MVP-Tuning: Multi-View Knowledge Retrieval with Prompt Tuning for Commonsense Reasoning

Yongfeng Huang[1], **Yanyang Li**[1,2], **Yicong Xu**[4],
Lin Zhang[3], **Ruyi Gan**[3], **Jiaxing Zhang**[3], **Liwei Wang**[1*]
[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]SenseTime Group Inc.
[3]International Digital Economy Academy (IDEA), China
[4]Microsoft Cognitive Services Research
{yfhuang22, yyli21, lwwang}@cse.cuhk.edu.hk
yicxu@microsoft.com,{zhanglin, ganruyi, zhangjiaxing}@idea.edu.cn

Abstract

Recent advances in pre-trained language models (PLMs) have facilitated the development of commonsense reasoning tasks. However, existing methods rely on multi-hop knowledge retrieval and thus suffer low accuracy due to embedded noise in the acquired knowledge. In addition, these methods often attain high computational costs and nontrivial knowledge loss because they encode the knowledge independently of the PLM, making it less relevant to the task and resulting in a poor local optimum. In this work, we propose **M**ulti-**V**iew Knowledge Retrieval with **P**rompt Tuning (MVP-Tuning). Our MVP-Tuning leverages similar question-answer pairs in training set to improve knowledge retrieval and employs a single prompt-tuned PLM to model knowledge and input text jointly. We conduct our experiments on five commonsense reasoning QA benchmarks to show that MVP-Tuning outperforms all other baselines in 4 out of 5 datasets with only as most 2% trainable parameters. The ensemble of our MVP-Tuning models even gets a new state-of-the-art performance on OpenBookQA and is ranked first place on the leaderboard[1]. Our code and data are available[2].

1 Introduction

Endowing machines with human-like commonsense reasoning capabilities has gained increasing interest in natural language processing in recent years (Talmor et al., 2019; Rajpurkar et al., 2018). Large pre-trained language models (Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019; Brown et al., 2020a; Roberts et al., 2020; He et al., 2020) offer unprecedented potential to mine knowledge because of their unique capability in in-context learning. However, given their black-box nature, these models lack essential interpretability, resulting in the embedded knowledge that is always implicit, difficult to interpret, and fragmented. Therefore, people have developed methods to explicitly inject external knowledge, such as knowledge graphs (KG), as contextual knowledge into downstream tasks like commonsense reasoning.

The main challenge of the above solution lies in utilizing knowledge to serve individual queries while suffering the scalability issue since there can be millions of nodes in the knowledge graph. Intuitively, how to extract a partial knowledge graph, i.e., a subgraph, effectively and efficiently is crucial. Recent efforts focus on the multi-hop knowledge retrieval strategy (Feng et al., 2020a), which anchors input context entities to KG nodes and obtains relevant subgraphs from these nodes and the corresponding multi-hop neighbors. Knowledge triplets retrieved by multi-hop retrieval need to be directly connected in the knowledge graph and form a path. This process is highly sensitive to the quality of the knowledge graph, e.g., it tends to fail when necessary triplets are distant from the query and even in another subgraph. Therefore, the knowledge extracted by this strategy is often incomplete and biased as the neighbors of the input context entities bound the search span. To this, we propose *multi-view retrieval*, which expands the pool of knowledge triplet candidates with additional highly-related question-answer pairs. This method does not suffer from the limitation of multi-hop retrieval and is able to connect distant or disjoint triplets via some similarity measurements, resulting in broader and more diverse triplets from the KG. Figure 1 compares these two retrieval strategies. For example, given the question "What are candles good for eliminating?", two retrieved multi-view knowledge triplets "(candle, CapableOf, emit

*Corresponding author.
[1]The anonymous submission is in https://leaderboard.allenai.org/open_book_qa/submission/cdtvnvg4kc1nql1dnu3g
[2]https://github.com/kochsnow/MVP-Tuning/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13417–13432

