KnowLA: Enhancing Parameter-efficient Finetuning via Knowledgeable Adaptation

Anonymous ACL submission

Abstract

Parameter-efficient finetuning (PEFT) is a crucial technique to adapt large language models (LLMs) to downstream tasks. In this paper, we study using knowledge graph embeddings to improve the effectiveness of PEFT. We propose a knowledgeable adaptation method called KnowLA. It inserts an adaptation layer into a LLM to integrate the embeddings of entities that appear in the input text. The adaptation layer is trained in combination with LoRA on instruction data. Experiments with two popular LLMs and three knowledge graphs on six datasets demonstrate the effectiveness and robustness of KnowLA. We show that KnowLA can help activate the relevant parameterized knowledge in a LLM to answer a question without changing its parameters or input prompts.

1 Introduction

In the era of large language models (LLMs) with billions and possibly trillions of parameters (Du et al., 2022; OpenAI, 2023; Touvron et al., 2023a), parameter-efficient finetuning (PEFT) stands out as a crucial technique enabling the necessary adaptation of LLMs to downstream tasks. PEFT can efficiently improve the performance of a LLM on a specific task. It freezes most or even all parameters of LLMs and only finetunes a small number of parameters using limited instruction data. LoRA (Hu et al., 2022) is a widely-used PEFT method that trains small low-rank adapters to approximate the large layers in LLMs. Follow-up work improves the efficiency of LoRA by using quantized weights (Dettmers et al., 2023). In contrast, our work seeks to improve the effectiveness of LoRA while preserving comparable efficiency.

Inspired by knowledge-injected pre-trained language models (PLMs), e.g., ERNIE (Zhang et al., 2019), we explore knowledge graphs (KGs) to enhance the PEFT of LLMs with LoRA. A KG is a large-scale structured knowledge base, containing a massive amount of trustworthy knowledge. The typical way of injecting KGs into PLMs in the past several years is incorporating pre-trained entity embeddings at the input layer of a PLM and finetuning the full model on NLP tasks (Zhang et al., 2019; Peters et al., 2019; Yang et al., 2019; Lauscher et al., 2019; Levine et al., 2020; Liu et al., 2021; Lu et al., 2021; Wang et al., 2022). Knowledge injection has improved many PLMs, e.g., BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021). However, previous knowledge injection methods require fully tuning PLMs, which is inapplicable to LLMs. Furthermore, these methods are founded on the encoder-based architecture of PLMs, and their effectiveness for recent decoder-based LLMs remains unknown. The following questions thereby arise: *Can knowledge injection still enhance the PEFT of LLMs? Also, how can knowledge injection be used to enhance PEFT?*

To answer the above questions, in this paper, we propose a knowledgeable adaptation method for PEFT, especially for LoRA, called KnowLA. It inserts an adaptation layer into a pre-trained LLM. The layer integrates external KG embeddings of entities that appear in the input text of the LLM. Entity embeddings and parameters of the LLM are frozen in PEFT. The proposed adaptation layer is trained combined with LoRA on instruction data. The parameters in our adaptation layer are significantly fewer than those in the LLM and even fewer than those in LoRA. Thus, our KnowLA is also a parameter-efficient method without changing the original parameters of the LLM.

We evaluate the effectiveness of KnowLA on six datasets, including commonsense reasoning on CommonsenseQA (Talmor et al., 2019), social interaction reasoning on (Sap et al., 2019) and BIG-Bench Hard (Suzgun et al., 2023), single-hop reasoning of KBQA on WebQuestionSP (Yih et al., 2016), and close-book QA on TriviaQA (Joshi et al., 2017) and TruthfulQA (Lin et al., 2022).

According to experimental results, KnowLA can enhance the effectiveness of LoRA at the expense of a limited number of additional parameters. Furthermore, even when compared to a larger LoRA with a similar number of parameters, KnowLA with a smaller LoRA achieves better results.

We assess the robustness of KnowLA by evaluating it with two popular foundation models (i.e., LLaMA1 (Touvron et al., 2023a) and LLaMA2 (Touvron et al., 2023b)), different instruction data (instruction-following demonstrations in Alpaca2 and Vicuna2), various KGs (i.e., WordNet (Miller, 1995), ConceptNet (Speer et al., 2017) and Wikidata (Vrandecic and Krötzsch, 2014)), and typical embedding learning models (i.e., RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013) and RotatE (Sun et al., 2019)), combined with two PEFT methods (i.e., LoRA (Hu et al., 2022) and AdaLoRA (Zhang et al., 2023)). Experiments show that KnowLA can offer stable improvements.

