate the input text, and both prompts to form a final text structure of data envelope: $\boldsymbol{x'} = [\boldsymbol{x}, \textbf{prompt}_k, \textbf{prompt}_l]$. This in turn yields the dataset applied for the adaptive pre-training phase, $\mathcal{D}' = \{x'\}_{m=1}^{M}$[2] in a self-supervised manner.

Invisible Attention To adapt to input in the form of prompted data, we modify the embedding and encoder layers upon a vanilla Transformer (see Figure 1 using the same Kylian Mbappé example). For token embedding, we insert prompts into the original context. We tag the original text with identical positional encoding with BERT and extend it by incrementing the token index for the appended prompt.

To mitigate the effect of prompt on irrelevant context other than its corresponding entity, we use a mask matrix $M \in \mathbb{R}^{n*n}$ (Liu et al., 2020a) to block part of the self-attention messages. For example, tokens in the original text $\boldsymbol{x}$ that are not the entity $w^{(e)}$ should not have direct attention to tokens in prompt sentence $\boldsymbol{prompt_{k/l}}$. The representation

[2] $M \leq N$ since we sample some examples from the original dataset for FAvPD, see Section A.1 for more details.

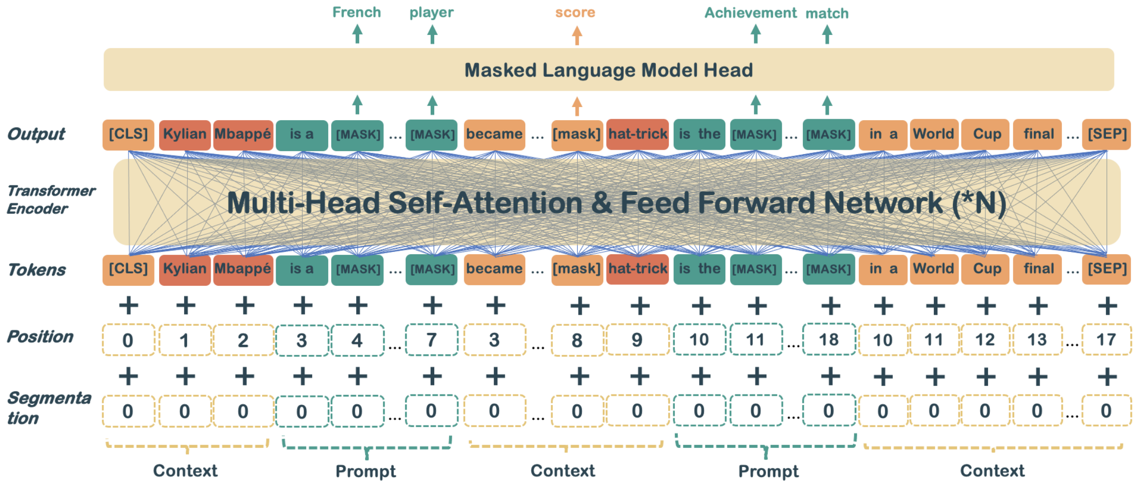
French
player
score
Achievement
match
Masked Language Model Head
Output
[CLS] Kylian Mbappé is a [MASK] ... [MASK] became ... [mask] hat-trick is the [MASK] ... [MASK] in a World Cup final ... [SEP]
Transformer Encoder
Multi-Head Self-Attention & Feed Forward Network (*N)
Tokens
[CLS] Kylian Mbappé is a [MASK] ... [MASK] became ... [mask] hat-trick is the [MASK] ... [MASK] in a World Cup final ... [SEP]
Position
0 1 2 3 4 ... 7 3 ... 8 9 10 11 ... 18 10 11 12 13 ... 17
Segmentation
0 0 0 0 0 ... 0 0 ... 0 0 0 0 ... 0 0 0 0 0 ... 0
Context
Prompt
Context
Prompt
Context

Figure 1: The model architecture of FAvPD. Context refers to the input text itself, and Prompt is the incremental text based on external source information. Fig 2 provides an example.

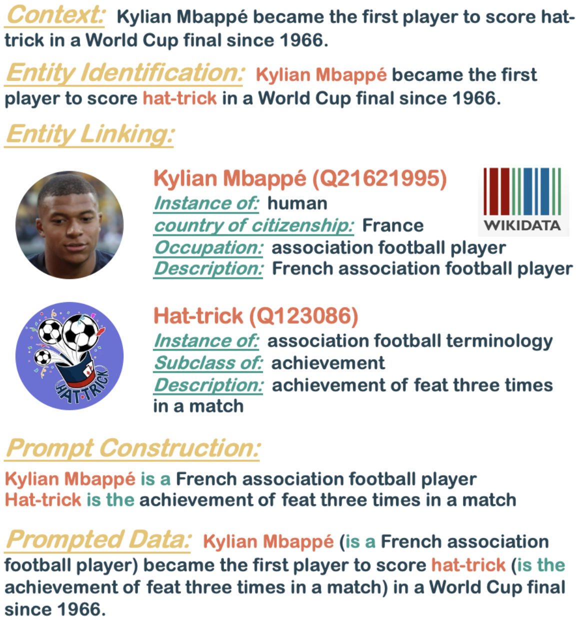
Context: Kylian Mbappé became the first player to score hat-trick in a World Cup final since 1966.
Entity Identification: Kylian Mbappé became the first player to score hat-trick in a World Cup final since 1966.
Entity Linking:
Kylian Mbappé (Q21621995)
Instance of: human
country of citizenship: France
Occupation: association football player
Description: French association football player
WIKIDATA
Hat-trick (Q123086)
Instance of: association football terminology
Subclass of: achievement
Description: achievement of feat three times in a match
HAT TRICK
Prompt Construction:
Kylian Mbappé is a French association football player
Hat-trick is the achievement of feat three times in a match
Prompted Data: Kylian Mbappé (is a French association football player) became the first player to score hat-trick (is the achievement of feat three times in a match) in a World Cup final since 1966.