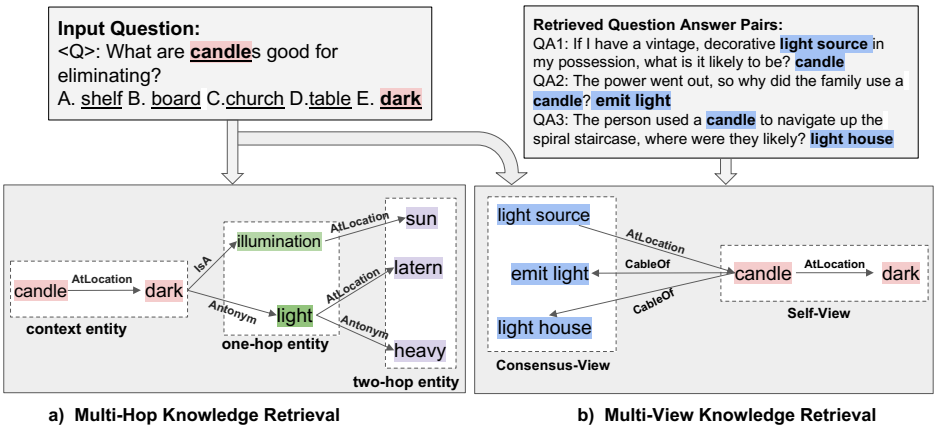

Figure 1: Multi-hop knowledge retrieval vs. Multi-view knowledge retrieval.

light)" and "(candle, AtLocation, dark)" are sufficient to guide the PLM to reason and output the answer "dark" directly. On the other hand, the conventional multi-hop strategy needs to retrieve three triplets "(dark, Antonym, light)", "(light, Antonym, heavy)", and "(dark, IsA, illumination)", which are much noisier and more challenging to reason.

Having extracted the target knowledge, how can we harness this knowledge to serve the ultimate purpose of reasoning? An intuitive way is to employ Graph Neural Networks (GNNs) to output node embeddings for KGs and then fuse them with embeddings of other input texts from PLMs (Schlichtkrull et al., 2018; Lin et al., 2019; Feng et al., 2020a; Yasunaga et al., 2021; Wang et al., 2021a; Jinhao Jiang and Wen, 2022). Despite being straightforward, this solution inherits critical issues from GNNs, such as over-smoothness. Instead, we explore a new way of encoding knowledge from KGs, simple yet effective. For encoding, we directly combine the retrieved knowledge triplets as texts and concatenate them with other textual information as the input of the PLM. Our approach can alleviate the computational cost and reduce the information loss compared to previous GNNs based approaches.

In this paper, our proposed multi-view knowledge retrieval scheme can outperform existing work regarding efficiency and accuracy by a large margin when built with recent successful parameter-efficient learning techniques, such as prompt-tuning (Li and Liang, 2021; Liu et al., 2021c; Lester et al., 2021) with a PLM. Therefore, we name our framework as **M**ulti-**V**iew Knowledge Retrieval with **P**rompt Tuning (MVP-Tuning). The multi-view knowledge retrieval strategy brings more accurate knowledge localization with less computational cost. The obtained knowledge is fed into a PLM model to augment the information in text. To further improve the capability of our model, we integrate parameter-efficient learning in this context.

In summary, our primary contributions are:

- We proposed a multi-view knowledge graph retrieval algorithm that acquires knowledge using similar question-answer pairs as complementary queries.
- We point out that the KG encoder is non-essential and propose a simple yet effective unified paradigm, that is, a single PLM jointly models the text and knowledge without any KG encoder, for commonsense reasoning tasks.
- We present a systematic study on the effectiveness of prompt learning in commonsense QA, including the impact of prompt length and initialization strategy.
- We conduct experiments on five popular commonsense QA benchmarks, including CommonsenseQA, OpenBookQA, SoicalIQA, PIQA, and Riddle-Sense, and compare with extensive baselines. Our MVP-Tuning outperforms other approaches in 4 out of 5 datasets with at most 2% of the trainable parameters of the same-scale PLM. MVP-Tuning also improves previous state-of-the-art results on CommonsenseQA and OpenBookQA under the low-resource setting. We submitted the predictions of our MVP-Tuning model to the leaderboards of CommonsenseQA and OpenBookQA, where it achieves state-of-the-art results in comparison to other models with a similar scale PLM. Our MVP-Tuning ensembled predictions even obtain the best result, to date, on OpenBookQA's leaderboard.

