Knowledgeable Parameter Efficient Tuning Network for Commonsense Question Answering

Ziwang Zhao[1] **Linmei Hu**[2*] **Hanyu Zhao**[3] **Yingxia Shao**[1] **Yequan Wang**[3]
[1]Beijing University of Posts and Telecommunications
[2] Beijing Institute of Technology [3]Beijing Academy of Artificial Intelligence
{zhaoziwang,shaoyx}@bupt.edu.cn hulinmei@bit.edu.cn
hyzhao@baai.ac.cn tshwangyequan@gmail.com

Abstract

Commonsense question answering is important for making decisions about everyday matters. Although existing commonsense question answering works based on fully fine-tuned PLMs have achieved promising results, they suffer from prohibitive computation costs as well as poor interpretability. Some works improve the PLMs by incorporating knowledge to provide certain evidence, via elaborately designed GNN modules which require expertise. In this paper, we propose a simple knowledgeable parameter efficient tuning network to couple PLMs with external knowledge for commonsense question answering. Specifically, we design a trainable parameter-sharing adapter attached to a parameter-freezing PLM to incorporate knowledge at a small cost. The adapter is equipped with both entity- and query-related knowledge via two auxiliary knowledge-related tasks (i.e., span masking and relation discrimination). To make the adapter focus on the relevant knowledge, we design gating and attention mechanisms to respectively filter and fuse the query information from the PLM. Extensive experiments on two benchmark datasets show that KPE is parameter-efficient and can effectively incorporate knowledge for improving commonsense question answering.

1 Introduction

Commonsense question answering is the process of combining observations and the basic knowledge that reflects our natural understanding of the world and human behaviors, to make presumptions about ordinary situations in our daily life (Johnson-Laird, 1980). It has emerged as an important task in natural language understanding.

Pre-trained language models (PLMs), which revolutionize many areas with superior performance, have been applied for the commonsense question answering task based on full fine-tuning as shown in Figure 1(a). For example, Lourie et al. (2021) fully fine-tuned the PLM Unicorn and achieved competitive performance on 8 commonsense benchmarks. However, they inevitably incur prohibitive computation costs as the scale of parameters increases, and are lacking in transparency and interpretability (Houlsby et al., 2019a; Lin et al., 2019).

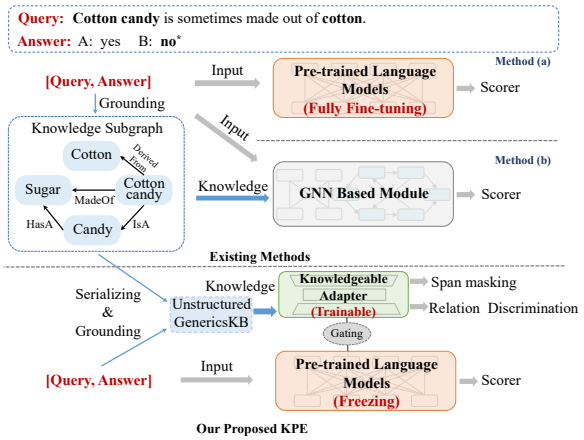

Figure 1: Comparison of existing methods and our proposed method.

Furthermore, some works couple the PLMs with knowledge to improve the interpretability of the reasoning process. As shown in Figure 1(b), they typically extract the relevant knowledge subgraphs about entities in the query and then elaborately design a graph neural network (GNN) module to perform reasoning (Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021; Sun et al., 2022). Despite the fact that they provide certain evidence for the reasoning process, it requires expertise to design effective GNN modules. Additionally, they generally consider only the structured triple knowledge about the entities in the query, while ignoring the textual knowledge about the query itself.

In this work, we propose a simple **K**nowledgeable **P**arameter **E**fficient model (KPE) for commonsense question answering. In particular, we design a parameter-sharing adapter

*Corresponding author

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9051–9063

plugin for incorporating knowledge into the frozen PLM as shown in Figure 1, which largely reduces the scale of trainable parameters. Our adapter plugin integrates both the entity- and query-related knowledge (uniformly grounding to the unstructured commonsense knowledge base GenericsKB) through two auxiliary knowledge-related tasks (i.e., span masking and relation discrimination). Additionally, to make the adapter focus on the relevant knowledge for commonsense question answering, we design gating and attention mechanisms to respectively filter and fuse the query information from the PLM. Overall, our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose a knowledgeable parameter efficient tuning network for commonsense question answering, which adopts a new parameter-sharing adapter to incorporate knowledge.
- Our designed adapter integrates both the entity- and query-related knowledge with two auxiliary knowledge-related tasks. Additionally, the gating and attention mechanisms are respectively employed to filter and fuse the query information from the PLM to make the adapter focus on relevant knowledge for commonsense question answering.
- Extensive experiments on two benchmark datasets have demonstrated that our proposed KPE can effectively incorporate knowledge for improving commonsense question answering, with a tiny computation cost.

2 Related Work

In this section, we review the related works on commonsense question answering and parameter efficient tuning.