Figure 2: The illustration on prompted data construction.

after masked self-attention is denoted as

$$\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v} = \boldsymbol{x}'\boldsymbol{W^q}, \boldsymbol{x}'\boldsymbol{W^k}, \boldsymbol{x}'\boldsymbol{W^v} \tag{1}$$

$$\textbf{Attn}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = softmax(\frac{\boldsymbol{qk^T} + \boldsymbol{M}}{\sqrt{\boldsymbol{d_k}}})\boldsymbol{v} \tag{2}$$

where $\boldsymbol{x}' = [\boldsymbol{x}, \textbf{prompt}_k, \textbf{prompt}_l] \in \mathbb{R}^{n*d}$ is the prompted input sequence. $\boldsymbol{W^q}, \boldsymbol{W^k}, \boldsymbol{W^v} \in \mathbb{R}^{d*d_k}$ are learnable parameters. $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}$ are the query vectors, key vectors and value vectors, respectively. n is the length of the input sequence. d is the dimension of the input embedding. d_K is the dimension of the key vectors. $\boldsymbol{M} \in \mathbb{R}^{n*n}$ is the mask matrix defined as

$$M_{ij} = \begin{cases} -\infty & w_i \in \boldsymbol{x} \wedge w_j \in \textbf{prompt}_{k(l)} \\ 0 & otherwise. \end{cases} \tag{3}$$

where $w_i, w_j \in \boldsymbol{x}'$.

3.4. Adaptive Tuning Objective

Based on the above text processing, FAvPD will further tune the LLM on the prompted data $\mathcal{D}' = \{x'\}_{m=1}^{M}$. In this section, we introduce the adaptive tuning procedure on the LLM. In particular, we adopt different training objectives for the text part ($\boldsymbol{x} \in \boldsymbol{x}'$) and the prompt part ($\textbf{prompt}_k, \textbf{prompt}_l \in \boldsymbol{x}'$) respectively. Thus we get two parts of loss, i.e. $\mathcal{L}_{\text{text}}$ and $\mathcal{L}_{\text{prompt}}$.

3.4.1. Common Pretraining Objective

The goal of $\mathcal{L}_{\text{text}}$ is to maintain the language capabilities that the model has acquired. Thus, we followed the original pre-trained objective function of the model. Specifically, for **encoder-only** models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), we adopt the masked language modeling objective. For the **decoder-only** models such as GPT (Radford et al., 2018b), LLaMA (Wu et al., 2023), we adopt a next-token prediction objective.

Encoder-only Models We follow the masking strategy and the masked language modeling objective of BERT (Devlin et al., 2019) in the text part. Specifically, given a sequence $\boldsymbol{x} = \{w_i\}$, we corrupt it into $\tilde{\boldsymbol{x}}$ by masking 15% of its tokens at random and then re-train the LLM parameterized by θ to reconstruct $\boldsymbol{x} = \{\tilde{w_i}\}$ by predicting the masked tokens $\tilde{x}$ conditioned on $\tilde{\boldsymbol{x}}$:

$$\mathcal{L}_{\text{text}}(\theta) = -\sum_{i \in \mathcal{C}} log\ p_\theta(\tilde{w_i} = w_i | \tilde{\boldsymbol{x}}) \tag{4}$$

where $\mathcal{C}$ is the index set of the masked tokens in the sequence $\boldsymbol{x}$.

Decoder-only Models For the decoder-only models, we follow GPT's method (Radford et al., 2018b). Specifically, for the sequence $\boldsymbol{x} = \{w_i\}$, we use a standard language modeling objective to minimize the following loss:

$$\mathcal{L}_{\text{text}}(\theta) = -\sum_{i \in \mathcal{C}} log\ p_\theta(w_i | w_{i-k}, \ldots, w_{i-1}) \tag{5}$$

where k is the size of the context window.

3.4.2. Information Reconstruction Objective

In order to enhance the pre-trained model's ability to process downstream domain-specific text, we introduced the i reconstruction objective, i.e., $\mathcal{L}_{\text{prompt}}$. Unlike general knowledge enhancement methods that directly integrate knowledge information into the model, this loss function assists the model in learning the implicit knowledge pattern in the information from distinct sources.