2 Related Work

GNN-Powered Structured Knowledge Utilization Existing techniques often combine PLMs with a variety of KG encoders to leverage knowledge and context information. There are a number of developed knowledge encoders. Some encode retrieved knowledge using complex neural network designs, like RGCN (Schlichtkrull et al., 2018), GconAttn (Lin et al., 2019), MHGRN (Feng et al., 2020a), and QAGNN (Yasunaga et al., 2021). Others attempt to build knowledge encoders with simpler designs that exhibit superior performance. GSC (Wang et al., 2021b) creates a basic graph neural counter that beats more complicated approaches, indicating that GNN-based encoders are merely doing simple counting. SAFE (Jinhao Jiang and Wen, 2022) encodes relation routes using a simple two-layer MLP. However, these approaches encode text and knowledge independently. GreaseLM (Zhang et al., 2022), on the other hand, integrates the representations of both KG and PLM encoders over multiple layers of modal interaction processes.

Prompt-Based Unstructured Knowledge Utilization A line of research has investigated the use of unstructured knowledge sources, such as Wikipedia and dictionaries, for commonsense reasoning (Xu et al., 2021b; Lv et al., 2020a). These methods append related knowledge to the input context as a prompt to improve commonsense reasoning. For example, Bhakthavatsalam et al. (2020) combined knowledge from ConceptNet, WordNet, and other corpora to create 3.5 million generic statements and show that this knowledge can enhance both accuracy and explanation quality. Other studies have focused on comparing different methods for incorporating external knowledge from relevant corpora (Mitra et al., 2020). Additionally, there have been efforts to generate missing facts from PLMs to supplement external knowledge sources. For instance, Bosselut et al. (2019) fine-tuned a PLM on ATOMIC for commonsense knowledge graph completion, and Liu et al. (2021a) prompted GPT-3 (Brown et al., 2020b) directly to obtain knowledge for reasoning.

Prompt Learning Prompt learning is a simple yet effective approach to adapt a PLM for specific downstream tasks by adding prompt tokens in the input. A line of prompt learning works utilizes automated search techniques to identify appropriate discrete prompting words (Shin et al., 2020; Deng et al., 2022). In contrast to these discrete prompt learning methods, there are also a number of works known as soft prompt learning that has been developed. These include Prompt Tuning (Lester et al., 2021), P-tuning (Liu et al., 2021c), Prefix-Tuning (Li and Liang, 2021), and P-Tuning v2 (Liu et al., 2021b). These approaches use trainable soft prompt tokens to steer PLMs' generation.

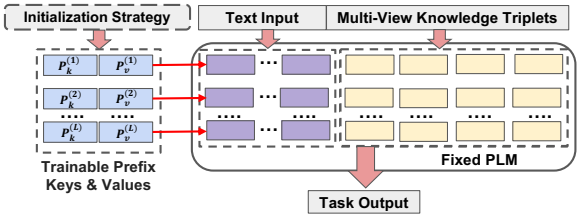

Figure 2: MVP-Tuning framework.

3 Problem Statement

In this work, we study the multiple-choice commonsense question answering (Talmor et al., 2019; Mihaylov et al., 2018). Given a natural language question q and a collection of n response candidates $C = \{c_1, \cdots, c_n\}$, the purpose is to pick the most appropriate candidate $c^\star \in C$ to answer the question q based on the requisite commonsense knowledge. In accordance with previous work (Lin et al., 2019), we address this commonsense reasoning task in a *knowledge-aware* setting that embraces a commonsense knowledge graph (CSKG) as the commonsense knowledge source.

An external CSKG can be described formally as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is the set of all concept nodes (e.g., *leg* and *fire*), $\mathcal{R}$ is the set of relation types (e.g., *CapableOf* and *IsA*), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of relational edges that connects two concept nodes in the $\mathcal{V}$. Specifically, We employ ConceptNet (Speer et al., 2017), which consists of 799,273 nodes and 2,487,003 edges.

4 Approach: MVP-Tuning

As shown in Figure 2, MVP-Tuning is based on the PLM and includes the multi-view knowledge retrieval module and the prompt tuning module. We augment the input context with multi-view retrieved knowledge, and the prompt tuning module optimizes the PLM in a parameter-efficiency way.