2.1 Commonsense Question Answering

With the remarkable success of PLMs on various tasks (Liu et al., 2019; Raffel et al., 2020), some researchers propose to fully fine-tune PLMs on the commonsense question answering task. For example, Lourie et al. (2021) fully fine-tuned the PLM Unicorn on 8 commonsense benchmarks respectively, and achieved promising results. Khashabi et al. (2020) built a universal PLM for the question answering task and fully fine-tuned it on 10 factoid and commonsense QA datasets. Despite the prevalence of PLMs, fine-tuning all the parameters brings prohibitive computation costs as the scale of PLM parameters grows. Moreover, due to the lack of modules explicitly modeling knowledge, the PLMs suffer from poor transparency and interpretability. In light of this, some methods improve the PLMs with well-designed GNN modules to integrate relevant knowledge from knowledge graphs (Feng et al., 2020; Sun et al., 2022; Wang et al., 2022). For example, MHGRN (Feng et al., 2020) combines PLMs with a graph relation network to perform multi-hop reasoning on knowledge subgraphs and provides certain evidence for the reasoning process. GreaseLM (Zhang et al., 2022) and JointLK (Sun et al., 2022) introduce GNN-based modules to perform joint reasoning over both the text and knowledge subgraphs for commonsense question answering. Nevertheless, they require expertise to design an effective GNN module for encoding the knowledge subgraph. Additionally, they only consider the entity-related structured knowledge, ignoring the query-related knowledge which could be in the form of text.

Differently, in this work, we present a simple knowledgeable parameter efficient tuning network which utilizes a parameter-sharing adapter to incorporate both entity- and query-related knowledge for improving commonsense question answering.

2.2 Parameter Efficient Tuning

Since fine-tuning all the parameters of PLMs causes prohibitively expensive costs, researchers propose to fine-tune a small part of the model parameters while freezing the rest. Adapter-tuning, firstly proposed by Houlsby et al. (2019a), is a prevalent parameter efficient tuning method which inserts trainable adapter modules between the layers of frozen PLMs to bootstrap PLMs (Mahabadi et al., 2021; Pfeiffer et al., 2021). Wang et al. (2021) adopted adapters to infuse knowledge into the large pre-trained language model. Inspired by the prompting methods, some researchers also exploit the prefix-tuning (Li and Liang, 2021) and prompt-tuning (Lester et al., 2021; Liu et al., 2022b). They preset a sequence of trainable prompt tokens to the input or intermediate layers and only update these tokens during training. Additionally, some works explore the low-rank adaptation method which injects and optimizes the low-rank matrices of attention weight in the frozen PLMs for parameter efficient tuning (Hu et al., 2022; Ma-

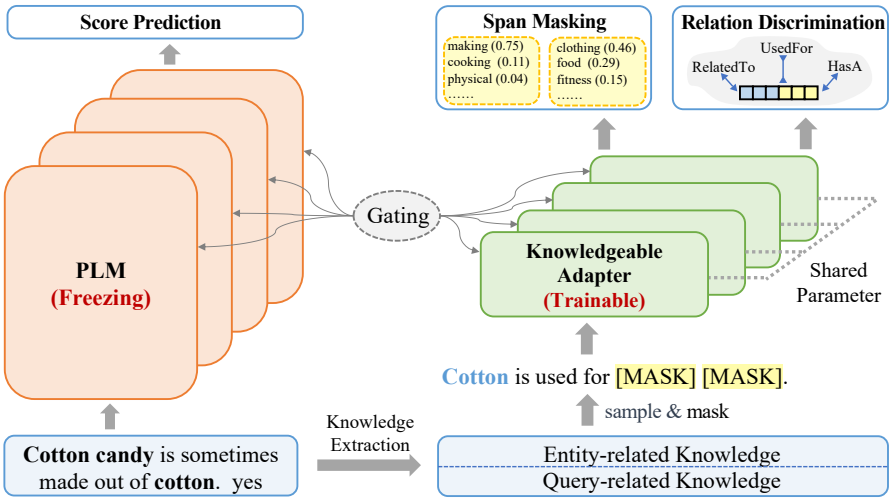

Figure 2: The architecture of our proposed KPE.

habadi et al., 2021).

In this work, we focus on the commonsense question answering task and propose a knowledgeable parameter efficient tuning network that effectively couples PLMs with external knowledge.

3 KPE Model

Following previous works (Feng et al., 2020; Yasunaga et al., 2021), we focus on the commonsense question answering task in the form of multiple-choice question answering. Formally, given a natural language query q and a set of candidate answers $\mathcal{A} = \{a\}$, we will measure the plausibility score $\boldsymbol{\rho}(q, a)$ for each answer and choose the most plausible one a^*. To promote the commonsense question answering process, we resort to external knowledge bases to extract both entity- and query-related knowledge pieces $\mathcal{K} = \{k\}$ based on q and $\mathcal{A}$.