To understand how KnowLA changes the output of LLMs, we analyze the results from two perspectives, which show some interesting findings: (i) KnowLA with LoRA can align the space of the LLM with the KG embedding space, and (ii) KnowLA can activate the parameterized potential knowledge that originally existed in the LLM, even though the used KG does not contain such knowledge. According to our findings, in some cases, a LLM outputs incorrect answers, not because it does not know answers, but because its relevant knowledge is not activated by the input prompt. Our KnowLA can help activate its relevant knowledge without changing its parameters or input prompts.

2 Related Work

2.1 Knowledge Injection for PLMs

There are three typical knowledge injection methods for PLMs. The first method involves KG embeddings at the input layer of PLMs for joint learning (Zhang et al., 2019; Wang et al., 2021b; Lu et al., 2021). These methods incorporate entity embeddings for classification tasks, and their knowledge injection module is independent of PLMs. This poses challenges for aligning semantic spaces of entities and PLMs. These knowledge injection methods necessitate updating the entire model of PLMs. The second method converts relevant triples in KGs into natural-language sentences to augment the input text of PLMs (Liu et al., 2020; Sun et al., 2020, 2021). The third method introduces adapters into PLMs to enable them learn the KG knowledge (Wang et al., 2021a). Our KnowLA relates to the first line of knowledge injection methods. It is also a variant of the third method. However, previous methods are built on PLMs while our method is the first attempt to LLMs. KnowLA does not update the parameters of LLMs. Instead, it introduces a knowledge adapter during parameter-efficient finetuning to enhance the LLM's capabilities not only for natural language understanding tasks. The injected entity knowledge can be deeply integrated with the LLM's knowledge in subsequent decoding steps.

2.2 Parameter-efficient Tuning for LLMs

Parameter-efficient tuning methods aim to optimize LLMs while minimizing the computational resources and data required. Among them, Adapter Tuning (Houlsby et al., 2019) is a lightweight alternative that inserts small neural modules called adapters in each layer of the PLMs while keeping the majority of the pre-trained parameters frozen. Inspired by the prompt engineering methods, Prefix Tuning (Li and Liang, 2021) sets adjustable prefix tokens in the input or hidden layers, and only these soft prompts are trained. LoRA (Hu et al., 2022) is a low-rank adaptive method that allows training dense layers indirectly by optimizing low-rank factorized matrices that capture changes in dense layers during the adaptation process while keeping the pre-trained weights unchanged. QLoRA (Dettmers et al., 2023) improves LoRA by using NF4 quantization and double quantization techniques. Adalora (Zhang et al., 2023) is an improvement on LoRA, addressing the limitation of the fixed incremental matrix rank r in LoRA, which fails to achieve global optimality. Adalora introduces a method that dynamically allocates rank for downstream tasks, yielding promising results. Our KnowLA follows the research mainstream of the LLM, achieving efficient parameter finetuning with fewer parameters combined with LoRA. Similarly, during the finetuning process, the parameters of LLMs and entity representations are fixed, allowing only gradient backpropagation through the parameters of adapters. This enables the utilization of external knowledge to unleash the potential of LLMs.

3 Method

Considering that the hidden states in Transformer layers encapsulate the parameterized knowledge

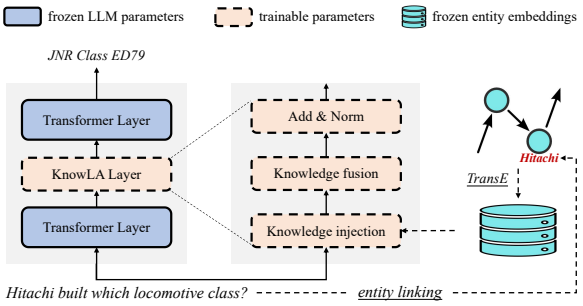

Figure 1: Illustration of knowledgeable adaptation. The KnowLA layer is inserted between two decoder layers of a LLM. It consists of knowledge injection and fusion.

extracted by the LLM (Li et al., 2023), we propose to fuse entity embeddings in a KG with the hidden states of a LLM during PEFT. KnowLA inserts an adaptation layer into a LLM, as shown in Figure 1.

Given a KG and its pre-trained KG embeddings, for an input question $Q = \{q_i\}_{i=1}^{t}$ to a LLM, where q_i corresponds to a set of entities $C(q_i)$ in the KG, and each entity $\overline{c_i}$ belongs to $C(q_i)$ with the corresponding pre-trained embedding $\mathbf{e}_i \in \mathbb{R}^{100}$. Our key idea is to enhance PEFT by injecting the parameterized $\mathbf{e}_i$ from the KG into q_i appearing in the text. This method can be divided into three modules: (i) *Decoder layer*, which learns and propagates the semantic information within the sentences. (ii) *Knowledge mapping*, which maps the entity embeddings from a KG to the LLaMA2 space and infuses it corresponding to the specific words in the question. (iii) *Knowledge fusion*, which further integrates the entity embedding with the textual representation. Given the powerful abilities, popularity and open-source nature of the LLaMA family (Touvron et al., 2023a,b), we currently consider it the foundation to build our KnowLA.

