Structure-Aware Adapter for Large Language Model

Anonymous ACL submission

Abstract

Pre-trained Large Language Models (LLMs) have been shown effective in various natural language processing tasks, especially when fine-tuned on specific downstream scenarios. However, the full fine-tuning of LLMs is usually computationally expensive and time-consuming due to the ever-increasing parameter size. In addition, while the LLMs are pre-trained to memorize the facts and knowledge from unstructured textual corpora, they cannot be well generalized to some domain-specific scenarios where additional structured knowledge is required, such as enterprise databases or social graphs. In this paper, we design a novel structure-aware adapter for LLMs to utilize structured relational information from knowledge graphs with a structure-aware relational attention mechanism. The proposed adapter framework only introduces *a small scale* of new parameters and therefore significantly reduces the cost of fine-tuning, without perturbing the initial pre-trained parameters of LLMs. We also propose a knowledge-graph-induced path-of-thought prompt to enhance the utilization of the LLM adapter to retrieve information from the knowledge graph. We evaluate the proposed model on two question-answering benchmarks. The evaluation results show that the proposed method outperforms the state-of-the-art LLM adapters by 4.1%-15.9% and 1.4%-17.6% in question-answering accuracy of CSQA and OBQA datasets. Ablation studies are also discussed to prove the effectiveness of the proposed modules.

1 Introduction

Pre-trained Large Language Models (LLMs), such as LLaMA (Touvron et al., 2023), GPT-3 (Brown et al., 2020), Alexa Teacher Model (FitzGerald et al., 2022; Soltan et al., 2022), and RoBERTa (Liu et al., 2019), have achieved remarkable success in a wide range of natural language processing (NLP) tasks, such as question answering, language translation, text generation, text summarization, etc. The success of LLMs can be attributed to the massive number of model parameters, the pre-training on diverse and extensive text data, and the fine-tuning of specific tasks. However, the full fine-tuning of LLMs is usually computationally expensive and time-consuming. In addition, it can also lead to the problems of catastrophic forgetting and over-fitting, where the model forgets previously learned information or overfits as it adjusts to new task-specific data.

The adaption-based fine-tuning models *freeze* pre-trained parameters of LLMs and only introduce a small scale of trainable parameters. The state-of-the-art adapters include (i) prompt-tuning adaption models such as LLaMA-Adapter (Zhang et al., 2023b), Prefix-Tuning (Li and Liang, 2021), P-tuning (Liu et al., 2021b), and Prompt Tuning (Lester et al., 2021); (ii) low-rank parameter adaption models such as LoRA (Hu et al., 2021) and AdaLoRa (Zhang et al., 2023a). While the adapters help significantly reduce the computational cost and adapt LLMs faster for various downstream tasks, they may still suffer from hallucination problems and generate factually incorrect content, when the pre-trained knowledge is not well generalized to the new specific tasks. This can limit the application of adapted LLMs in some downstream scenarios where domain-specific or personalized knowledge is required, such as medical diagnosis (Varshney et al., 2023), social networks (Li et al., 2022), and personalized virtual assistant (Sun et al., 2022).

To address this challenge, additional external knowledge bases and knowledge retrieval mechanisms are required to enhance the adaption of LLMs. Knowledge graphs (KGs) have enormous potentials to encapsulate and condense rich structured and relational information that textual data inherently lacks (Schneider et al., 2022). In addition, with a domain-specific knowledge graph as additional input, the LLM can be trained to leverage domain-specific knowledge and relieve hallucination problems, especially for adaption methods that only update a limited scale of parameters. Many previous works have shown the effectiveness of integrating KGs into the *pre-training* (Zhang et al., 2019; Shen et al., 2020; Zhang et al., 2020; Wang et al., 2021) or *inference* (Baek et al., 2023; Sun et al., 2021; Zhang et al., 2021) of LLM to enhancing various NLP tasks.

However, limited work has effectively integrated LLMs with KGs for *parameter-efficient adaption*. The CKGA (Lu et al., 2023) model has explored leveraging pre-trained knowledge graph embedding (KGE) to adapt BERT (Devlin et al., 2018), but it still requires an additional training objective of link prediction for graph convolutional networks (GCNs), and the LLMs cannot directly sense the structure of KGs. In this paper, we propose the **Structure-Aware Adapter (SAA)** for LLMs to discerningly attend to the structure of knowledge graphs at a granular level. The framework of the SAA model is shown in Figure 1. We first ground and match the concepts, and retrieve the knowledge subgraphs for input sequences. Then, we propose (i) the *structure-aware relational attention* for the pre-trained LLM to attend to an external knowledge graph. The proposed mechanism has a hierarchical attention strategy that attends to the source nodes in the first level and then attends to the relations and target nodes using relational attention in the second level. This technique allows the LLM to engage with the pivotal knowledge at a more intricate granularity while neglecting the redundant information. (ii) The *path-of-thought (PoT) prompting* method is also proposed to retrieve and integrate the reasoning path from the knowledge graph to enforce the training of proposed relational attention to utilize the information from the knowledge graph.

We evaluate the proposed SAA model in two public question-answering benchmark datasets, CommonSenseQA (CSQA) (Talmor et al., 2018) and OpenBookQA (OBQA) (Mihaylov et al., 2018). We compare the proposed model with state-of-the-art LLM adapter models, as well as their extensions which incorporate pre-trained knowledge graph embedding (KGE) or knowledge graph triplets. We train the adapter models over LLaMA-7B (Touvron et al., 2023) and LLaMA-3B (Geng and Liu, 2023) and repeat the experiments for 5 times to report the average question-answering accuracy and standard deviation. The evaluation result shows that the proposed SAA model outperforms the state-of-the-art LLM adapters by 4.1%-15.9% and 1.4%-17.6% in question-answering accuracy of CSQA and OBQA datasets for LLaMA-7B. Ablation studies also show the effectiveness of the proposed structure-aware relational attention and path-of-thought prompting modules.