Since the structure of the prompt part conforms to the preset template (starting with an entity, followed by linking verbs and ending with a knowledge phrase), we redesign the mask strategy for its MLM optimization. To ensure semantic integrity, we replace the token of the knowledge phrase with the [MASK] symbol with a probability of 30% and keep the other part of the prompt template unchanged. Specifically, given a sequence **prompt**$_k$(or **prompt**$_l$) $= \{w_i^p\}$, we corrupt it into $\{\tilde{w_i^p}\}$ by masking 30% of its tokens at random and then re-train the LLM parameterized by θ to reconstruct $\{\tilde{w_i^p}\}$ by predicting the masked tokens $\tilde{w^p}$ conditioned on $\{\tilde{w_i^p}\}$:

$$\mathcal{L}_{\text{prompt}}(\theta) = -\sum_{i \in \mathcal{C}_p} log\ p_\theta(\tilde{w_i^p} = w_i^p | \{\tilde{w_i^p}\}) \tag{6}$$

where $\mathcal{C}_p$ is the index set of the masked tokens in the sequence **prompt**$_k$(or **prompt**$_l$).

3.4.3. The Total Loss

Finally, the total loss can be computed as:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \mu \mathcal{L}_{\text{prompt}} \tag{7}$$

where $\mu \in (0, 1)$ is the balancing coefficient.

4. Experiment

In this section, we present the experimental results obtained from employing multiple model architectures across various task types. These results effectively demonstrate the efficacy of the FAvPD.

4.1. Baselines and Notations

We experiment with BERT-base (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), RoBERTa-large (Liu et al., 2019), LLaMA-7b and LLaMA-13b (Touvron et al., 2023), implemented by Huggingface[3], as our baseline and backbones.

In our comparison map, we are mainly concerned with the following setup:

- **Standard Supervised Fine-Tuning.** We primarily stick to Wolf et al. (2020) and apply supervised fine-tuning to LLMs.
- **Plain Adaptation.** As we mentioned, the plain adaptation methods involve an unlabeled downstream text dataset fed back to the LLMs for additional self-supervised training (Gururangan et al., 2020). These comparisons would manifest the effect of incorporating external knowledge and gold labels.
- **Knowledge-enhanced LLMs.** Knowledge-enhanced LLMs (KE-PLMs) inject knowledge (of various forms) from the considered domain. See Section A.2 for more details.
- **LLMs with the prompt.** Recently, there has been a popular method of giving external information to large models in the form of prompts. Therefore, in order to better demonstrate the effectiveness of FAvPD, we utilize the unified external information prompt in the FAvPD framework as a prompt to LLaMA (Touvron et al., 2023). Then, we evaluated the performance of LLaMA-7b in the downstream tasks after obtaining the prompt
- **Our Method.** We perform three implementations of FAvPD: FAvPD (text + knowledge), FAvPD (text + label), and FAvPD (text + knowledge + label). These notations are denoted with various forms of data envelope. Essentially, on the one hand, we attempt to justify the necessity of a unified form of available downstream and domain information. On the other hand, with lateral comparison — for example, FAvPD (text + knowledge) comparing with knowledge-injection methods — we want to showcase the efficacy and efficiency advantages of our very simple method.

[3] https://github.com/huggingface/transformers

Architecture	Models	Injection		Datasets					
				Open Entity			FIGER		
		Knowledge	Label	Prec.	Rec.	Mi-F1	Acc.	Ma-F1	Mi-F1
BERT-base	Standard Supervised Fine-Tuning	-	-	76.37	70.96	73.56	52.04	75.16	71.63
	ERNIE (Zhang et al., 2019)	✓	-	78.42	72.90	75.56	57.19	76.51	73.39
	KnowBERT (Peters et al., 2019)	✓	-	78.6	71.6	75.0	57.0	79.8	75.0
	K-BERT (Liu et al., 2020a)	✓	-	76.7	71.5	74.0	56.5	77.1	73.8
	CoKeBERT (Su et al., 2021b)	✓	-	78.0	73.3	75.6	57.9	79.7	75.3
	Adaptation (Gururangan et al., 2020)	✗	✗	76.44	73.63	75.01	53.46	76.92	72.64
	FAvPD (Our Method)	✓	✗	74.15	78.05	76.05	62.17	77.72	76.77
	FAvPD (Our Method)	✗	✓	77.64	74.21	75.89	60.92	79.96	76.90
	FAvPD (Our Method)	✓	✓	74.99	79.36	**77.11**	**64.12**	**83.26**	**78.81**
RoBERTa-base	Standard Supervised Fine-Tuning	-	-	77.4	73.6	75.4	56.3	76.9	74.2
	CoLAKE (Sun et al., 2020b)	✓	-	77.0	75.7	76.4	-	-	-
	KEPLER (Wang et al., 2021c)	✓	-	77.8	74.6	76.2	62.0	81.8	77.4
	CoKeBERT (Su et al., 2021b)	✓	-	76.8	74.2	75.6	62.2	82.3	77.7
	Adaptation (Gururangan et al., 2020)	✗	✗	75.83	75.32	75.57	56.13	81.80	77.28
	FAvPD (Our Method)	✓	✗	78.51	74.05	76.22	66.61	83.04	79.52
	FAvPD (Our Method)	✗	✓	78.28	74.58	76.39	65.72	83.64	79.56
	FAvPD (Our Method)	✓	✓	78.66	76.84	**77.74**	**68.56**	**85.26**	**81.71**
RoBERTa-large	Standard Supervised Fine-Tuning	-	-	77.55	74.95	76.23	56.31	82.43	77.83
	K-Adapter (Wang et al., 2021b)	✓	-	79.30	75.84	77.53	59.50	84.52	80.42
	LUKE (Yamada et al., 2020)	✓	-	79.9	76.6	78.2	61.7	82.4	77.8
	CokeBERT (Su et al., 2021b)	✓	-	78.1	76.9	77.5	58.3	82.3	77.8
	Adaptation (Gururangan et al., 2020)	✗	✗	78.86	75.63	77.21	56.31	82.66	77.75
	FAvPD (Our Method)	✓	✗	78.73	76.21	77.45	67.14	84.31	80.02
	FAvPD (Our Method)	✗	✓	78.29	76.52	77.40	66.79	83.83	79.88
	FAvPD (Our Method)	✓	✓	78.22	78.26	**78.24**	**70.16**	**86.84**	**82.01**