3.1 LLM Encoding

Given a LLM, e.g., LLaMA2, it first encodes the input text to get embeddings for prompts and questions. Specifically, given the prompt p, the LLM first converts it into $Q = \{[s], p, [/s]\}$. The decoder of the LLM tokenizes Q with the bytepair encoding (BPE) algorithm (Sennrich et al., 2016), using the implementation from SentencePiece (Kudo and Richardson, 2018). After tokenization, Q turns into $\{\mathbf{h}_i\}_{i=1}^{k} \in \mathbb{R}^{d_1}$. We take it as the input to the LLM.

3.2 Knowledge Mapping and Injection

The text representation of the L-th decoding layer in the LLM is denoted by $\mathbf{h}^l$. In the knowledge mapping module, to align with the pre-norm mode adopted by the decoder and mitigate the issues of gradient vanishing or exploding, we apply RMSNorm (Zhang and Sennrich, 2019) to the input $\mathbf{h}^l$ received by the decoder. We also map the semantic space of entity embeddings to the semantic space of the LLM for transformation, aiming to improve knowledge injection and integration.

The BPE encoding method employed by many LLMs would let each token have multiple sub-tokens after encoding. Let the number of tokens be p, which can be represented as $\{\mathbf{h}_i^l\}_{i=1}^{p}$. The corresponding entities are denoted by $C(q_i)$. To better calculate the relevance between different entities and the given word, we unify the representations of the p sub-tokens as $\mathbf{u}_i$ using mean pooling:

$$\mathbf{u}_i = \mathrm{AvgPooling}(\mathbf{h}_1^l, \ldots, \mathbf{h}_p^l). \quad (1)$$

Since LLMs are employed for handling complex natural language tasks, it is essential to have input dimensions sufficiently large to accommodate the intricacies. To enhance the expressive ability of entity representation $\mathbf{e}_i$ and align with the semantic space of LLaMA2, we expand its dimension to enrich the representation of $\mathbf{e}_i$:

$$\mathbf{e}_i = \mathbf{W}_d\big(\mathrm{SwiGLU}(\mathbf{W}_u\,\mathbf{e}_i + \mathbf{b}_u)\big), \quad (2)$$

where $\mathbf{W}_u \in \mathbb{R}^{d_3 \times d_2}$, $W_d \in \mathbb{R}^{d_1 \times d_3}$ and $b_u \in \mathbb{R}^{d_3}$ are trainable weight parameters. SwiGLU (Shazeer, 2020) is a activation function.

3.3 Knowledge Fusion

To prevent the LLM from encountering unfamiliar entities during finetuning in downstream tasks, as well as to ensure the extracted entities are relevant to the input tokens, we follow (Yang et al., 2019) to introduce a knowledge sentinel $\overline{e}$. We also calculate its similarity β with each token:

$$\overline{\mathbf{u}}_i = \sum_j \alpha_{ij}\,\mathbf{e}_j + \beta\,\overline{\mathbf{e}}, \quad (3)$$

$$\overline{\mathbf{h}}_i = \theta\,\mathrm{SwiGLU}\big(\mathbf{W}_m[\overline{\mathbf{u}}_i; \mathbf{u}_i] + \mathbf{b}_m\big) + \mathbf{h}_i, \quad (4)$$

where α_{ij} represents the relevance between the i-th token and the j-th entity. Here we limit that: $\sum_j \alpha_{ij} + \beta = 1$. θ serves as a trainable balancing factor to equalize the impact of KG and text. $\mathbf{W}_m \in \mathbb{R}^{2d_1 \times d_1}$ and $\mathbf{b}_m \in \mathbb{R}^{d_1}$ are trainable weight parameters. $\overline{\mathbf{h}}_i$ represents the final representation of knowledge injection and serves as the output of the current adapter, which is then passed as input to the next layer of the decoder.

Similar to other parameter-efficient modules like LoRA (Hu et al., 2022), KnowLA achieves the alignment between KG knowledge and textual semantics by freezing the LLM during finetuning. Alternatively, it can be used in conjunction with LoRA to achieve efficient learning of LLMs with a limited number of parameters. The effectiveness of this module is shortly discussed in the experiments.

4 Experiments

We seek to answer the following research questions through our experiments and analyses:

- What about the effectiveness of KnowLA for different tasks? What about its robustness against different LLMs and KGs?
- Is the improved performance related to the increased number of trainable parameters? Can injecting random noise embeddings also improve the effectiveness of LoRA?
- Why can KnowLA collaborate with LoRA to improve LLMs? Is it also applicable to other LoRA variants such as AdaLoRA?