2 Structure-Aware Adapter

In this section, we introduce the formulation of the tasks, the proposed structure-aware relational attention technique, and path-of-thought prompt. While the proposed method can be generalized to many large language models and tasks, in this section we focus on the decoder-based language models and the question-answering task for the brevity.

2.1 Preliminaries and Formulation

We model the adaption objective as the causal language modeling for the decoder-based language models such as LLaMA (Touvron et al., 2023). The causal language modeling involves autoregressively predicting the next token in a sequence given the previous tokens. Assume the tokens in the input sequence of length n is denoted as $t_1, t_2, \cdots, t_n$, the objective is formulated as,

$$p(t_i|t_1, \cdots, t_{i-1}) = \frac{\exp(\phi(t_i, t_1, \cdots, t_{i-1}))}{\sum_t \exp(\phi(t, t_1, \cdots, t_{i-1}))}, \tag{1}$$

where $\phi(t, t_1, t_2, \cdots, t_{i-1})$ is a scoring function or model that computes the compatibility between the context and the candidate token t. Most natural language processing tasks can be modeled as an autoregressive text generation task with the causal language modeling objective and a prompt incorporating the original input and contexts. For example, we model the question-answering task with a prompt shown in Figure 2. The question-answering task provides the question context and choices as input, requiring the model the predict the correct choice. We use $T_q = \{t_q^1, t_q^2, \cdots, t_q^n\}$ to denote the question tokens and $T_c = \{t_c^1, t_c^2, \cdots, t_c^n\}$ for the choice tokens. The sequence after prompting is denoted as $T = prompt(T_q, T_c) = \{t_1, t_2, \cdots, t_n\}$.

In our task, the model receives an additional knowledge graph G as input. We assume the knowledge graph is a heterogeneous directed graph. This formulation can be generalized to most existing knowledge graphs or structured data. Assume there are N nodes and R relations. The adjacency matrix can be denoted as $\mathbf{A} \in \mathbb{Z}_2^{N\times N\times R}$. $\mathbf{A}_{i,j}^k = 1$ rep-

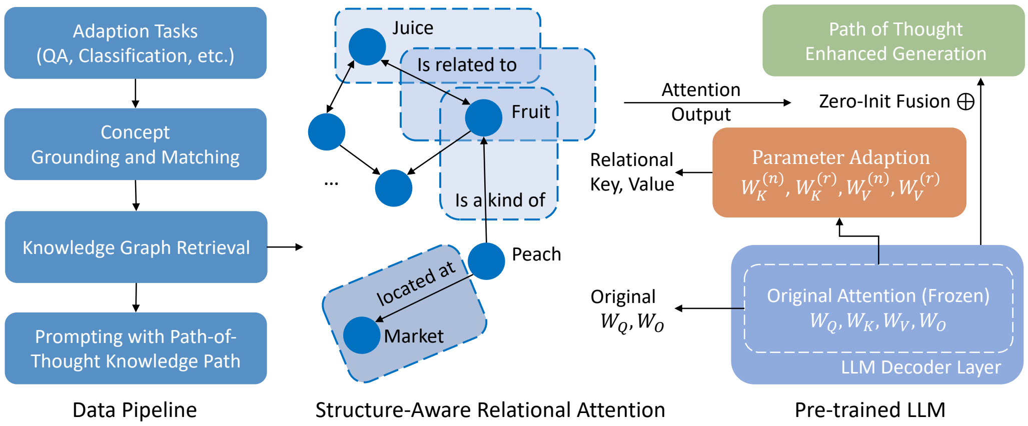

Figure 1: The framework of the proposed structure-aware adapter (SAA). The SAA model freezes the original attention weights and introduces parameter-efficient adaption to produce weights for node and relation, respectively (in orange). Hierarchical relational attention is further proposed to directly allow the LLM attends to the graph structure in Figure 3 and Figure 4. A zero-init fusion is applied to integrate the outputs. The path-of-thought prompting for adaption training is also proposed to enhance the utilization of the retrieved knowledge graph.

> Given the following question, pick the best answer from given choices.
>
> Question: The only baggage the woman checked was a drawstring bag, where was she heading with it?
>
> Choices:
> (A) garbage can (B) military (C) jewelry store (D) safe (E) airport
>
> Answer: **(E) airport**
>
> **Contexts: drawstring is part of drawstring bag, drawstring bag is at location of airport. baggage is at location of airport**

Figure 2: Example of an induced path-of-thought prompt in CSQA training dataset. During the inference in the test or validation set, the blue sentences are the expected generation. The sentence after "*Contexts:*" is the path-of-thought path retrieved from KG.

resents there is an edge between the i-th node and j-th node with k-th relation. In knowledge graphs, the feature of a node or a relation is represented by the representations denoted as x and r, respectively. Practically, the model retrieves subgraphs from the original full knowledge graph for each data sample, containing the related concepts, k-hop neighbors, and the respective relations. We denote the subgraphs with the same notation as illustrated above.

We focus on the adaption-based fine-tuning for LLMs, which freezes the original parameters (denoted as Φ) of LLMs pre-trained on the large-scale textual corpora. While the gradient computation via Φ is still required, there is no update on the original parameters. In our model, the adaption-based fine-tuning model only introduces a small scale of new parameters (denoted as ϕ^{Δ}, $|\phi^{\Delta}| \ll |\Phi|$). ϕ^{Δ} can be represented as either parameter tuning for pre-trained weight matrices like LoRA or prompt embedding like LLaMA-Adapter. The proposed structure-aware adapter tries to combine the advantage of both, while efficiently incorporate the knowledge from non-textual structured data and generalize to downstream scenarios.