Table 1: The Performance of FAvPD on Entity Typing Task. We designed experiments on models with three different architectures. Injection represents the external information we combine when doing adaptive training on the task text.

Architecture	Models	Injection		Datasets	
		Knowledge	Label	Re-TACRED	TACREV
RoBERTa-large	Standard Supervised Fine-Tuning	-	-	84.9	76.0
	LUKE (Yamada et al., 2020)	✓	-	90.3	80.6
	Adaptation (Gururangan et al., 2020)	✗	✗	88.2	78.3
	FAvPD (Our Method)	✓	✗	90.5	81.7
	FAvPD (Our Method)	✗	✓	90.7	82.0
	FAvPD (Our Method)	✓	✓	**90.7**	**82.1**

Table 2: The Performance of FAvPD on Relation Extraction Task. We use Micro-F1 as our evaluation metric. TACREV is the abbreviation of the TACRED Revisited dataset.

4.2. Datasets and Metrics

We evaluate our method on two entity typing tasks, i.e., Open Entity (Choi et al., 2018) and FIGER (Ling et al., 2015). Further, we evaluate our method on a relation extraction task, i.e., Re-TACRED (Stoica et al., 2021) and TACRED Revisited (Alt et al., 2020). The statistics of these three datasets are shown in Table 5. To speed up the adaptation process, we sample part of the data from the training set as the corpus for the prompted data envelope. As shown in Table 6, to provide a rough estimation of data complexity, we enumerate the size of adaptation data x' required in the phase of adaptive pre-training; note this includes the sample number of input sentences x, prompted knowledge $\textbf{prompt}_k$ and prompted label $\textbf{prompt}_l$.

Metrics Generally, we follow the prior work's evaluation metrics on the above datasets: precision, recall, and micro-F1 for Open Entity and TACRED, and accuracy, loose macro/micro (Ling and Weld, 2012) for FIGER. Besides, we further demonstrate the efficiency advantage of FAvPD. In that, we mainly concerned with the following items: (i)-the convergence speed and (ii)-the quantity of extra data acquired to attain a decent performance.

4.3. Results and Discussion

The results shown in Table 1 verify the effectiveness of the FAvPD framework. The implementations of FAvPD have achieved the best F1 Score among those baselines implemented on the BERT-base. Compared with standardized fine-tuning, FAvPD exceeds the baseline by **3.55** of F1-Score for Open Entity and **7.18** for FIGER, which demonstrates our method effectively solves the domain discrepancies. Besides, FAvPD also outperforms Plain Adaptation

Methods	Settings	Injection		Datasets	
		Knowledge	Label	MedQA-USMLE	MedMCQA
ChatGPT (OpenAI, 2022)	Zero-shot	-	-	57.00	44.70
OPT-6.7b (Zhang et al., 2022)		-	-	27.34	28.64
Galactica-6.7b (Taylor et al., 2022)		-	-	30.16	30.48
LLaMA-7b (Touvron et al., 2023)		-	-	27.10	24.30
LLaMA-7b with prompt		-	-	26.71	25.89
Standard Supervised Fine-Tuning	PEFT*	-	-	27.34	32.37
Adaptation (Gururangan et al., 2020)		✗	✗	27.73	35.81
FAvPD on LLaMA-7b (Our Method)		✓	✗	31.26	40.59
FAvPD on LLaMA-7b (Our Method)		✗	✓	29.85	39.44
FAvPD on LLaMA-7b (Our Method)		✓	✓	**32.68**	**42.58**

Table 3: The Performance of FAvPD on Question Answering Task. Accuracy Score Reported. PEFT* indicates we apply LoRA (Hu et al., 2022a) tuning (100M trainable parameters) to LLaMA-7b Architecture.

by a large margin, showing the important impact of prompted knowledge and prompted labels. Further, FAvPD outperforms the knowledge-enhanced LLMs, which proves that LLM can efficiently capture domain-related knowledge in the process of FAvPD.