4.1 Baseline LLMs and Implementation

We consider the following LLMs with 7B parameters as baselines in our main experiments.

- **LLaMA2** is a group of open-source LLMs trained on public datasets with trillions of tokens. We use the LLaMA2-7B model.
- **Alpaca2** is a LLaMA2 variant finetuned with 52,000 instruction-following demonstrations using LoRA (Hu et al., 2022).

In the main experiments, we use the instruction data of Alpaca2 to finetune LLaMA2 with LoRA and our KnowLA. Our KnowLA layer is inserted between the 31st and 32nd layers of LLaMA2. We also consider LLaMA1 and the instruction data of Vicuna2 (Chiang et al., 2023) in Sect. (4.10).

For a fair comparison, we use the official hyperparameters and instruction data of Alpaca to finetune LLaMA2-7B to get Alpaca2 and Alpaca2-KG. To study the impact of the number of trainable parameters, we train two LoRA models with different ranks: $r = 16$ and $r = 32$. We keep the input prompts the same for different models in experiments. All models are finetuned on A800 GPUs.

4.2 Datasets and Settings

We consider three types of tasks: multi-choice QA, Closed-book QA, and truthful QA. We use CommonsenseQA (Talmor et al., 2019) and SIQA (Sap et al., 2019) as the multiple-Choice QA datasets, and choose 15 challenging multi-choice tasks from BIG-Bench Hard (BBH) (Suzgun et al., 2023). We use WebQuestionSP (Yih et al., 2016) and TriviaQA (Joshi et al., 2017) for Closed-book QA evaluation. We also use TruthfulQA (Lin et al., 2022) to evaluate whether KnowLA is truthful in generating answers to questions. To assess the direct improvement of our KnowLA to enhance PEFT, we do not introduce other relevant models and employ zero-shot settings for all tasks.

- In **CommonsenseQA**, each sample consists of a question, five candidate answers, and the correct answer. To run LLMs for CommonsenseQA, we adopt the same experimental settings as in (Shwartz et al., 2020) and consider it as a text completion problem.
- **SIQA** is a QA dataset for testing social commonsense intelligence, where each sample consists of a question, three candidate answers, and the correct answer. To evaluate prompt-based methods, we do not use the provided knowledge in the dataset. The settings are the same as in CommonsenseQA.
- **WebQuestionSP** is a KBQA dataset that enhances the original WebQuestion dataset by annotating each answer with corresponding SPARQL queries and removing ambiguous, unclear, or unanswerable questions. Here we treat it as a closed-book QA task.
- **TriviaQA** includes 95K question-answer pairs authored by trivia enthusiasts, that provide high-quality distant supervision for answering the questions.
- **BBH** is a popular aggregated benchmark that focuses on tasks challenging for LLMs. In order to compare scores of different methods on correct answers, we selected fifteen multiple-choice QA datasets from this benchmark.
- **TruthfulQA** is a benchmark to measure whether a language model is truthful in generating answers to questions.

4.3 KGs and Configurations

We use WordNet (Miller, 1995), ConceptNet (Speer et al., 2017), and Wikidata (Vrandecic and Krötzsch, 2014) as the KGs in our method.

- **WordNet** is a lexical KG in English. Nouns,

		CommonsenseQA		SIQA		BIG-Bench Hard	
	#Params	Accuracy	Score	Accuracy	Score	Accuracy	Score
LLaMA2 (7B)	7B	45.37	36.40	46.42	40.58	26.95	24.87
Alpaca2 ($r = 16$)	+0.24%	56.18	46.21	52.30	46.04	28.93	**25.42**
Alpaca2 ($r = 32$)	+0.49%	57.20	46.63	52.76	46.15	28.79	25.36
KnowLA (random)	+0.48%	57.49	47.82	52.61	46.56	29.26	25.34
KnowLA (WordNet)	+0.48%	58.07	**48.35**	**53.22**	46.76	30.00	25.39
KnowLA (ConceptNet)	+0.48%	**58.39**	48.19	**53.22**	**46.81**	**30.19**	25.29
KnowLA (Wikidata)	+0.48%	57.90	47.39	53.21	46.64	29.39	**25.42**

Table 1: QA results on CommonsenseQA, SIQA, and BBH. For KnowLA, the rank of LoRA is $r = 16$.

verbs, adjectives, and adverbs are arranged into synsets, each denoting a separate notion.

- **ConceptNet** is a multi-lingual KG of things people know and computers should know.
- **Wikidata** is a comprehensive repository of structured knowledge across diverse domains. It encompasses various entity types, including individuals, places, concepts, and more.