2.2 Structure-Aware Relational Attention

Parameter-Efficient External Node Attention (Level-1) Typically, the self-attention layer l includes weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$, and optionally $\mathbf{W}_O$, for computing the queries, keys, values, and output mapping, respectively. We have $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O \in \mathbb{R}^{d\times d}$ where d is the dimension of LLM hidden states. In the proposed Structure-Aware Relational Attention (SARA) model, we adapt $\mathbf{W}_K$, $\mathbf{W}_V$ with low-rank adaption as LoRA (Hu et al., 2021) to produce the weight matrices for nodes (n) and relations (r) for the external attention on KG,

$$\begin{aligned}\mathbf{W}^{(n)}_{K,V} &= \mathbf{W}_{K,V} + \mathbf{P}^{(n)}_{K,V}(\mathbf{Q}^{(n)}_{K,V})^{\top}\\ \mathbf{W}^{(r)}_{K,V} &= \mathbf{W}_{K,V} + \mathbf{P}^{(r)}_{K,V}(\mathbf{Q}^{(r)}_{K,V})^{\top},\end{aligned} \tag{2}$$

where the $\mathbf{P} \in \mathbb{R}^{d\times z}$ and $\mathbf{Q} \in \mathbb{R}^{z\times d}$ are low-rank decomposition matrices designed to adjust the original LLM weight matrices. z is the rank and we have $z \ll d$. Therefore, the matrix multiplication of $\mathbf{PQ}^{\top}$ contains much fewer parameters compared with $\mathbf{W}$.

Since the knowledge graph provides text descriptions for all the nodes and relations, we compute the node embeddings $\mathbf{x}$ and relation embedding $\mathbf{r}$ of KG using the text descriptions. We apply the same tokenization as LLM and use the output of the embedding layer to compute $\mathbf{x}$ and $\mathbf{r}$. For those nodes and relations with $k > 1$ tokens, we

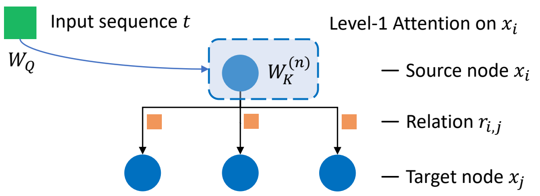

Figure 3: The Level-1 attention of the proposed SARA which attends to the source nodes with external attention.

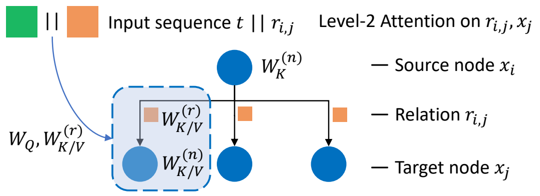

Figure 4: The Level-2 attention of the proposed SARA which attends to the relations and target nodes using relational attention.

take the average embedding, i.e. $\mathbf{x} = \frac{1}{|k|}\sum_i^k \mathbf{x}_i$, $\mathbf{r} = \frac{1}{|k|}\sum_i^k \mathbf{r}_i$. With the computed $\mathbf{x}$, $\mathbf{r}$, and adjacency matrix $\mathbf{A}$, we design a 2-level hierarchical relational attention for the knowledge graph.

The framework of Level-1 attention is shown in Figure 3. The intuition is to conduct external attention to query relevant knowledge from the structured knowledge graph. For the computing of the query, we use the original weight $\mathbf{W}_Q$. For the keys and values, we apply adapted matrices $\mathbf{W}_K^{(n)}$, $\mathbf{W}_K^{(r)}$, $\mathbf{W}_V^{(n)}$, $\mathbf{W}_V^{(r)}$ as introduced in Equation 2. In the first level of the hierarchical attention, as is shown in Figure 3, we first compute the attention score $\sigma^{(1)}$ between the input sequence $\mathbf{T}$ and all the *source nodes* $\mathbf{x}_i$, which can be formulated as,

$$\sigma^{(1)}(\mathbf{T}, \mathbf{x}_i) = Softmax\left(\frac{(\mathbf{T}\mathbf{W}_Q)(\mathbf{x}_i\mathbf{W}_K^{(n)})^\top}{\sqrt{d}}\right), \tag{3}$$

where $\mathbf{W}_Q$ is the pre-trained frozen weight from LLM. $\mathbf{W}_K^{(n)}$ is the trainable weight of key for nodes. d is the dimension of hidden states. Note that while exploited, the formulation of *multi-head attention* is omitted here for brevity.

Relational Attention (Level-2) The relational attention, or graph transformer (Diao and Loynd, 2022) was initially proposed to improve the reasoning of graph representation learning tasks. Inspired by relational attention, we propose the hierarchical relational attention in Figure 4 for adapting LLMs to incorporate the relational information from knowledge graphs. The idea of hierarchical relational attention is to concatenate the node representations with the relation representations, as well as concatenate the weight matrices. Then we compute the attention on a more fine-grained level. More specifically, for each edge triplet $(\mathbf{x}_i, \mathbf{r}_{i,j}, \mathbf{x}_j)$ we have

$$\begin{aligned}
\mathbf{q}_{i,j} &= [\mathbf{T}; \mathbf{r}_{i,j}][\mathbf{W}_Q^\top; \mathbf{W}_Q^\top]^\top \\
\mathbf{k}_{i,j} &= [\mathbf{x}_j; \mathbf{r}_{i,j}][(\mathbf{W}_K^{(n)})^\top; (\mathbf{W}_K^{(r)})^\top]^\top \\
\mathbf{v}_{i,j} &= [\mathbf{x}_j; \mathbf{r}_{i,j}][(\mathbf{W}_V^{(n)})^\top; (\mathbf{W}_V^{(r)})^\top]^\top,
\end{aligned} \tag{4}$$

where $\mathbf{T}$ is the tokens of the input sequence. $\mathbf{r}_{i,j}$ is the *relation* between node i and j. $\mathbf{x}_j$ is the *target node*. $\mathbf{W}_Q$ is the original query weight matrix in the LLM attention layer. The computation can also be simplified as

$$\begin{aligned}
\mathbf{q}_{i,j} &= \mathbf{T}\mathbf{W}_Q + \mathbf{r}_{i,j}\mathbf{W}_Q \\
\mathbf{k}_{i,j} &= \mathbf{x}_j\mathbf{W}_K^{(n)} + \mathbf{r}_{i,j}\mathbf{W}_K^{(r)} \\
\mathbf{v}_{i,j} &= \mathbf{x}_j\mathbf{W}_V^{(n)} + \mathbf{r}_{i,j}\mathbf{W}_V^{(r)}.
\end{aligned} \tag{5}$$