4.1 Multi-View Knowledge Retrieval Module

We retrieve knowledge in CSKG from two views: 1) self-view that selects triplets related to the question-choice pair (q, c), and 2) consensus-view

that retrieves triplets of other question-answer pairs that are semantically similar to (q, c).

Self-View Knowledge Retrieval Following KEAR (Xu et al., 2021a), we implement self-view knowledge by retrieving the most relevant relation triplet in the CSKG. We denote the self-view knowledge retrieval process as $\mathbf{K}_{\mathrm{SV}}$. Given a question-choice pair (q, c), $\mathbf{K}_{\mathrm{SV}}(q, c)$ returns the most relevant triplet (e_1, r, e_2) in ConceptNet as self-view knowledge. The self-view knowledge retrieval process $\mathbf{K}_{\mathrm{SV}}$ is performed as the following: Firstly, we use the entity linking tool (Loper and Bird, 2002) to find all entities $E_q = \{e_q^{(1)}, ..., e_q^{(n_q)}\}, E_c = \{e_c^{(1)}, ..., e_c^{(n_c)}\}$ appearing in the question q and choice c respectively, where n_q and n_c are the number of entities in q and c. We filter out entities in E_q and E_c whose lengths do not match the Wiktionary definition. After that, we select the entity with the maximum length in E_q and E_c as the question and choice entity e_q and e_c[3]. Then we find all triplets in ConceptNet containing both e_q and e_c and choose the one with the maximum total length as retrieved self-view knowledge (e_1, r, e_2). If there is no such triplet, we retrieve all triplets sourcing from the choice entity e_c in ConceptNet. Each triplet j's score s_j is the product of its confidence w_j (given by ConceptNet) and the relation type weight t_{r_j}: $s_j = w_j \cdot t_{r_j} = w_j \cdot \frac{N}{N_{r_j}}$, where r_j is the relation type of j, N is the total number of triplets originating from the choice entity e_c, and N_{r_j} is the number of triplets having relation r_j among these triplets. We select the triplet with the largest score as self-view knowledge (e_1, r, e_2).

Consensus-View Knowledge Retrieval Self-view knowledge is obtained by querying KG with the question-choice pair, which is limited in scope. Meanwhile, the knowledge retrieved by conventional multi-hop knowledge retrieval is still restricted or even noisy, as depicted in Section 1. To address this limitation, we propose consensus-view knowledge retrieval with query expansion to improve the retrieval performance. In query expansion, a given query is reformulated by first discovering semantically related queries, and then re-weighting the terms in the original query (Vechtomova and Wang, 2006; Azad and Deepak, 2019; Carpineto and Romano, 2012). In our consensus-view knowledge retrieval, similar question-answer pairs in the training set collectively retrieve more relevant knowledge from KG. We define the consensus-view knowledge retrieval process as $\mathbf{K}_{\mathrm{CV}}$. Given a question-choice pair (q, c) and the number of retrieved items m, the consensus-view knowledge retrieval process is as follow: We employ BM25 (Robertson et al., 2009) to choose the m most pertinent question-answer pairs $\{(q_1, a_1), (q_2, a_2), \cdots, (q_m, a_m)\}$ from the training data for the given the question-choice pair (q, c). Then we use the self-view knowledge of these selected question-answer pairs to construct the consensus-view knowledge of (q, c), denoted as $\mathbf{K}_{\mathrm{CV}}(q, c) = \{\mathbf{K}_{\mathrm{SV}}(q_1, a_1), \mathbf{K}_{\mathrm{SV}}(q_2, a_2), \cdots, \mathbf{K}_{\mathrm{SV}}(q_m, a_m)\}$.

Constructing Multi-View Knowledge Augmented Input Given the question q and its related choices $(c_1, \cdots, c_i, \cdots, c_n)$, we first obtain the self-view knowledge $\mathbf{K}_{\mathrm{SV}}(q, c_i)$ and the consensus-view knowledge $\mathbf{K}_{\mathrm{CV}}(q, c_i)$ for each possible question-choice pair (q, c_i). We then append the corresponding multi-view knowledge $\mathbf{K}_{\mathrm{SV}}(q, c_i)$ and $\mathbf{K}_{\mathrm{CV}}(q, c_i)$ to each (q, c_i) to construct its augmented text representation $\mathrm{text}_i = q \oplus c_i \oplus \mathbf{K}_{\mathrm{SV}}(q, c_i) \oplus \mathbf{K}_{\mathrm{CV}}(q, c_i)$, where $\oplus$ denotes the string concatenation. Finally, we merge the augmented text representations of all question-choice pairs as the multi-view knowledge augmented input $\mathrm{text} = \mathrm{text}_1 \oplus \mathrm{text}_2 \oplus \cdots \oplus \mathrm{text}_n$ for predicting the answer of the question q.