For KG embedding, we follow (Zhang et al., 2019) and pre-train entity embeddings through TransE (Bordes et al., 2013) as external knowledge. The maximum number of relevant entities selected for each textual token in a question is set to 5.

4.4 Experiments on Multi-choice QA

We evaluate the effectiveness and robustness of KnowLA on multi-choice QA compared with LLaMA2, Alpaca2 ($r = 16$) and Alpaca2 ($r = 32$) which has similar trainable parameters with KnowLA. Following (Shwartz et al., 2020), we compute scores using cross entropy which indicates the confidence of each model for correct answers. We evaluate the impact of WordNet, ConceptNet, and Wikidata on QA performance. Additionally, we introduce randomly initialized embeddings to assess the quality of KG entity embeddings.

The accuracy results are shown in Table 1. Our KnowLA has shown the best performance across three datasets combined with LoRA. Additionally, Alpaca2 ($r = 32$) outperformed Alpaca2 ($r = 16$) on all three datasets, because more trainable parameters typically lead to improved performance.

Moreover, our method with LoRA ($r = 16$) achieves better performance, indicating that our model can better integrate with PEFT methods, surpassing the LoRA with the same parameters. Specifically, when combined with ConceptNet, it achieves an increase from 56.18% to 58.39% on CommonsenseQA, from 52.30% to 53.22% on SIQA and 28.93% to 30.19% on BBH. Due to the fact that ConceptNet stores rich entity knowledge and a greater number of relation types compared to WordNet, its entity embeddings can better enhance LLaMA2's reasoning ability. This suggests that the more extensive the entity coverage in KnowLA, the more significant the increase becomes.

Simultaneously, the performance of KnowLA (random) is inferior to KnowLA (KG), highlighting the greater utility of entity knowledge in KGs for LLMs. Based on the score of each model on the correct answers, it can be seen that after incorporating the KnowLA, all models assign a higher confidence to the correct answers. Therefore, KnowLA can offer a certain degree of improvement for LLMs in commonsense reasoning.

4.5 Experiments on Closed-book QA

In this experiment, we evaluate KnowLA using WebQuestionSP and TriviaQA. Following the answer matching strategy in (Tan et al., 2023), we utilize the subtree labels provided by the constituent tree to extract all noun phrases from the textual answers, calculate their similarities, and determine the correctness of answers exceeding a certain threshold.

The results are shown in Table 2. We find that Alpaca2 ($r = 16$) has a better performance than Alpaca2 ($r = 32$). The reason may be that more parameters in LoRA prone to overfitting in the closed-book QA task. On TriviaQA, KnowLA combined with WordNet improves the results from 68.70% to 69.27%, while combined with ConceptNet, the performance is further enhanced to 69.40%. This indicates that the parameterized entity embeddings can enrich the textual representations. The experimental results demonstrate that this knowledge-enhanced textual representation after finetuning with LoRA can help mitigate the hallucination problem of LLaMA2 to some extent.

On WebQuestionSP, KnowLA (WordNet) and

Methods	WebQuestionSP	TriviaQA
Alpaca2 ($r = 16$)	67.55	68.70
Alpaca2 ($r = 32$)	67.43	67.97
KnowLA (random)	67.68	69.34
KnowLA (WordNet)	67.43	69.27
KnowLA (ConceptNet)	**68.12**	**69.40**
KnowLA (Wikidata)	67.49	68.92

Table 2: QA results on WebQuestionSP and TriviaQA. For KnowLA, the rank of LoRA is $r = 16$.

Methods	BLEU	Rouge-1	Rouge-2	Rouge-L
Alpaca2 ($r = 16$)	0.1657	0.4094	0.2831	0.3892
Alpaca2 ($r = 32$)	0.1637	0.4048	0.2802	0.3851
KnowLA (random)	0.1677	0.4110	0.2850	0.3897
KnowLA (WordNet)	0.1714	0.4143	0.2874	0.3927
KnowLA (ConceptNet)	**0.1747**	**0.4190**	**0.2922**	**0.3975**
KnowLA (Wikidata)	0.1703	0.4135	0.2895	0.3931

Table 3: Results on TruthfulQA. For KnowLA, the rank of LoRA is $r = 16$.

KnowLA (Wikidata) produces similar results. Also, the two Alpaca2 models with different ranks perform similarly. This suggests that the reasoning ability of Alpaca2 performs well on this task, and the performance does not change significantly after knowledge enhancement with KnowLA. We attribute this bottleneck to the model size and the training data of LLaMA2 and Alpaca2.

4.6 Experiments on TruthfulQA

In this experiment, we use TruthfulQA to measure whether our method is truthful in generating answers to questions. Here, we evaluate the content generated by the models using the best answer provided by the TruthfulQA, along with the commonly used metrics BLEU, Rouge-1, Rouge-2, and Rouge-L. The results are shown in Table 3.