With the above definition, the second-level attention weight $\alpha^{(2)}$ and attention score $\sigma^{(2)}$ can be computed as

$$\begin{aligned}
\alpha_{i,j}^{(2)}(\mathbf{T}, \mathbf{r}_{i,j}, \mathbf{x}_j) &= \frac{\mathbf{q}_{i,j}(\mathbf{T}, \mathbf{r}_{i,j})\mathbf{k}_{i,j}^\top(\mathbf{r}_{i,j}, \mathbf{x}_j)}{\sqrt{d}} \\
\sigma_{i,j}^{(2)} &= \frac{\exp(\alpha_{i,j}^{(2)})}{\sum_{\mu \in \mathcal{N}_i} \exp(\alpha_{i,\mu}^{(2)})},
\end{aligned} \tag{6}$$

where $\mathcal{N}_i = \{\mathbf{x}_j | \mathbf{A}_{i,j}^k \neq 0\}$ represents all the neighbors of node i w.r.t. any relation $\mathbf{r}^k$.

Zero-Init Fusion of 2-level Attentions Finally, we compute the hierarchical attention score by multiplying the scores of two levels with adjacency matrix of subgraph.

$$\sigma_{i,j}^{(KG)} = \sum_{i,j} \sigma_i^{(1)} \mathbf{A}_{i,j} \sigma_{i,j}^{(2)} \tag{7}$$

We fuse the output of SARA with the original output of LLM with a zero-init gate (Zhang et al., 2023b),

$$\mathbf{h}_\mathbf{t}^\mathbf{l} = \mathbf{W}_O([\sigma^{(KG)} \cdot g;\ \sigma^{(LLM)}] \cdot [\mathbf{V}^{(KG)}; \mathbf{V}^{(LLM)}]), \tag{8}$$

where g is the zero-init gate and the semicolon represents concatenation. $\mathbf{W}_O$ is the output mapping in the original LLM attention. $\sigma^{(LLM)}$ is the original softmax attention score for the input sequence T. $\mathbf{V}$ is the value matrix in Equation 5, we have

$\mathbf{v}_{i,j} \in \mathbf{V}$. $\mathbf{h}_{\mathrm{t}}^{\mathrm{l}}$ is the output hidden state for the token t at layer l.

The proposed SARA can be applied to adapt multiple attention layers of original LLM attention, practically the last L attention layers. With multiple adapted layers fused with the proposed KG attention, the LLM can learn to attend to complex graph structures. Compared with the existing method which directly attends to textual triplets of trained KG representations, the proposed mechanism adapts LLM to attend to the graph structures in a more fine-grained manner. In addition, since the knowledge graph usually contains a lot of redundant information (Akrami et al., 2020), the proposed relational attention enables the LLM to selectively retrieve essential information and neglect the redundant or unrelated nodes and relations.

2.3 Enhance Knowledge Reasoning with Path-of-Thought Prompt

In the previous section, we have introduced the structure-aware relational attention, which retrieves and fuses the fine-grained knowledge output from the knowledge graph (KG). While it provides the mechanism for LLM to retrieve additional knowledge, it's not guaranteed whether the model can learn to utilize it during adaption (especially with fewer trainable parameters). One straight-forward idea is to pre-train the LLM with the KG module on additional large textual corpora (Yasunaga et al., 2022), which will result in heavy computation cost. In this paper, we propose a knowledge-induced path-of-thought (PoT) prompt to enforce the utilization of KGs.

The idea of the proposed PoT prompt is inspired by the chain-of-thought prompt (Wei et al., 2022), which was proposed to enhance the *zero-shot inference* of LLM, where several examples with manually labeled chain-of-thought contexts are provided before we input the actual sequence into the LLM. In our case, instead of prompting at inference time, we retrieve and integrate PoT in the training prompt to *enhance the adaption training*. More specifically, we design an algorithm to retrieve the reasoning path between pairs of matched concepts in KG. We denote the concepts from the question as $c_q \in C_q$, the choice concepts as $c_p \in C_p$, and the concepts of correct choice (answer) as $\hat{c}_p \in \hat{C}_p$. Then, for every pair of concepts from $(c_q, c_p) \in C_q \times \hat{C}_p$, we compute the shortest paths between them using Dijkstra algorithm (Cormen, 2001). Finally, we concatenate the text of nodes and relations along the shortest paths to form the final prompt, together with the question, choices, and answer. One example of computed PoT prompts is shown in Figure 2.

This technique is different from the previous works transforming the KG triplets or knowledge contexts into texts as additional input (Wang et al., 2021; Baek et al., 2023). In the proposed PoT prompting, the retrieved reasoning path works as the additional learning objective instead of input. The proposed prompting actually enforces the adapted LLM to (i) generate the answer prediction, and (ii) generate the context of the reasoning path. This additional objective, therefore, enhances the model to utilize the information from KG.