To further demonstrate the effectiveness of our approach, we conduct experiments on RoBERTa-base architectures. As shown in Table 1, we observed similar experimental results. FAvPD based on RoBERTa-base architecture makes a further improvement compared with FAvPD based on BERT-base. Our method outperforms most knowledge injection methods, showing its incredible effectiveness and efficiency. Notably, in FIGER, FAvPD exceeds the baseline by **12.26** of accuracy and **7.51** of F1 score.

We conduct a further experiment on the architecture of a large version of Roberta. The results shown in Table 1 and Table 2 verify the effectiveness of the FAvPD framework in RoBERTa-large architecture. In the experiment of FAvPD based on RoBERTa-large, we achieved state-of-the-art results for Open Entity and FIGER. Specifically, FAvPD achieved 78.24 of F1-Score for Open Entity and 86.84 of F1-Score for FIGER, showing that our method is also helpful for large models with rich knowledge. For the relation extraction task, FAvPD exceeds the baseline by **5.8** of F1-Score for Re-TACRED and **6.1** for TACRED Revisited.

To further verify the generalizability of our method on larger parameter scale models, we implement our method on the LLaMA-7b (Touvron et al., 2023) model and evaluate it on two medical question answering datasets, i.e. MedQA-USMLE (Jin et al., 2020) and MedMCQA (Pal et al., 2022). As shown in Table 3, we achieved FAvPD in combination with the method of Parameter-Efficient Fine-Tuning (PEFT), and the accuracy exceeded the baseline by **5.34** and **10.21**, respectively.

4.4. Low Resource Costs and High Efficiency

In addition to achieving significant performance improvements, FAvPD also enjoys low resource costs. In this section, we will analyze the efficiency advantages of FAvPD from multiple dimensions: data cost, computing power cost, and time cost.

4.4.1. Computational Speedup

To demonstrate that our approach is faster and lighter than knowledge injection frameworks, we analyze the scale of the training corpus and time consumption. As shown in Table 4, we conclude that FAvPD requires only **1%** of the corpus than the typical knowledge injection frameworks while achieving superior performance. For the last column from this table, FAvPD generally costs orders of magnitude less computational resources. To name a few, FAvPD only requires 1 NVIDIA 2080Ti within 1 hour. If converted to FLOPs, it is at most 1% of other methods.

This is, in particular, in sharp contrast to the prior knowledge-injection frameworks, as shown in Table 4. Overall, the above results demonstrate that FAvPD enjoys improvement in both performance efficacy and training efficiency compared to prior work.

4.4.2. Convergence Rate

To qualitatively analyze the influence of FAvPD on LLM in the fine-tuning stage, we meter the F1 Score throughout the training process. As shown in Figure 3, FAvPD further accelerates the convergence rate on the downstream tasks. Overall, FAvPD achieves optimal results with fewer fine-tuning steps.

4.5. Transferability Assessment

In addition to superior performance on downstream tasks, we also measure the effect of FAvPD by transferability, and the results show that FAvPD

Model	Statistics of Corpus and Knowledge	Computational Consumption (FLOPs)	Computational Consumption Details
ERNIE (Zhang et al., 2019)	4500M subwords, 140M entities	9.30×10^{18}	8 NVIDIA 2080Ti GPUs for 24 hours
CoLAKE (Sun et al., 2020b)	26M examples, 3M entities	1.53×10^{19}	8 32G NVIDIA V100 GPUs for 38 hours
K-Adapter (Wang et al., 2021b)	5.5M sentences, 1M examples	1.94×10^{19}	4 16G NVIDIA V100 GPUs for 4 days
LUKE (Yamada et al., 2020)	3.5B words, 11M entities	5.81×10^{20}	16 NVIDIA Tesla V100 GPUs for 30 days
FAvPD(Ours)	10K sentences, 26K entities	4.84×10^{16}	1 NVIDIA 2080Ti GPU within 1 hour

Table 4: Comparison of our method and Knowledge Enhanced LLMs. The above “FLOPs" results are inferred estimates based on relevant hardware data reported in other work and are not our reproducible results. The reference data shows that the computing power of NVIDIA RTX 2080 Ti is 13.45TFLOPS, and the computing power of NVIDIA Tesla V100 is 14TFLOPS. By default, the GPU uses peak performance and a utilization of 100%

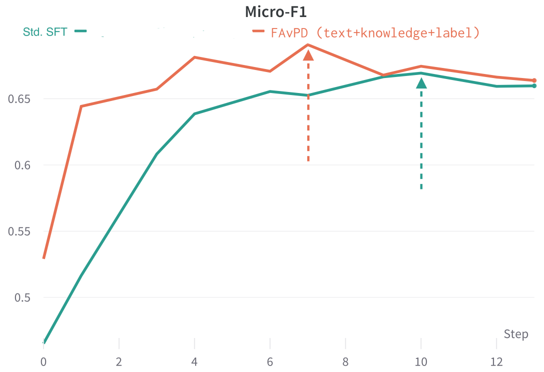

Figure 3: Variation of Micro-F1 Score with training steps during fine-tuning. The upper curve indicates the implementation of FAvPD (text+knowledge+label). The down curve indicates the implementation of standardized Fine-tuning. The arrow indicates the moment when the highest Micro-F1 Score is reached.

can effectively improve the transferability of the model. Transferability measures how well a model performs when transferring from a pre-training task to a downstream task. FAvPD aims to improve the transferability of LLMs, to perform better in the downstream task. In this work, we use the Logarithm of Maximum Evidence (LogME) (You et al., 2021) as our assessment. LogME takes the last layer features of the LLM and labels as input and outputs the transferability score of the LLM. A pre-trained model with a higher LogME value will likely have better transfer performance. As shown in Figure 4, plain adaptation slightly improves the transferability of LLM. In contrast, FAvPD significantly improves the transferability performance of LLM.