Alpaca2 ($r = 32$) still shows lower performance than Alpaca2 ($r = 16$). This further substantiates our conclusion that larger parameters do not necessarily guarantee the accuracy and reliability of the model's output. KnowLA (ConceptNet) performs the best among these models, which indicates that the integration of our method with LoRA can mitigate the hallucination problem of LLaMA2 to some extend and generate more high-quality content.

Besides, we observe that KnowLA (ConceptNet) outperforms KnowLA (WordNet) in all evaluation tasks, and KnowLA (WordNet), in turn, surpasses KnowLA (Wikidata). This further indicates that the entity knowledge within ConceptNet is more suitable for both LoRA and LLaMA2.

4.7 Case Study

We present some improved results of Alpaca2 by incorporating WordNet, ConceptNet, and Wikidata in KnowLA in Figure 3. In Case 1, we discover that after integrating ConceptNet and WordNet with KnowLA, the response precisely describes the correct answers. The contents generated by KnowLA (ConceptNet) and KnowLA (WordNet) are very similar. The content generated by Alpaca2 not only missed significant answers but also misinterpreted the song "Can't Hold Me Down" in the question. Therefore, we believe that KnowLA helps the model better understand questions.

By examining the answers of three models in Case 2, it can be observed that Alpaca2 does not provide an accurate and relevant response, which is similar to the content generated by KnowLA (Wikidata). They both generate deceptive answers. However, after incorporating ConceptNet, KnowLA accurately provides the correct answer in the response. According to Table 2, we believe that the enhancement is not accidental. Moreover, by examining the token-to-entity linking results, we find that *the answer entity "Boojum" does not exist in ConceptNet.* Therefore, we conclude that the KnowLA can stimulate the underlying reasoning abilities of LLMs incorporating with LoRA.

4.8 Why Knowledgeable Adaptation Works?

In this section, we explore why KnowLA works with LoRA in two aspects: the semantic space alignment of entity embeddings and LLaMA2 and the knowledge recall of LLaMA2.

4.8.1 Perspective of Space Alignment

Our KnowLA incorporates a pre-trained LLM and pre-trained KG embeddings for instruction tuning with LoRA. We hereby investigate whether the two heterogeneous representation spaces of the KG and LLM are aligned, to understand how KnowLA works. The results are illustrated in Figure 4, where the last column represents the entity "sentinel". We first acquire the representations of the input tokens in a specific layer, e.g., the 31st layer. Then, we retrieve the top five similar entity embeddings in the KG for each token. Next, to establish the relevance of each token and its corresponding entities, we calculate the attention weights between them. A larger weight suggest a stronger semantic correlation between the token and the mapped entity.

For LLaMA2 (left part of Figure 4), the similarity between entity embeddings and textual repre-

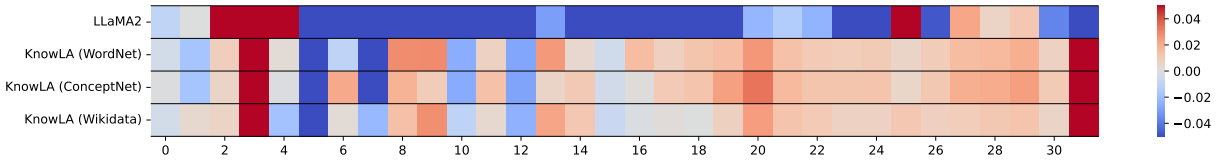

Figure 2: The heatmap indicates the knowledge extraction capabilities between the introduction of KnowLA and Alpaca2, which is calculated by averaging the changes in cosine similarities of the last tokens from 95000 queries across all FFN layers. The x-axis denotes the 32 layers of LLaMA2.

Case 1: Who is featured on Puff Daddy's Can't Hold Me Down?

Answer: Mase

Alpaca2: Puff Daddy's 1997 single "Can't Hold Us Down" features R&B singers Faith Evans and 112. It was released as the second single from Puff Daddy's debut album, No Way Out.

KnowLA (ConceptNet): The song "Can't Hold Me Down" by Puff Daddy features R&B singer Faith Evans and rapper Mase. The song was released in 1997 as the lead single from Puff Daddy's debut studio album, No Way Out.

KnowLA (WordNet): The song "Can't Hold Me Down" by Puff Daddy features R&B singer Faith Evans, rapper Mase, and R&B singer 112. It was released in 1997 as the lead single from Puff Daddy's debut album, No Way Out.

Case 2: In Lewis Carroll's poem The Hunting of the Snark, what did the elusive, troublesome snark turn into to fool hunters?