3 Experiments

In this paper, we focus on the question-answering task which emphasizes the knowledge reasoning of LLMs. The proposed models and baselines are evaluated on two public question-answering benchmark datasets, including CommonSenseQA (CSQA) (Talmor et al., 2018) and OpenbookQA (OBQA) (Mihaylov et al., 2018) (see Appendix A for details). We compare our algorithm with two state-of-the-art LLM adapters, LoRA (Hu et al., 2021) and LLaMA Adapter (Zhang et al., 2023b), as well as their knowledge-enhanced variants. The baseline details and hyper-parameters are introduced in Appendix A.3. In the experiments, we use two pre-trained LLMs as the base models for adaption: (i) LLaMA-7B[1], a pre-trained LLaMA model (Touvron et al., 2023) by Meta AI containing 7-billion parameters. (ii) LLaMA-3B (Geng and Liu, 2023) , a smaller pre-trained LLaMA model by OpenLM Research (Geng and Liu, 2023), with 3-billion parameters.

3.1 Knowledge Graph Retrieval

For each query, we retrieve a knowledge sub-graph based on the heuristic concept match (Yasunaga et al., 2022). We extract the concepts from questions and choices after lemmatization and match them with the concept nodes in KG, based on the with the *en_core_web_sm* pipeline in the spaCy library[2]. The average numbers of matched concepts in CSQA and OBQA datasets are 14.04 and 14.59. Based on the matched concepts, we further retrieve and include top-100 *2-hop neighbors*, sorted and filtered based on semantic similarity score.

[1] https://github.com/facebookresearch/llama
[2] https://spacy.io/models/en#en_core_web_sm

Table 1: Evaluation result of question-answering accuracy in CSQA and OBQA datasets. We report the average accuracy and the respective standard deviation with 5 random seeds. The first two columns are the results of LLaMA-7B pre-trained LLM and the last two columns are the result of a relatively smaller LLaMA-3B model. The proposed structure-aware achieves the highest average accuracy.

	LLaMA-7B		LLaMA-3B	
Model Name	CSQA	OBQA	CSQA	OBQA
Zero-Shot	0.3073	0.2780	0.1957	0.2760
LLAMA-Adapter	$0.6124^{\pm 0.0119}$	$57.08^{\pm 0.0139}$	$0.6169^{\pm 0.0112}$	$0.4480^{\pm 0.0772}$
LLAMA-Adapter + KGE	$0.5920^{\pm 0.0163}$	$0.5416^{\pm 0.0190}$	$0.2069^{\pm 0.0111}$	$0.3016^{\pm 0.0099}$
LLAMA-Adapter + KG Triplets	$0.5951^{\pm 0.0070}$	$0.6368^{\pm 0.0129}$	$0.3053^{\pm 0.1265}$	$0.5172^{\pm 0.0095}$
LoRA	$0.6822^{\pm 0.0110}$	$0.6624^{\pm 0.0144}$	$0.5297^{\pm 0.1789}$	$0.6028^{\pm 0.0212}$
LoRA + KGE	$0.6943^{\pm 0.0050}$	$0.6652^{\pm 0.0088}$	$0.6401^{\pm 0.0090}$	$0.5928^{\pm 0.0145}$
LoRA + KG triplets	$0.6644^{\pm 0.0050}$	$0.6696^{\pm 0.0112}$	$0.3735^{\pm 0.0925}$	$0.6048^{\pm 0.0119}$
LLAMA-Adapter + LoRA	$0.6994^{\pm 0.0032}$	$0.6396^{\pm 0.0067}$	$0.6624^{\pm 0.0102}$	$0.6100^{\pm 0.0163}$
SSA (Ours)	$\mathbf{0.7100}^{\pm 0.0058}$	$\mathbf{0.6715}^{\pm 0.0042}$	$\mathbf{0.6650}^{\pm 0.0115}$	$\mathbf{0.6140}^{\pm 0.0171}$

3.2 Evaluation Metrics

We provide the model a prompt containing the questions, choices, and optionally path-of-thought contexts as is shown in Figure 2. During inference, we have the adapted LLM to generate the next 5 tokens after the "*Answer:*" in the prompt. We use the multiple choice symbol binding (MCSB) method (Robinson et al., 2022) to compute the prediction label. More specifically, we find the choice token (e.g. "(A)") with the maximum number of appearances and use it as the model prediction. Finally, we report the accuracy of question answering as the evaluation metric. We repeat all the experiments for 5 times and report the average accuracy and the standard deviation.

3.3 Experimental results

We compare our proposed structure-aware adapter model with the baselines in both the CSQA and OBQA datasets. The learning rate is set as 0.0003. We apply the proposed adapter to the last 20 layers of LLM attention, the same as the settings of baselines. The low-rank dimension and alpha are set as $z = 2$ and $\alpha = 8$ for the adaption of weight matrices. In our model with the path-of-thought prompting, we limit the maximum length of the shortest path of thought as 50 tokens in the training prompt. The experimental results of the proposed structure-aware adapter and the baselines are shown in Table 1. The proposed structure-aware adapter outperforms the state-of-the-art baselines. When adapting on LLaMA-7B in the CSQA dataset, our model achieves 15.9% and 4.1% higher accuracy than LLaMA-Adapter and LoRA, respectively. When adapting on LLaMA-7B in the OBQA dataset, our model achieves 17.6% and 1.4% higher accuracy than LLaMA-Adapter and LoRA.

We also compare the proposed model with several extensions of LLaMA-Adapter and LoRA enhanced with pre-trained knowledge graph embedding (KGE) or textual KG triplets. In the KGE extensions, we integrate the pre-trained KGE of the matched concepts and related neighboring concepts by applying a linear mapping, and then adding to the prompt embedding of LLaMA-Adapter or applying LoRA-adapted external attention on KGE. The pre-trained KGE improves the performance of LoRA in most cases of datasets and PLMs. This is because the KGE is pre-trained to include the information of relations and adjacency concepts, which serve as external knowledge for LoRA to answer questions. The KG triplets also help the adaption models in the question-answering task, especially for LLaMA-Adapter on the OBQA dataset. However, integrating either the pre-trained KGE or the textual KG triplets is not optimal. While already being filtered with some rule-based pre-processing, there is still a lot of redundant information stored in KGE as well as KG triplets. The methods incorporating KGE and KG triplets do not allow the LLM to sense the relational structure and selectively retrieved the key information. The proposed structure-aware relational attention naturally allows LLM to attend to the relational structure of KG at a more fine-grained level, which enhances the ability of the proposed module to denoise the redundant information and achieve higher average accuracy.