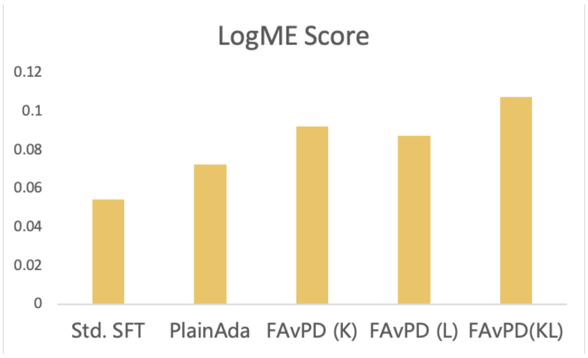

Figure 4: The LogME Score Analysis for BERT-base on Open Entity (The higher the LogME, the stronger the transferability). The notations of Std. Standard Supervised Fine-Tuning (SFT), PlainAda, FAvPD(K), FAvPD(L) and FAvPD(KL) indicate implementation of Standard Supervised Fine-Tuning, Plain Adaptation(text), FAvPD(text+knowledge), FAvPD(text+label) and FAvPD(text+knowledge+label), respectively.

5. Conclusion

Domain discrepancies between downstream tasks and pre-training corpora can significantly impact the performance of LLMs. We address this issue by proposing a lightweight and fast adaptation solution for language models and unifing the employment of external knowledge, domain-specific text and ground-truth labels. The proposed framework, dubbed FAvPD, is extensively validated as effective on many NLP tasks and highly versatile. In this study, we mainly focus on concept-proving the validity of FAvPD on the empirical side. Besides, we hope to explore further in the future the theoretical reasons behind its high efficiency in the adaptive tuning stage.

Limitations

In this work, in much resemblance to the mainstream domain tuning or knowledge injection methodologies, FAvPD generally trades in generalizability to obtain much higher training efficiency and better performance, adapted for each specific domain. In the future, we hope to develop further on this line towards multi-domain generalization.

6. Acknowledgements

This work is majorly supported by the NSFC under Grants (No. 62206247), and in part by the Zhejiang Provincial “Jianbing” “Lingyan” Research and Development Program of China (2024C01035). We also thank the Fundamental Research Funds for the Central Universities (Project QiZhen @ZJU)

7. Bibliographical References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1558–1569, Online. Association for Computational Linguistics.

Dogu Finbert Araci and Zulkuf Genc. 2019. Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems, volume 26. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

BSI. 1973a. Natural Fibre Twines, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. Applied Intelligence, 2(1):37–53.

J.L. Chercheur. 1994. Case-Based Reasoning, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

N. Chomsky. 1973. Conditions on transformations. In A festschrift for Morris Halle, New York. Holt, Rinehart & Winston.

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.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. arXiv preprint arXiv:2108.10604.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. arXiv preprint arXiv:2301.00234.

Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. 2023. Statistical knowledge assessment for generative language models. arXiv preprint arXiv:2305.10519.

Umberto Eco. 1990. The Limits of Interpretation. Indian University Press.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Online. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul Gerhard Hoel. 1971a. Elementary Statistics, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. Elementary Statistics, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Pedram Hosseini, David A Broniatowski, and Mona T Diab. 2022. Knowledge-augmented language models for cause-effect relation classification. In ACL 2022 Workshop on Commonsense Representation and Reasoning.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022b. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Otto Jespersen. 1922. Language: Its Nature, Development, and Origin. Allen and Unwin.

ZIWEI JI, NAYEON LEE, RITA FRIESKE, TIEZHENG YU, DAN SU, YAN XU, and ETSUKO ISHII. 2022. Survey of hallucination in natural language generation. ACM Comput. Surv, 1(1).

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4):1234–1240.

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

Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. Transactions of the Association for Computational Linguistics, 3:315–328.

Xiao Ling and Daniel Weld. 2012. Fine-grained entity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 26, pages 94–100.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020a. K-bert: Enabling language representation with knowledge graph. Proceedings of the AAAI Conference on Artificial Intelligence, 34(03):2901–2908.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020b. K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 2901–2908.

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. arXiv preprint arXiv:1907.11692.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2022. Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. In ICLR 2022 Workshop on Deep Learning on Graphs for Natural Language Processing.

Vivek Madan, Ashish Khetan, and Zohar Karnin. 2021. TADPOLE: Task ADapted Pre-training via AnOmaLy DEtection. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5732–5746, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. Advances in Neural Information Processing Systems, 35:17359–17372.

Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-partitions: Infusing large biomedical knowledge graphs into bert. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4672–4681.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. arXiv preprint arXiv:1811.01088.