Answer: Boojum

Alpaca2: The elusive, troublesome snark was able to fool the hunters by transforming itself into a variety of shapes and forms. The snark was able to take the shape of a dragon, a boa constrictor, a rhinoceros, a hippopotamus, a crocodile, and a camel.

KnowLA (ConceptNet): The snark was able to fool hunters by transforming into a variety of shapes and forms. The snark first turned into a Boojum, a creature that could only be seen by those who believed in it.

KnowLA (Wikidata): The snark was able to fool the hunters by transforming into a variety of shapes and forms. The snark was able to take the shape of a beetle, a butterfly, a snake, a fish, a bird, and a mouse, all in an attempt to escape the hunters.

Figure 3: Examples of Alpaca2 and KnowLA (KG) for TriviaQA.

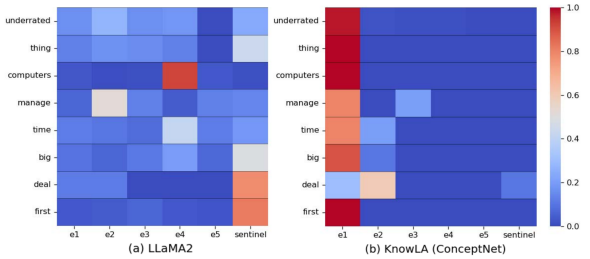

Figure 4: The similarity heatmap between the output representations of text tokens and their corresponding entity embeddings. The x-axis denotes the top-5 similar entities with tokens in the y-axis. (a) The left heatmap presents the similarity of LLaMA2 without finetuning, while (b) the right heatmap presents the similarity after finetuning with our KnowLA (ConceptNet).

sentation is random, without any specific pattern. However, after applying KnowLA, the obtained results demonstrate higher accuracy on the most relevant entities (e_1 in the x-axis). For token "underrated", the relevant entities found in the ConceptNet are "underrated", "underrate", etc. After the finetuning process, the token 'underrated' has displayed the highest correlation with the entity "underrated". This indicates that KnowLA can alleviate the gap between the KG and LLM spaces using instruction tuning with LoRA.

4.8.2 Perspective of Knowledge Recall

We hereby investigate the role of KnowLA in activating LLMs' knowledge. According to (Li et al., 2023; Geva et al., 2021; Meng et al., 2022), the feed-forward network (FFN) layers, which constitute two-thirds of a LLM's parameters, primarily extracts its own knowledge. So, we explore the impact of KnowLA on the FFN layers, to see how KnowLA influences these layers in extracting the knowledge stored in the LLM.

We compute the differences between the hidden state representations of the last token before and after each FFN layer in the LLM. And we analyze the trends in differences of all 32 layers after inserting the KnowLA. We utilize the 95,000 questions from TriviaQA as queries to explore the knowledge stored in the FFN layers of LLaMA2 (7B). The last token in each input query aggregates information from the query. According to (Li et al., 2023), there is a positive correlation between the similarity of hidden states and the consistency of knowledge. Intuitively, we believe that higher differences in representations can indicate the model's ability to extract more information from FFN layers. Therefore, we extract the representations of the last token before and after each FFN layer and compute their

	CSQA		SIQA		BBH	
	Accuracy	Score	Accuracy	Score	Accuracy	Score
RESCAL	**58.39**	46.71	52.10	44.91	27.50	**25.96**
TransE	**58.39**	**48.19**	**53.22**	**46.81**	**30.19**	25.29
RotatE	57.58	46.05	52.00	44.65	27.31	24.94

Table 4: The impact of KG embedding learning models on CommonsenseQA, SIQA, and BBH, which are pretrained on ConceptNet for LLaMA2.

	Methods	Accuracy	Score
LLM side	Alpaca1	56.59	46.03
	KnowLA (LLaMA1)	**57.74**	**46.81**
Data side	Vicuna2	51.52	42.31
	KnowLA (Vicuna2)	**53.56**	**49.09**
PEFT side	Alpaca2 (AdaLoRA)	57.58	**46.67**
	KnowLA (AdaLoRA)	**57.66**	46.30

Table 5: The results with different LLMs, instruction data, and PEFT methods on CommonsenseQA

cosine similarity. After calculating the token similarities, we further evaluate the KnowLA's capacity to extract richer knowledge. The capacity is calculated by subtracting the similarities obtained by KnowLA from those obtained by Alpaca2. The results are shown in Figure 2.

The red color indicates that the representation of the last token, after introducing the KnowLA and undergoing the FFN layers, exhibits a greater change compared to that of Alpaca2. Conversely, the blue color indicates the opposite. We think the representations with greater changes capture more internal knowledge.