As shown in Figure 2, our KPE couples the PLM with external knowledge via a parameter-sharing knowledgeable adapter attached to the frozen PLM. The PLM takes (q, a) as input and outputs the plausibility score $\boldsymbol{\rho}(q, a)$. The knowledgeable adapter aims to integrate the knowledge pieces k. In the following, we first introduce the *knowledge extraction* process. Then we describe the *knowledgeable adapter* that effectively integrates the extracted knowledge based on two auxiliary tasks (i.e., span masking and relation discrimination tasks), as well as gating and attention mechanisms for information interaction with the PLM.

3.1 Knowledge Extraction

A traditional source of commonsense knowledge is triple-based knowledge graphs such as ConceptNet (Speer et al., 2017). However, they encode limited types of the knowledge. Here, we use a corpus of *generic sentences* about commonsense facts, i.e., GenericsKB (Bhakthavatsalam et al., 2020) as the final knowledge source. The text can represent more complex commonsense knowledge, involving facts that relate three or more concepts. Next, we introduce how to extract entity- and query-related knowledge from GenericsKB.

Entity-related Knowledge. For entity-related knowledge, we first recognize all the entities in the query and candidate answers, and ground them to triples in ConceptNet. Then, we serialize the triples to sentences and use them as the keys to retrieve knowledge pieces from GenericsKB.

Triple Grounding. Given the query q and candidate answers $\mathcal{A} = \{a\}$, we first extract the entities e from them. Then, we ground all the triples in ConceptNet originating from e to obtain the triple set $\mathcal{T} = \{h, r, t\}$. To condense and filter the extracted triples, we follow Xu et al. (2022) to score each triple:

$$p_i = w_i * \frac{N}{N_{r_i}}, \tag{1}$$

where p_i denotes the score of the i-th triple (h_i, r_i, t_i), w_i is the triple weight provided by ConceptNet, N is the size of $\mathcal{T}$ and N_{r_i} is the number of triples with relation r_i in $\mathcal{T}$. If p_i is higher than the predefined score threshold p^*, the triple

(h_i, r_i, t_i) will be added to the selected triple set $\mathcal{T}^* \subseteq \mathcal{T}$.

Knowledge Retrieval. Now, we convert these triples in $\mathcal{T}^*$ into a series of sentences for knowledge retrieval from the unstructured commonsense knowledge base GenericsKB. Specifically, for each triple (h_i, r_i, t_i), we employ a set of pre-defined relation templates (Ma et al., 2021) to generate a sentence s_i at first. For example, the triple (sweltering, RelatedTo, hot) can be serialized to the sentence "sweltering is related to hot". Then, we take s_i as a key to retrieve the related knowledge pieces (in the form of sentences) from GenericsKB. The knowledge pieces without the entity pair (h_i, t_i) are directly disregarded. Afterwards, we select the knowledge piece which is most relevant to the query to enhance the commonsense question answering. Particularly, we use the pre-trained SimCSE (Gao et al., 2021) to obtain sentence embeddings, based on which, we compute the cosine similarity between each retrieved knowledge piece k_i and the query q as the knowledge relevance score. Finally, after processing all the triples, we choose top K ($K = 5$ in this work) retrieved knowledge pieces as entity-related knowledge $\mathcal{K}^{\mathrm{E}} = \{k_i^{\mathrm{E}}\}$ according to the computed knowledge relevance scores.

Query-related Knowledge. Considering the rich semantic information contained in the query q, we also explore the query-related knowledge for improving the commonsense question answering task. Specifically, similar to the entity-related knowledge retrieval, we retrieve query-relevant knowledge pieces from GenericsKB by concatenating the query with all the candidate answers as the key for retrieval. We also compute the knowledge relevance scores and choose top K knowledge pieces with the highest scores as the query-related knowledge $\mathcal{K}^{\mathrm{Q}} = \{k_i^{\mathrm{Q}}\}$.

3.2 Knowledgeable Adapter

In this subsection, we detail our knowledgeable adapter that effectively incorporates the above extracted knowledge for commonsense question answering.

Knowledgeable Adapter Layer. For parameter-efficiency, we connect each PLM layer with a parameter-sharing adapter layer as shown in Figure 3. For the l-th ($l \in [1, L]$) adapter layer, the input $\boldsymbol{H}_{\mathrm{A}}^{l} \in \mathbb{R}^{(m+n)\times d}$ is formed by vertically concatenating the output features $\tilde{\boldsymbol{H}}_{\mathrm{A}}^{l-1} \in \mathbb{R}^{n\times d}$ of

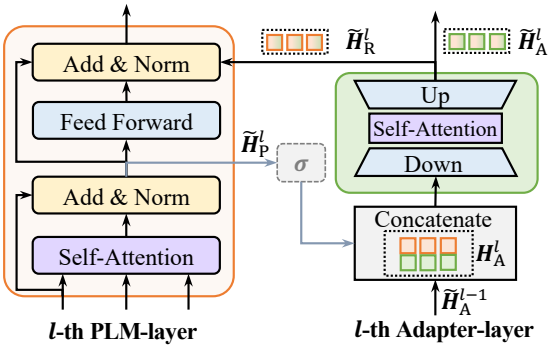

Figure 3: Illustration of our knowledgeable adapter.