Yujia Qin, Cheng Qian, Xu Han, Yankai Lin, Huadong Wang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Recyclable tuning for continual pre-training. arXiv preprint arXiv:2305.08702.

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

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

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference.

In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2339–2352, Online. Association for Computational Linguistics.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. A history of technology. Oxford University Press, London. 5 vol.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. Proceedings of the AAAI Conference on Artificial Intelligence, 35(15):13843–13850.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021a. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. AI Open, 2:127–134.

Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021b. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. AI Open, 2:127–134.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020a. Colake: Contextualized language and knowledge embedding. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3660–3670.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020b. CoLAKE: Contextualized language and knowledge embedding. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. Superheroes experiences with books, 20th edition. The Phantom Editors Associates, Gotham City.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1405–1418.

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

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. KEPLER: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics, 9:176–194.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021d. Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics, 9:176–194.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Further finetuning llama on medical papers.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6397–6407, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6442–6454, Online. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, Maosong Sun, and Zhiyuan Liu. 2022. A simple but effective pluggable entity lookup table for pre-trained language models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 523–529.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 12133–12143. PMLR.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A. Appendix

A.1. Dataset Details

We present the downstream task dataset statistics in Table 5. Further, as shown in Table 6, we show the number of samples for the training set, the number of entity links, and the number of reference labels during the FAvPD process.

Dataset	Train	Dev	Test
Open Entity	$2,000$	$2,000$	$2,000$
FIGER	$2,000,000$	$10,000$	563
TACRED	$68,124$	$22,631$	$15,509$

Table 5: Dataset statistics for Adaptation and finetuning.

Dataset	Adaptive Sentences	Entity Annotation	Label Information
Open Entity	$2,000$	$2,649$	$1,914$
FIGER	$8,267$	$22,306$	$22,033$
TACRED	$13,012$	$26,580$	$13,012$

Table 6: Dataset statistics for Adaptation and finetuning.

A.2. Knowledge Enhanced LLMs Baselines

We compare our method with Knowledge Enhanced PLMs (KE-PLMs) based on the different backbones, i.e., BERT-base, RoBERTa-base, and RoBERTa-large.

- BERT-base: ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019), K-BERT (Liu et al., 2020a), CokeBERT (Su et al., 2021b).
- RoBERTa-base: CoLAKE (Sun et al., 2020b), KEPLER (Wang et al., 2021c), CokeBERT (Su et al., 2021b).
- RoBERTa-large: K-Adapter (Wang et al., 2021b), LUKE (Yamada et al., 2020), CokeBERT (Su et al., 2021b).

A.3. Experimental Setup

A.3.1. Entity Typing

We evaluate our method on Open Entity (Choi et al., 2018) and FIGER (Ling et al., 2015) for entity typing tasks. Open Entity is a collection of about 6,000 sentences with fine-grained entity type annotations, which describe appropriate types for the role the target entity plays in the sentence. FIGER is a much larger dataset with 2M data and more fine-grained classification with 113 types. To fine-tune our models for entity typing, we apply the entity features to the classification layer by adding the special token "$" before and after the labeled entity. To evaluate the model performance, we adopt the evaluation metrics of previous work, i.e., precision, recall, micro-f1 score for Open Entity and accuracy, loose macro, loose micro (Ling and Weld, 2012) score for FIGER.

A.3.2. Relation Extraction

We evaluate our method on TACRED (Zhang et al., 2018) for the relation extraction task. TACRED is a large-scale relation extraction dataset with 106,264 examples built over newswire and web text. To fine-tune our models for relation extraction, we apply the entity features to the classification layer by adding the special token "$" before and after the first entity and adding the special token "#" before and after the second entity. To evaluate the model performance, we adopt the evaluation metrics of previous work, i.e., micro-precision, micro-recall, and micro-f1 score for TACRED.

A.4. TACRED Result

As mentioned in the Experimental Setup, we followed the setting of the previous work and used the original Tacred dataset to conduct the experiment. Fig 7, Fig 8 and Fig 9 are the experimental results under the same setting.

However, we did not put the relevant results in the main part of this paper because we found in subsequent work that the tacred dataset contained annotation errors. This problem was also discovered by Alt et al. (2020) and Stoica et al. (2021). Here are more details:

- TACRED is one of the most widely used RC datasets. Each instance includes a natural sentence sequence, the types and spans of the entity mentioned, and the relation held between the entities or no relation label if no relation was found.
- TACREV (TACRED Revisited (Alt et al., 2020)) is a dataset revised from TACRED, which has the same training data as the original TACRED and extensively relabeled development and test sets.
- Re-TACRED (Stoica et al., 2021) is another completely re-annotated version of the TACRED dataset through an improved crowdsourcing strategy. They re-define relation labels to make them more clear and intuitive and re-annotate the full TACRED dataset.

We use the Re-TACRED and TACREV datasets instead of the TACRED dataset for the following reasons: The TACRED dataset itself has major quality issues, with more than 50% of the most challenging sentences in the validation and test sets being mislabeled and causing an average 8% drop in model performance in the F1-score.