After introducing entity embeddings, we observe that our KnowLA enables the LLM to extract richer knowledge at the FFN layers. In contrast, the LLaMA2 extracts less knowledge than Alpaca2. Additionally, according to (Geva et al., 2021), the lower layers of the model's FFN tend to capture shallow patterns, while higher layers learn more semantic patterns. Our framework demonstrates enhanced knowledge extraction capabilities at the higher layers. We attribute the superior results over Alpaca2 to the improvement in the ability to capture semantic patterns. By examining the differences in similarity across the last 16 layers, we find that ConceptNet is maximized across the three KGs. KnowLA (ConceptNet) performs the best on TriviaQA. This further emphasizes that the introduction of ConceptNet substantially extracts more knowledge stored internally in the LLaMA2.

4.9 Impact of KG Embedding Models

We study the impact of embedding learning models used to learn entity embeddings for KnowLA. We obtain entity embeddings of ConceptNet by three representative KG embedding models: RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013), and RotatE (Sun et al., 2019). We show the results of KnowLA with these embeddings on CSQA, SIQA and BBH datasets in Table 4.

We can observe that the entity embeddings obtained through TransE have achieved favorable results. This is attributed to the fact that the embeddings generated by TransE are more suitable for LLaMA2. RotatE employs complex vector representation for entities and achieves subpar results on LLaMA2. This suggests that aligning the complex space of entities with the semantic space of LLaMA2 during finetuning is challenging, leading to a loss of original entity information.

4.10 Robustness of KnowLA

We evaluate the robustness of KnowLA against three factors: On the foundation model side, we use LLaMA1 (Touvron et al., 2023a) as a LLM. On the instruction data side, we finetune LLaMA2 using the Vicuna multi-round dialog data (Chiang et al., 2023) to get Vicuna2 and KnowLA (Vicuna2). On the PEFT method side, we use AdaLoRA (Zhang et al., 2023) to replace LoRA and get Alpaca2 (AdaLoRA) and KnowLA (AdaLoRA).

Table 5 presents the performance of the above models on the commonsense reasoning dataset CommonsenseQA. We can observe that the three KnowLA variants still outperform baselines. This experiment shows that KnowLA is robust and can bring stable improvement when combined with different LLMs, instruction data, and PEFT methods.

5 Conclusion

In this paper, we propose a knowledgeable adaptation method KnowLA. It can be integrated with LoRA and injects entity knowledge into the LLM during the PEFT process. Our experiments demonstrated that, compared to Alpaca2 which is finetuned with LoRA alone, KnowLA exhibits better performance on six commonly used datasets. Also, the entity embeddings pre-trained by TransE are more compatible with LLaMA2. We find that the introduction of KnowLA enables the LLM to activate more diverse knowledge related to semantic patterns from the FFN layers, thereby achieving an improvement in its performance.

Limitations

Currently, our work only incorporates one KG to enhance PEFT. As KGs are incomplete by nature, integrating multiple KGs into our method may further improve performance with knowledge fusion and transfer. Recent work (Huang et al., 2022) reveals that multi-source KG embeddings are more expressive than the embeddings of a single KG. We plan to study multi-source KnowLA in future work.

Besides, we have not attempted other LLMs such as ChatGLM (Zeng et al., 2023) in this work. In the future, we will consider how to efficiently inject KG knowledge with smaller parameters. Meanwhile, we have observed that, with the introduction of random perturbations, LLaMA2 seems to outperform Alpaca2 on some tasks. This discovery may provide interesting directions for future research.

Ethical Considerations

LLMs may produce incorrect and potentially biased content. Experiments show that our method can alleviate this problem to a certain extent, but it is inevitable that the LLM will generate offensive answers. Therefore, extreme caution should be exercised if deploying such systems in user-facing applications. All datasets and models used in this work are publicly available under license.

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

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

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495.

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

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

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual knowledge graph completion with self-supervised adaptive graph alignment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 474–485, Dublin, Ireland. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4*, pages 1601–1611.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Anne Lauscher, Ivan Vulic, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavas. 2019. Informing unsupervised pretraining with external linguistic knowledge. *CoRR*, abs/1909.02339.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4656–4667.

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*, pages 4582–4597, Online.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: precise model editing in a transformer. *CoRR*, abs/2308.08742.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, Virtual Event, February 2-9, 2021*, pages 6418–6425.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2021. KELM: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *CoRR*, abs/2109.04223.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, 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 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Noam Shazeer. 2020. GLU variants improve transformer. *CoRR*, abs/2002.05202.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4615–4629.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. 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.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench

tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4149–4158, Minneapolis, Minnesota.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418.

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

Xinyi Wang, Zitao Wang, Weijian Sun, and Wei Hu. 2022. Enhancing document-level relation extraction by entity knowledge injection. In *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*, volume 13489 of *Lecture Notes in Computer Science*, pages 39–56.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12360–12371.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 1441–1451.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China.