the (l-1)-th adapter layer and the output features $\tilde{\boldsymbol{H}}_{\mathrm{P}}^{l} \in \mathbb{R}^{m\times d}$ of the l-th PLM layer, where m and n respectively denote the length of PLM input sequence and knowledge piece, and d is the hidden size. Note that, a *learnable gating function* is applied to filter the PLM output features $\tilde{\boldsymbol{H}}_{\mathrm{P}}^{l}$ to obtain crucial information of the query. Formally,

$$\boldsymbol{H}_{\mathrm{A}}^{l} = [\tilde{\boldsymbol{H}}_{\mathrm{P}}^{l} \odot \sigma(\boldsymbol{G}); \tilde{\boldsymbol{H}}_{\mathrm{A}}^{l-1}], \tag{2}$$

where $\boldsymbol{G} \in \mathbb{R}^{m\times d}$ is a trainable matrix and is learned in the training process, $\odot$ denotes the element-wise multiplication.

Now, given the input $\boldsymbol{H}_{\mathrm{A}}^{l}$, the adapter layer first projects it down to r dimension with a linear projection layer. Then we apply a ***self-attention layer*** to better fuse the knowledge and the query information from the PLM. After that, another linear projection layer is applied to project it up to the original dimension d. Finally, we split the output features $\boldsymbol{H'}_{\mathrm{A}}^{l} \in \mathbb{R}^{(m+n)\times d}$ of the up projection layer into two parts: $\tilde{\boldsymbol{H}}_{\mathrm{R}}^{l} \in \mathbb{R}^{m\times d}$ for the residual connection layer of the PLM and $\tilde{\boldsymbol{H}}_{\mathrm{A}}^{l} \in \mathbb{R}^{n\times d}$ for the next adapter layer. To enhance the knowledge modeling ability of the adapter, we also design the following two knowledge-related tasks, which take the final output $\tilde{\boldsymbol{H}}_{\mathrm{A}}^{L}$ of the adapter as input.

Span Masking Task. Mask prediction task can help promote the knowledge memorization of the adapter (Sun et al., 2021). For the entity-related knowledge k_i^{E} corresponding to the triple (h_i, r_i, t_i), we mask out the corresponding tokens of the tail entity mention and replace them with the same number of [MASK] to yield the corrupted sequence. Then, we fed the corrupted sequence into the adapter for forward reasoning. Based on the final adapter output $\tilde{\boldsymbol{H}}_{\mathrm{A}}^{L}$, we predict the masked tokens and calculate cross-entropy loss $\mathcal{L}_{\mathrm{MLM}}$ over them. For the query-related knowledge piece k_i^{Q},

we mask 15% tokens in total at the span level and predict them in the same way as SpanBERT (Joshi et al., 2020).

Relation Discrimination Task. Relation discrimination task can facilitate the adapter to understand the intrinsic relational facts in text and improve the robustness of the learned representations through contrastive learning (Chen et al., 2022). This task applies only to entity-related knowledge pieces that include entity pairs. Given the entity-related knowledge piece k_i^{E} and its corresponding triple (h_i, r_i, t_i), we conduct mean pooling operation over the token embeddings (from the adapter output $\tilde{\boldsymbol{H}}_{\mathrm{A}}^{L}$) of the entity mentions to obtain entity representations $\boldsymbol{v}_i^{\mathrm{H}}$ and $\boldsymbol{v}_i^{\mathrm{T}}$. Then, we follow Qin et al. (2021) to concatenate $\boldsymbol{v}_i^{\mathrm{H}}$ and $\boldsymbol{v}_i^{\mathrm{T}}$ as the relation representation $\boldsymbol{v}_i^{\mathrm{R}} = [\boldsymbol{v}_i^{\mathrm{H}}, \boldsymbol{v}_i^{\mathrm{T}}]$. For improving the understanding of relational facts, we treat the relation r_i as its positive sample and the rest relations as negative samples. Finally, we adopt the InfoNCE (van den Oord et al., 2018) loss to make the positive pair closer and push away the negative pairs:

$$\mathcal{L}_{\mathrm{RD}} = -\log \frac{\exp\left(\boldsymbol{v}_i^{\mathrm{R}} \cdot f(r_i)/\tau\right)}{\sum_{j=1}^{|\mathcal{E}|} \exp\left(\boldsymbol{v}_i^{\mathrm{R}} \cdot f(r_j)/\tau\right)}, \quad (3)$$

where τ is a temperature hyper-parameter, $|\mathcal{E}|$ is the number of relations r_i in ConceptNet, and $f(r_i)$ denotes the lookup operation for the token id of the relation r_i based on the PLM. If there are multiple tokens in r_i, we will apply mean pooling.