A.5. An Example of the Hyperparameters

In this section we provide some hyperparameters to facilitate readers to reproduce. Below is an examples of hyperparameters we use. For the same set of experiments, we use exactly the same hyperparameters. The following are the hyperparameters used by Roberta in experiments on the Figer dataset:

Model Hyperparameters

- SEED: 120
- TRAIN_EPOCH: 3
- TRAIN_BATCH_SIZE: 64
- EVAL_BATCH_SIZE: 16
- GRADIENT_ACCUMULATION_STEPS: 2
- LEARNING_RATE: 1e-5
- ADAM_BETA_1: 0.9
- ADAM_BETA_2: 0.98
- ADAM_EPSILON: 1e-6
- WARMUP_PROPORTION: 0.1
- WEIGHT_DECAY: 0.01
- MAX_GRAD_NORM: 0.0
- MAX_SEQ_LENGTH: 128
- LOGGING_STEPS: 100
- EARLYSTOP_PATIENCE: 10

We provide more hyperparameter settings in GitHub, https://github.com/Hyatio/FAvPD.

Dataset	Open Entity			FIGER			TACRED		
Model	**P**	**R**	**Mi-F1**	**Acc**	**Ma-F1**	**Mi-F1**	**P**	**R**	**Mi-F1**
Vanilla Fine-Tuning	76.37	70.96	73.56	52.04	75.16	71.63	67.23	64.81	66.00
Plain Adaptation (text)	76.44	73.63	75.01	53.46	76.92	72.64	69.26	63.82	66.43
ERNIE (Zhang et al., 2019)	78.42	72.90	75.56	57.19	76.51	73.39	69.97	66.08	67.97
KnowBERT (Peters et al., 2019)	78.6	71.6	75.0	57.0	79.8	75.0	71.1	66.8	68.9
K-BERT (Liu et al., 2020a)	76.7	71.5	74.0	56.5	77.1	73.8	68.1	66.1	67.1
CokeBERT (Su et al., 2021b)	78.0	73.3	75.6	57.9	79.7	75.3	71.0	66.9	68.9
FAvPD (text+knowledge)	74.15	78.05	76.05	62.17	77.72	76.77	71.84	65.38	68.46
FAvPD (text+label)	77.64	74.21	75.89	60.92	79.96	76.90	71.50	66.02	68.65
FAvPD (text+knowledge+label)	74.99	79.36	**77.11**	64.12	83.26	**78.81**	71.19	67.85	**69.48**

Table 7: The Performance of FAvPD on Entity Typing and Relation Extraction Task (BERT-base Architecture).

Dataset	Open Entity			FIGER			TACRED		
Model	**P**	**R**	**Mi-F1**	**Acc**	**Ma-F1**	**Mi-F1**	**P**	**R**	**Mi-F1**
Vanilla Fine-Tuning	77.4	73.6	75.4	56.3	76.9	74.2	70.8	69.6	70.2
Plain Adaptation (text)	75.83	75.32	75.57	56.13	81.80	77.28	71.89	68.84	70.33
CoLAKE (Sun et al., 2020b)	77.0	75.7	76.4	-	-	-	-	-	-
KEPLER (Wang et al., 2021c)	77.8	74.6	76.2	62.0	81.8	77.4	71.5	72.5	**72.0**
CokeBERT (Su et al., 2021b)	76.8	74.2	75.6	62.2	82.3	77.7	71.3	71.0	71.1
FAvPD (text+knowledge)	78.51	74.05	76.22	66.61	83.04	79.52	70.52	70.74	70.63
FAvPD (text+label)	78.28	74.58	76.39	65.72	83.64	79.56	71.49	70.74	71.11
FAvPD (text+knowledge+label)	78.66	76.84	**77.74**	68.56	85.26	**81.71**	74.71	67.88	71.13

Table 8: The Performance of FAvPD on Entity Typing and Relation Extraction Task (RoBERTa-base Architecture).

Dataset	Open Entity			FIGER			TACRED		
Model	**P**	**R**	**Mi-F1**	**Acc**	**Ma-F1**	**Mi-F1**	**P**	**R**	**Mi-F1**
Vanilla Fine-Tuning	77.55	74.95	76.23	56.31	82.43	77.83	70.17	72.36	71.25
Plain Adaptation (text)	78.86	75.63	77.21	56.31	82.66	77.75	72.12	70.56	71.33
K-Adapter (Wang et al., 2021b)	79.30	75.84	77.53	59.50	84.52	80.42	69.39	74.59	71.89
LUKE (Yamada et al., 2020)	79.9	76.6	78.2	61.7	82.4	77.8	70.4	75.1	**72.7**
CokeBERT (Su et al., 2021b)	78.1	76.9	77.5	58.3	82.3	77.8	71.6	73.0	72.2
FAvPD (text+knowledge)	78.73	76.21	77.45	67.14	84.31	80.02	73.12	70.53	71.80
FAvPD (text+label)	78.29	76.52	77.40	66.79	83.83	79.88	71.46	72.36	71.91
FAvPD (text+knowledge+label)	78.22	78.26	**78.24**	70.16	86.84	**82.01**	71.96	72.39	72.17

Table 9: The Performance of FAvPD on Entity Typing and Relation Extraction Task (RoBERTa-large Architecture).