In addition, we study the effectiveness of the proposed model and baselines on a LLaMA-3B model, which contains fewer parameters and is pre-trained on smaller and unofficial corpora, and fewer subtasks. The adaption in LLaMA-3B is more challenging because it contains much less pre-trained knowledge and, meanwhile, it's more difficult to enforce it to leverage the external knowledge from KG. In this case, we observe the adaption training of many baselines becomes unstable and sometimes fails to converge. This leads to lower average accuracy scores and high standard deviation. The instability of training is especially significant after incorporating the KGE and KG triplets. While the proposed structure-aware adapter also leverages external knowledge, we in addition propose the path-of-thought prompting to enforce the model to attend to KGs and therefore stabilize the training. Compared with baselines, the training of the proposed model is more stable and we do not observe a collapse of convergence.

3.4 Efficiency Analysis

We report the number of trainable parameters, the memory cost, as well as the average training time on CSQA and OBQA datasets in Table 2. The proposed SSA model uses a comparable number of trainable parameters ($0.048‰$) as LoRA and LoRA-Triplets ($0.024‰$), and much fewer parameters than other baselines ($\geq 0.122‰$). In addition, while integrated with attention on KG, we do not observe a significant increase in training time for the proposed model (11 hours) compared with LoRA (8 hours) and LLAMA-Adapter (11 hours).

Table 2: The efficiency comparison of numbers of trainable parameters, memory, and average training time.

Model Name	#Trainable Param.	Mem	Time
Zero-Shot	0.00M (0.000‰)	-	-
L.Ada.	0.82M (0.122‰)	3.56M	11 hrs
L.Ada.+KGE	5.01M (0.744‰)	21.23M	13 hrs
L.Ada.+Triplets	0.82M (0.122‰)	3.62M	20 hrs
LoRA	0.16M (0.024‰)	0.62M	8 hrs
LoRA+KGE	4.36M (0.647‰)	17.34M	11 hrs
LoRA+Triplets	0.16M (0.024‰)	0.65M	18 hrs
L.Ada.+LoRA	0.98M (0.146‰)	4.21M	13 hrs
SSA (Ours)	0.32M (0.048‰)	1.22M	11 hrs

3.5 Ablation Study

We conducted ablation studies to evaluate the effectiveness of the proposed modules of the structure-aware adapter. We removed or modified the proposed modules to form the following ablation experiments: (i) **Without Relational Attention**: We remove the proposed structure-aware relational attention and use a typical attention mechanism to attend to the average embeddings of relations $r_{i,j}$ and target nodes x_j. (ii) **With Node Attention**: We simplify the proposed method to attend to only the nodes x_j of matched or related concepts in the retrieved knowledge subgraph. (iii) **Without Path-of-Thought**: The proposed SAA model without the path-of-thought (PoT) prompting, where we train the model without the "*Contexts:*" part.

Table 3: Ablation study of the Structure-Aware Adapter, after removing Relational Attention, replacing with Node Attention, or removing the path-of-thought (PoT).

Model Name	CSQA	OBQA
W/o Rel. Att.	$0.6968^{\pm 0.0074}$	$0.6608^{\pm 0.0231}$
W/ Node Att.	$0.6915^{\pm 0.0089}$	$0.6542^{\pm 0.0068}$
W/o PoT	$0.7076^{\pm 0.0096}$	$0.6674^{\pm 0.0093}$
SSA (Ours)	$\mathbf{0.7100}^{\pm 0.0058}$	$\mathbf{0.6715}^{\pm 0.0042}$

The experimental result is shown in Table 3. By removing the proposed hierarchical relational attention for the knowledge graph, the accuracy decreases for 1.32% and 1.07% respectively in CSQA and OBQA datasets, which illustrates the effectiveness of the relational attention. A further simplified ablation model is the one with node attention, which ignores the relation features and only attends to the matched concepts or their neighbors. We also observe a decrease of 1.85% and 1.73% in both datasets. While the neighbors of matched concepts also provide contexts for solving the question-answering task, neglecting the relations and graph structure leads to a significant decrease in the accuracy metrics. Finally, we also study the model without the proposed path-of-thought prompting. After removing PoT, there is a observed accuracy reduction in both datasets and the standard deviation also increases. This shows the benefit of applying the path-of-thought prompting in enhancing knowledge utilization and training stabilization.

4 Related Works

Large Language Model Adaption The adaption-based model fine-tuning, or parameter-efficient fine-tuning (PEFT) for large language models (Mangrulkar et al., 2022a) freezes the parameters of the initial pre-trained large language models and only introduces a small number of trainable parameters to save computational costs and preserve the pre-trained linguistic knowledge. The existing work has explored the prompt-tuning adaption methods (Zhang et al., 2023b; Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021b,a; Qin and Eisner, 2021) and parameter weight adaption methods (Hu et al., 2021; Zhang et al., 2023a; Hedegaard et al., 2022). One representative work of prompt-tuning is the LLaMA-Adapter (Zhang et al., 2023b), which attaches the embedding of the trainable adaption prompts as a prefix along with the input sequence and introduces a zero-init fusion mechanism to integrate the output of adaption prompt to the language model. The LoRA model (Hu et al., 2021) is a parameter weight adaption model proven to be effective in adapting the model for various generative tasks, the performance of which is close to the full fine-tuning of original large language models (LLMs). While the existing adaption models show promising performance in adapting PLMs to various downstream tasks, these methods may still suffer from hallucination problems and generate factually incorrect content due to limited trainable parameters for domain transferring. The adaption models still rely on the knowledge from the textual pre-training corpora and cannot utilize some external knowledge, which limits their application of domain-specific scenarios. In this paper, we propose a structure-aware adapter for PLMs that utilize the structured data to enhance the downstream generative tasks.