3.3 Model Training

Given the query context q and a candidate choice $a \in \mathcal{A}$, we leverage the output $\tilde{\boldsymbol{H}}_{\mathrm{P}}^{L}$ of the final PLM layer to compute the plausibility score $\boldsymbol{\rho}(q, a)$=MLP $(\tilde{\boldsymbol{H}}_{\mathrm{P}}^{L})$ and maximize the plausibility score of the correct answer a^* via a cross-entropy loss:

$$\mathcal{L}_{\mathrm{QA}} = \mathbb{E}_{q,a^*,\mathcal{A}}\left[-\log \frac{\exp\left(\boldsymbol{\rho}(q, a^*)\right)}{\sum_{a\in\mathcal{A}} \exp\left(\boldsymbol{\rho}(q, a)\right)}\right]. \quad (4)$$

Overall, the whole training objective of KPE is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{QA}} + \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{\mathrm{RD}}. \quad (5)$$

During training, we will randomly sample one piece of knowledge from the entity- and query-related knowledge ($\mathcal{K}^{\mathrm{E}}$ and $\mathcal{K}^{\mathrm{Q}}$) at each step. Note that for the query-related knowledge piece which is not applicable to the relation discrimination task, we will ignore the corresponding loss $\mathcal{L}_{\mathrm{RD}}$.

4 Experiments

In this section, we evaluate the effectiveness of our proposed KPE.

4.1 Datasets

We evaluate KPE on two benchmark datasets: OpenbookQA (Mihaylov et al., 2018) and CommonsenseQA 2.0 (Talmor et al., 2021).

OpenbookQA is a question answering dataset about elementary scientific knowledge and each question has four different options. This dataset contains 5,957 questions in total and we utilize the official data splits from Mihaylov et al. (2018).

CommonsenseQA 2.0 (CSQA2) is a binary classification dataset including 14,343 questions. Note that the test set of CSQA2 is not public, and we need to submit the model predictions to the official leaderboard to get the evaluation results.

4.2 Implementation Details

For knowledge retrieval, we first store GenericsKB via Elasticsearch[1] and use the retrieval function of Elasticsearch based on BM25 for retrieval. We choose the parameter values that achieve the best results on the development set. Experimentally, we set the temperature hyper-parameter τ in the relation discrimination task to 0.1 and set the score threshold p^* in triple grounding to 3.5. The down size in the adapter r is set to 256. Following previous works, the hidden dimension of the model d=1024, and the number of layers L=24. We use AdamW (Loshchilov and Hutter, 2018) optimizer in our experiments. For model training, we set the batch size to 32 and the learning rate to $2e$-5. We implement the parameter efficient tuning baselines for commonsense question answering based on AdapterHub (Pfeiffer et al., 2020).

4.3 Baselines

We compare our proposed KPE with the following parameter efficient tuning based methods and existing strong commonsense question answering methods.

Parameter Efficient Tuning based Methods. We compare KPE with the following parameter efficient tuning based methods.

[1] https://github.com/elastic/elasticsearch

Methods	OpenbookQA (RoBERTa-large)			CSQA2 (Unicorn-11B)	
	Dev Accuracy	Test Accuracy	Trainable parameters	Dev Accuracy	Trainable parameters
Bottleneck Adapter (Houlsby et al., 2019b)	62.100 (±0.7)	63.400 (±0.6)	6.345M	55.726 (±0.52)	6.345M
Prefix Tuning (Li and Liang, 2021)	63.000 (±1.2)	66.700 (±0.7)	25.772M	55.529 (±0.75)	77.313M
Compacter (Mahabadi et al., 2021)	59.800 (±1.0)	59.800 (±0.8)	**0.153M**	55.844 (±0.60)	**0.153M**
LoRA (Hu et al., 2022)	62.100 (±0.5)	64.400 (±0.8)	0.787M	55.726 (±0.76)	6.686M
MAM Adapter (He et al., 2022)	67.400 (±0.6)	70.000 (±0.2)	65.425M	55.765 (±1.31)	145.868M
KPE (ours)	**67.800 (±0.4)**	**71.300 (±0.3)**	2.369M	**68.373 (±0.58)**	2.106M

Table 1: Performance comparison with parameter efficient tuning based methods on two datasets. We use RoBERTa-large with 355.36M parameters as the PLM on OpenbookQA dataset, while taking Unicorn-11B with 11.31B parameters as the PLM on CSQA2 dataset since CSQA2 is much more difficult. The trainable parameters could differ between the two datasets due to the use of different base PLMs.

- **Bottleneck Adapter** (Houlsby et al., 2019b) is the first method to perform the adapter-based tuning in NLP.
- **Prefix Tuning** (Li and Liang, 2021) inserts a sequence of learnable prompts into the input or intermediate layers to decrease training costs.
- **LoRA** (Hu et al., 2022) presets trainable rank decomposition matrices in each layer of PLM for less trainable parameters.
- **MAM Adapter** (He et al., 2022) builds an effective adapter module that combines the advantages of adapter, prefix tuning and low-rank methods.
- **Compacter** (Mahabadi et al., 2021) is built on top of ideas from adapters, low-rank optimization, and parameterized hypercomplex multiplication layers, achieving a better trade-off between task performance and the number of trainable parameters.

For fair comparison, we improve these baseline methods with our extracted knowledge by concatenating the extracted knowledge with the original inputs of these baselines.

Existing Commonsense Question Answering Methods. We also compare KPE with the existing strong commonsense question answering methods. For OpenbookQA dataset, we compare our model with the following baselines that enhance PLMs with knowledge via GNN modules: (1) RN (Santoro et al., 2017), (2) RGCN (Schlichtkrull et al., 2018), (3) GconAttn (Wang et al., 2019), (4) MHGRN (Feng et al., 2020), (5) QA-GNN (Yasunaga et al., 2021), (6) GSC (Wang et al., 2022), (7) JointLK (Sun et al., 2022). For fair comparison, we use the same PLM (i.e., RoBERTa-large (Liu et al., 2019)) in all the above baselines and our KPE on OpenbookQA.

For CSQA2 dataset, we employ the vanilla Unicorn-11B (Lourie et al., 2021) as the PLM model for KPE, and compare KPE with the following fully fine-tuned model from the official leaderboard[2]: (1) T5-large (Raffel et al., 2020), (2) Unicorn-large (Lourie et al., 2021), (3) T5-11B (Raffel et al., 2020), (4) Unicorn-11B (Lourie et al., 2021), (5) GKP+Unicorn-11B-ft (Liu et al., 2022a). Among these baselines, GKP+Unicorn-11B-ft performs best. It handcrafts demonstration examples to guide GPT3 (Brown et al., 2020) to generate knowledge and integrates the knowledge via prompting for commonsense question answering.

4.4 Results and Analysis

Table 1 reports the results of our proposed KPE in comparison with the prevalent parameter efficient tuning based methods on both OpenbookQA and CSQA2 datasets. Note that, for a fair comparison, we concatenate our extracted commonsense knowledge with the original inputs of these baselines. Since the annotation of the CSQA2 test set is not released, we only report the comparison results on the dev set. From table 1, we can observe that: (1) KPE consistently outperforms all the baselines on both datasets. Compared to the best baseline method, KPE achieves around 12.5% and 1.3% improvements on CSQA2 dev set and OpenbookQA test set, respectively. We believe that KPE benefits from the designed knowledgeable adapter which is parameter-efficient and effectively incorporates the commonsense knowledge. Moreover, KPE achieves a much larger improvement on CSQA2 dataset (+12.5%) than OpenbookQA dataset (+1.3%). The reason could be that

[2]https://leaderboard.allenai.org/csqa2/submissions/public

Models	RoBERTa-large
Fine-tuned LMs (w/o KG)	64.80 (±2.37)†
+ RN (Santoro et al., 2017)	65.20 (±1.57)†
+ RGCN (Schlichtkrull et al., 2018)	62.45 (±1.48)†
+ GconAtten (Wang et al., 2019)	64.75 (±1.18)†
+ MHGRN (Feng et al., 2020)	66.85 (±1.19)†
+ QA-GNN (Yasunaga et al., 2021)	67.80 (±2.75)†
+ GSC (Wang et al., 2022)	70.33 (±0.81)†
+ JointLK (Sun et al., 2022)	70.34 (±0.75)
+ KPE (ours)	**71.30 (±0.30)**

Table 2: Test accuracy comparison on OpenbookQA. † denotes the reported results in GSC (Wang et al., 2022).

Models	Dev	Test	Trainable parameters
T5-large (Raffel et al., 2020)	53.8	54.6†	737.67M
Unicorn-large (Lourie et al., 2021)	56.4	54.9†	737.67M
T5-11B (Raffel et al., 2020)	68.5	67.8†	11307M
Unicorn-11B (Lourie et al., 2021)	69.9	70.2†	11307M
GKP+Unicorn-11B-ft (Liu et al., 2022a)	**72.37**	**73.03**†	11307M
KPE+Unicorn-11B	68.95	70.16‡	**2.106M**

Table 3: Performance comparison with fully fine-tuned methods on CSQA2. † denotes the reported results from papers (Talmor et al., 2021; Liu et al., 2022a) and ‡ denotes the reported result on official leaderboard.

Models	OpenbookQA		CSQA2
	Dev	Test	Dev
KPE-w/o-E	66.00 (±0.6)	69.40 (±0.4)	66.59 (±0.41)
KPE-w/o-Q	67.00 (±0.4)	68.80 (±0.4)	63.60 (±0.31)
KPE-w/o-E&Q	66.00 (±0.4)	68.50 (±0.7)	63.32 (±0.61)
KPE-w/o-S	66.40 (±0.8)	70.10 (±0.5)	64.93 (±1.04)
KPE-w/o-R	66.40 (±1.0)	70.70 (±0.3)	65.41 (±0.77)
KPE-w/o-S&R	67.60 (±0.2)	69.80 (±0.6)	63.75 (±0.56)
KPE-w/o-A	67.40 (±0.4)	70.40 (±0.2)	64.82 (±1.62)
KPE-w/o-G	66.90 (±0.5)	70.10 (±0.3)	66.57 (±0.51)
KPE	**67.80 (±0.4)**	**71.30 (±0.3)**	**68.37 (±0.58)**