Knowledge Graph Enhanced Language Modeling The knowledge graph, such as ConceptNet (Speer et al., 2017), Wikidata (Vrandečić and Krötzsch, 2014), is a structured knowledge base that has been proven to be effective in improving the performance of LLM on various natural language processing tasks (Pan et al., 2023). Many other graphs such as social graphs and entity interaction logs can also be represented as the knowledge graph to enhance LLMs (Li et al., 2022; Chang et al., 2021; El-Kishky et al., 2022). The exiting researches have explored utilizing the knowledge graph for improving the LLM **pre-training** such as ERNIE (Zhang et al., 2019), GLM (Shen et al., 2020), E-BERT (Zhang et al., 2020) KEPLER (Wang et al., 2021), K-BERT (Liu et al., 2020), **inferences** such as QA-GNN (Sun et al., 2021), GreaseLM (Zhang et al., 2021), KGLM (Logan IV et al., 2019), DRAGON (Yasunaga et al., 2022), and KAPING (Baek et al., 2023).

However, *limited research* has focused on enhancing the **adaption** of LLM with knowledge graph, while the adaption methods have become more interesting due to the ever-increasing scale of PLM parameters. The CKGA (Lu et al., 2023) model has explored leveraging pre-trained knowledge graph embedding to adapt BERT (Devlin et al., 2018), but it still requires an additional training objective of link prediction for graph convolutional networks (GCNs) and the LLM cannot directly attend to the structure of KGs. The existing research has explored the mechanisms to integrate the information from the knowledge graph. Some of the existing methods *transforms the knowledge graph triplets* like ERNIE (Zhang et al., 2019), SKILL (Moiseev et al., 2022), and KAPING (Baek et al., 2023), or retrieved knowledge contexts such as KEPLER (Wang et al., 2021) into text as additional input. However, the additional textual input usually cannot well represent the complex graph structure and may introduce additional noise. Some related works focus on generating KG entity embeddings as additional input for the language models such as KI-BERT (Faldu et al., 2021) and NTULM (Li et al., 2022). The other works exploit joint training of link prediction and masked language modeling (MLM) objectives for the pretraining of LLM, such as DRAGON (Yasunaga et al., 2022) and KEPLER (Wang et al., 2021). However, these methods usually use a single fusion bottleneck between LLM and the graph module and usually train additional graph neural networks (GNN) to encode the node embeddings, Therefore, the LLMs cannot directly attend to the structure of KG. On the contrary, we propose the structure-aware relational attention that allows LLMs to naturally attend to structures of the knowledge graph without bottleneck networks or additional graph learning objectives during the adaption training.

5 Conclusion

This paper proposes a structure-aware adapter for parameter-efficient fine-tuning of LLMs, leveraging structured information from knowledge graphs. We propose the hierarchical relational attention mechanism to allow LLMs to intrinsically attend to knowledge graphs at a granular level. In addition, a novel algorithm is proposed to extract the reasoning paths from knowledge graphs and derive the path-of-thought prompts to enforce the efficacy of proposed relational attention in knowledge extraction. The evaluation result in two question-answering benchmark datasets demonstrates that the proposed approach outperforms the state-of-the-art LLM adapters and their variants in QA accuracy. Ablation studies further illustrates the effectiveness of the proposed relational attention and path-of-though prompting in jointly enhancing the model's ability on QA reasoning.

Limitations

While the proposed hierarchical structure-aware relational attention designs the gradients of the external graph attention end-to-end with the adapted parameters of LLM for the text generative objective, the retrieval of the KG sub-graph is heuristic and rule-based. The rule-based sub-graph retrieval algorithms are usually robust for various datasets and tasks, however, they still face the challenge of precision-recall trade-off (either neglecting useful nodes or including redundant nodes). The empirical solution in this paper is leveraging relatively higher recall and adopting the proposed level-1 external node attention to denoise redundant nodes. However, another possible direction is adopting a neural retrieval algorithm and integrating end-to-end with the whole framework, which can dynamically update the retrieval results with the LLM fine-tuning objective, and may bring additional benefit for the knowledge-enhanced generation task.

Ethics

The datasets, knowledge graph databases, and pretrained large language models utilized in this paper were publicly available and open-sourced. All experiments involving these resources were conducted in compliance with their respective permissive licenses. This study did not involve any additional human-engaged annotation, investigation, or survey.

References

Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1995–2010.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

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

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. 2021. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. *arXiv preprint arXiv:2105.05457*.

Thomas H Cormen. 2001. Section 24.3: Dijkstra's algorithm. *Introduction to algorithms*, pages 595–601.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Cameron Diao and Ricky Loynd. 2022. Relational attention: Generalizing transformers for graph-structured tasks. *arXiv preprint arXiv:2210.05062*.

Ahmed El-Kishky, Thomas Markovich, Serim Park, Chetan Verma, Baekjin Kim, Ramy Eskander, Yury Malkov, Frank Portman, Sofía Samaniego, Ying Xiao, et al. 2022. Twhin: Embedding the twitter heterogeneous information network for personalized recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2842–2850.

Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akbari. 2021. Ki-bert: Infusing knowledge context for better language and domain understanding. *arXiv preprint arXiv:2104.08145*.

Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, Rakesh Chada, Amit Chauhan, Luoxin Chen, Anurag Dwarakanath, Satyam Dwivedi, Turan Gojayev, Karthik Gopalakrishnan, Thomas Gueudre, Dilek Hakkani-Tur, Wael Hamza, Jonathan J. Hüser, Kevin Martin Jose, Haidar Khan, Beiye Liu, Jianhua Lu, Alessandro Manzotti, Pradeep Natarajan, Karolina Owczarzak, Gokmen Oz, Enrico Palumbo, Charith Peris, Chandana Satya Prakash, Stephen Rawls, Andy Rosenbaum, Anjali Shenoy, Saleh Soltan, Mukund Harakere Sridhar, Lizhen Tan, Fabian Triefenbach, Pan Wei, Haiyang Yu, Shuai Zheng, Gokhan Tur, and Prem Natarajan. 2022. Alexa teacher model: Pretraining and distilling multi-billion-parameter encoders for natural language understanding systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2893–2902, New York, NY, USA. Association for Computing Machinery.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`.

Lukas Hedegaard, Aman Alok, Juby Jose, and Alexandros Iosifidis. 2022. Structured pruning adapters. *arXiv preprint arXiv:2211.10155*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Jinning Li, Shubhanshu Mishra, Ahmed El-Kishky, Sneha Mehta, and Vivek Kulkarni. 2022. Ntulm: Enriching social media text representations with non-textual units. *arXiv preprint arXiv:2210.16586*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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

Robert L Logan IV, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge-graphs for fact-aware language modeling. *arXiv preprint arXiv:1906.07241*.

Guojun Lu, Haibo Yu, Zehao Yan, and Yun Xue. 2023. Commonsense knowledge graph-based adapter for aspect-level sentiment classification. *Neurocomputing*, 534:67–76.

SOURAB Mangrulkar, S Gugger, L Debut, Y Belkada, and S Paul. 2022a. Peft: State-of-the-art parameter-efficient fine-tuning methods.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022b. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. Skill: structured knowledge infusion for large language models. *arXiv preprint arXiv:2205.08184*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. *arXiv preprint arXiv:2210.00105*.

Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. 2020. Exploiting structured knowledge in text via graph-guided representation learning. *arXiv preprint arXiv:2004.14224*.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Prem Natarajan. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2021. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. *arXiv preprint arXiv:2112.02732*.

Zhongkai Sun, Sixing Lu, Chengyuan Ma, Xiaohu Liu, and Chenlei Guo. 2022. Query expansion and entity weighting for query reformulation retrieval in voice assistant systems. *arXiv preprint arXiv:2202.13869*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Deeksha Varshney, Aizan Zafar, Niranshu Kumar Behera, and Asif Ekbal. 2023. Knowledge graph assisted end-to-end medical dialog generation. *Artificial Intelligence in Medicine*, 139:102535.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Denghui Zhang, Zixuan Yuan, Yanchi Liu, Fuzhen Zhuang, Haifeng Chen, and Hui Xiong. 2020. E-bert: A phrase and product knowledge enhanced language model for e-commerce. *arXiv preprint arXiv:2009.02835*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023a. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. Greaselm: Graph reasoning enhanced language models. In *International conference on learning representations*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

A Appendix

A.1 Dataset Details

CommonSenseQA (CSQA): The CSQA dataset (Talmor et al., 2018) is a 5-choice question answering benchmark which requires different types of commonsense knowledge to predict the correct answers. This dataset includes 9741 samples in the train set, 1221 in the validation set, and 1140 in the test set. Since the label of the test set in CSQA is not publicly available, we report the evaluation result in the validation set.

OpenbookQA (OBQA): OBQA (Mihaylov et al., 2018) is another advanced 4-choice question-answering dataset, probing a deeper understanding of the topic and the language it is expressed in. While the OBQA dataset also provides salient facts summarized as an open book, it is not used in our experiments for a fair comparison. The OBQA dataset includes 4957 samples for training, 500 for validation, and 500 for testing. In OBQA the label of the test set is publicly available.

A.2 Implementation and Environments

All the experiments are conducted on AWS G5 instances with 8 Nvidia A10G GPUs, 192-core CPUs, and 748GB memory. The implementation is based on Python 3.10.11 and PyTorch 2.0.0. We utilize the accelerate (Gugger et al., 2022) and deepspeed[3] libraries for distributed training.

A.3 Baselines and Hyper-parameters

Zero-Shot: We directly apply the pre-trained LLM for a generation without any fine-tuning or further adaption.

LLaMA-Adapter (Zhang et al., 2023b): The state-of-the-art prompt-embedding-based adapter for LLM. We apply LLaMA-Adapter to the last 20 attention layers with adaption prompt length equal to 10. The implementation is based on peft library (Mangrulkar et al., 2022b). All the other setting remains the same as the paper.

LLaMA-Adapter + KGE: Extension of the LLaMA-Adapter model to incorporate the pre-trained knowledge graph embedding (KGE), using the same framework of the image-incorporated extension of LLaMA-Adapter (Zhang et al., 2023b) with linear projection.

LLaMA-Adapter + KG triplets: The extension of LLaMA-Adapter model where we extract and integrate up to 100 tokens of triplets from KGs to the input.

LoRA (Hu et al., 2021): The state-of-the-art parameter adaption model for LLMs is based on trainable rank decomposition matrices. We apply LoRA to the last 20 attention layers. The learning rate is

[3] https://github.com/microsoft/DeepSpeed

set as 0.0003. The low-rank dimension and alpha are set as $z = 2$ and $\alpha = 8$. The implementation is based on peft library.

LoRA + KGE: The extension of the LoRA model to integrate the linear-mapped pre-trained KGE from ConceptNet. External attention is applied to the KGE.

LoRA + KG triplets: Extension of the LoRA model to include up to 100 tokens of triplets transformed from KGs. We integrate the triplets with the prompt.

LLaMA-Adapter + LoRA: We simultaneously apply the LLaMA-Adapter for prompt adaption and LoRA for parameter adaption, both applied to the last 20 attention layers with $z = 2$, $\alpha = 8$, and 10 adaption prompts.