Table 4: Ablation study on OpenbookQA and CSQA2.

the CSQA2 dataset is much more difficult, in which the knowledge is more needed. Thus, KPE achieves a greater improvement on CSQA2 dataset by effectively incorporating the external knowledge. (2) Compared to Bottleneck Adapter, Prefix Tuning and MAM Adapter, KPE introduces fewer parameters while achieving conspicuous improvements on both datasets. The reason is that our knowledgeable adapter employs an efficient parameter-sharing strategy and better integrates the knowledge via two auxiliary knowledge-related tasks. The gating and attention mechanisms also help the adapter to focus on useful knowledge for improving commonsense question answering. (3) The baseline methods Compacter and LoRA, although introducing fewer parameters, achieve much lower performance than KPE. Our method achieves a better trade-off between the number of trainable parameters and task performance.

Table 2 and 3 show the results of our model in comparison with the existing strong commonsense question answering methods on OpenbookQA dataset and CSQA2 dataset, respectively. As we can see from Table 2, our KPE outperforms all the GNN based methods and achieves the best performance. It demonstrates the effectiveness of our KPE with the knowledgeable adapter for incorporating knowledge to improve commonsense question answering. We believe that KPE could further benefit from the advancement of large language models and is of much value to the parameter efficient tuning research.

From Table 3, we can observe that our model KPE based on the PLM Unicorn-11B achieves comparable performance to the best fully fine-tuned models Unicorn-11B and GKP+Unicorn-11B-ft, through updating a much smaller amount of parameters (around 0.019% compared to their parameter scale).

4.5 Ablation Study

To verify the importance of each module in our KPE, we compared it with the following variants: (1) **KPE-w/o-E:** A variant of KPE that removes the entity-related knowledge. (2) **KPE-w/o-Q:** A variant of KPE that removes the query-related knowledge. (3) **KPE-w/o-E&Q:** A variant of KPE that removes both entity- and query-related knowledge. Accordingly, the span masking and relation discrimination tasks are removed. (4) **KPE-w/o-S:** A variant of KPE that removes the span masking task. (5) **KPE-w/o-R:** A variant of KPE that removes the relation discrimination task. (6) **KPE-w/o-S&R:** A variant of KPE that removes both span masking and relation discrimination tasks. (7) **KPE-w/o-A:** A variant of KPE that replaces the self-attention mechanism in the knowledgeable adapter with a conventional nonlinearity function. (8) **KPE-w/o-G:** A variant of KPE that replaces the learnable gating function in the knowledgeable adapter with direct concatenation.

Table 4 shows the results of the ablation study. We can obtain the following observations: (1) On both datasets, removing query-related knowledge results in a larger performance drop than removing entity-related knowledge. It demonstrates the importance of the query-related knowledge for com-

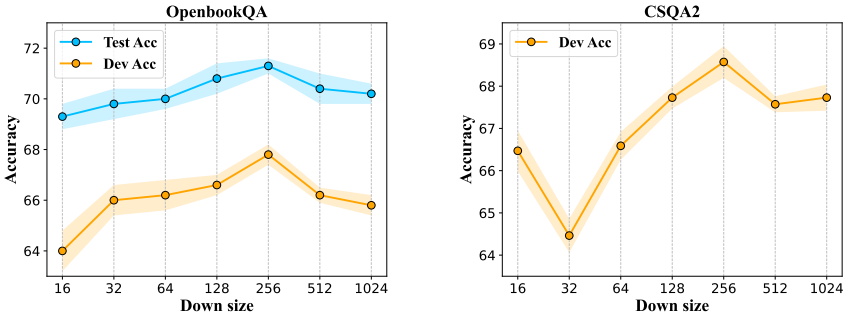

Figure 4: Impact of different down size r on two datasets.

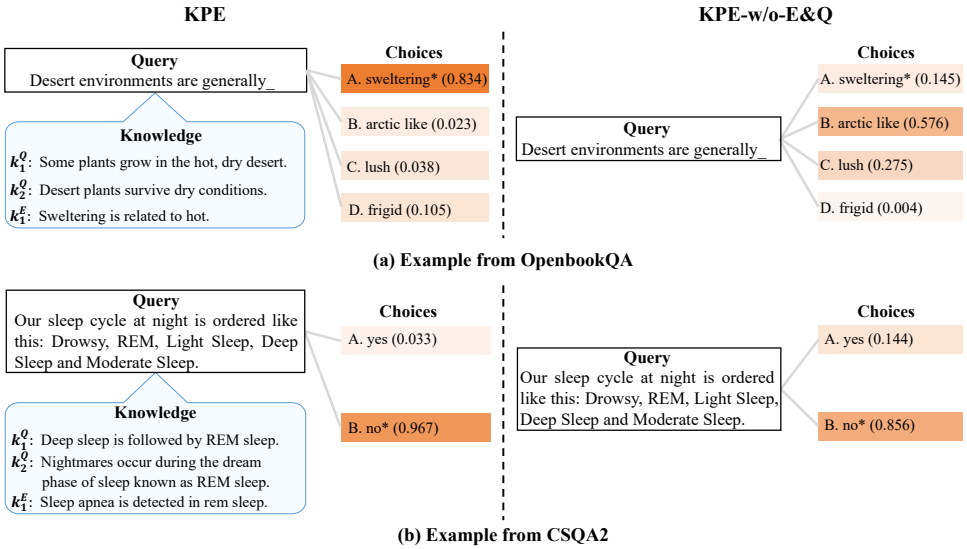

Figure 5: Illustration of KPE results.

monsense question answering. When removing both entity- and query-related knowledge, the performance largely decreases (-2.8% and -5.05% on OpenbookQA and CSQA2, respectively). (2) Disabling any auxiliary knowledge-related task will result in performance degradation, which shows that both tasks enable the adapter to better capture the knowledge, thus improving the commonsense question answering. (3) KPE consistently outperforms KPE-w/o-G and KPE-w/o-A on both datasets, which verifies that both the gating and self-attention mechanisms promote the knowledge integration for improving commonsense question answering.

4.6 Impact of Down Size r in Adapter

To explore the impact of the down size r on model performance, we vary r from 16 to 1024, and report the results on two datasets in Figure 4. We can observe that the accuracy on both datasets generally first grows and reaches the highest value from 16 to 256, while it begins to drop when r is larger than 256. Overall, KPE achieves the best performance at r=256 on both OpenbookQA and CSQA2 datasets.

4.7 Case Study

In order to intuitively understand how the external knowledge in KPE helps improve the commonsense question answering, we compare KPE with the variant KPE-w/o-E&Q. We visualize the predicted score distributions over the candidate choices using two examples from OpenbookQA and CSQA2 datasets. As can be seen from Figure 5(a), given the query "Desert environments are generally _", KPE makes the right choice "sweltering" while KPE-w/o-E&Q assigns a higher score to the incorrect choice "arctic like". We believe that the extracted knowledge (e.g., "some plants grow in the hot, dry desert", "sweltering is related to hot.") facilitates the commonsense question answering. In addition, we can observe from Figure 5(b) that although both KPE and KPE-w/o-E&Q correctly predict the answer, KPE is more confident with the prediction results by benefiting from the extracted knowledge.

5 Conclusion

In this work, we present a knowledgeable parameter efficient tuning network KPE to effectively incorporate both entity- and query-related knowledge for improving commonsense question answering. Particularly, we design a parameter-sharing knowledgeable adapter as the plugin attached to the frozen PLM to incorporate knowledge. Two auxiliary knowledge-related tasks are specifically designed for the adapter to better model and capture the knowledge. Moreover, to make the adapter integrate relevant knowledge, we introduce gating and attention mechanisms to respectively filter and fuse the query information from the PLM. Experiments on two benchmark datasets have demonstrated the effectiveness and parameter-efficiency of KPE for commonsense question answering. In future work, we will explore to integrate other parameter-efficient tuning tricks in KPE.

Limitations

The performance of KPE is also related to the used pre-trained language model (PLM), in addition to the proposed framework. KPE could suffer from unsatisfactory performance when the base PLM is not strong enough. Applying our proposed KPE to stronger PLMs, such as DeBERTa, may lead to further improvements.

Acknowledgement

This work was supported by the National Key R&D Program of China (2020AAA0105200), the National Science Foundation of China (NSFC No. U19B2020, No. 62276029, No. 62106249), Beijing Academy of Artificial Intelligence (BAAI) and CCF-Zhipu.AI Large Model Fund (No. 202217).

References

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *CoRR*.

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*, pages 1877–1901.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 4086–4092.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1295–1309.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019a. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019b. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 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*.

Philip N. Johnson-Laird. 1980. Mental models in cognitive science. *Cogn. Sci.*, pages 71–115.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, pages 64–77.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1896–1907.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4582–4597.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing,*, pages 2829–2839.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022a. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3154–3169.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 61–68.

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

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13480–13488.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13507–13515.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems*, pages 1022–1035.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 487–503.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 3350–3363.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, pages 140:1–140:67.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, pages 4967–4976.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - International Conference, ESWC*, Lecture Notes in Computer Science, pages 593–607.

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*, pages 4444–4451.

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.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5049–5060.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of AI through gamification. In *Proceedings of the Neural Information Processing Systems*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. 2022. GNN is a counter? revisiting GNN for question answering. In *International Conference on Learning Representations*.

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

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019. Improving natural language inference using external knowledge in the science questions domain. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 7208–7215.

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng, Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng, and Xuedong Huang. 2022. Human parity on commonsenseqa: Augmenting self-attention with external attention. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2762–2768.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 535–546.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☒ A2. Did you discuss any potential risks of your work?
There are no potential risks in our paper.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.