4.2 Prompt Tuning Module

To perform parameter-efficient learning, our MVP-Tuning framework employs prompt tuning (Li and Liang, 2021; Liu et al., 2021b; Lester et al., 2021) of the pre-trained Transformer encoder. The core mechanism of prompt tuning is to learn soft prompts, which steers a frozen pretrained language model to perform specific downstream tasks.

Transformers The Transformer encoder consists of a list of layers, each of which contains a multi-head self-attention module and a feed-forward network (FFN). In the multi-head self-attention, each attention head is defined as:

$$\mathrm{Attention}(x) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where d_k is the hidden size, $Q = xW_q, K = xW_k, V = xW_v$ and $W_q \in \mathbb{R}^{d_k \times d_k}, W_k \in \mathbb{R}^{d_k \times d_k}, W_v \in \mathbb{R}^{d_k \times d_k}$ are three learnable weight matrices. The multi-head self-attention performs N heads in parallel and concatenates their outputs

[3]For universality, we do not assume the annotations of question and choice entities e_q and e_c are always available.

to form the input to FFN. FFN is defined as:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

where $W_1 \in \mathbb{R}^{d_k \times d_h}, W_2 \in \mathbb{R}^{d_h \times d_k}$ are weights, $b_1 \in \mathbb{R}^{d_h}, b_2 \in \mathbb{R}^{d_k}$ are biases and d_h is the FFN hidden size.

P-Tuning v2 In our MVP-Tuning framework, we choose P-Tuning v2 (Liu et al., 2021b) as the prompt tuning technique because it shows decent performance on NLU tasks. P-Tuning v2 prepends a trainable prefix key and prefix value to the key K and value V in Eq. 1 at each layer, respectively.

Concretely, we denote the original key and value at the l-th Transformer encoder layer as K_l and V_l. We then define a learnable prefix key $P_k \in \mathbb{R}^{L \times n_p \times d_k}$ and prefix value $P_v \in \mathbb{R}^{L \times n_p \times d_k}$, where L is the number of layers and n_p is the length. These prefix key and value will be added to K_l and V_l via $K_l' = [P_k^{(l)}; K_l], V_l' = [P_v^{(l)}; V_l]$, where $[;]$ is the concatenation along the first dimension, $P_k^{(l)} \in \mathbb{R}^{n_p \times d_k}$ and $P_v^{(l)} \in \mathbb{R}^{n_p \times d_k}$ is the corresponding prefix key and value for l-th layer in P_k and P_v. K_l' and V_l' will replace the original K_l and V_l when performing the multi-head self-attention. During training, we only optimize P_k and P_v and freeze the pretrained model.

Previous work (Lester et al., 2021) suggests that the initialization of prefix key and value is crucial for the downstream task performance. We thus explore the following two initialization strategies:

Random Initialization P_k and P_v are randomly initialized by a Gaussian distribution.

Relation Augmentation Initialization To introduce additional relation information, we initialize P_k and P_v by the relation embeddings. We list all CSKG relations and encode them using the word embeddings from the pretrained model. Since a relation could contain multiple words, we average all word embeddings of one relation to build a fixed-length relation embedding. The concatenation of all relation embeddings $P_r \in \mathbb{R}^{n_p \times d_k}$ will pass through a MLP to obtain P_k and P_v (Liu et al., 2021b), where the prefix length n_p now equals the number of relations in CSKG.

5 Experiments

As shown in Table 1, we experiment on five commonsense reasoning multiple-choice QA datasets.

Task	Train	Dev	Test
CommonsenQA official split	9,741	1,221	-
CommonsenQA in-house split	8,500	1,221	1,241
OpenBookQA	4,957	500	500
SocialIQA	33,410	1,954	-
PIQA	16,113	1,838	-
RiddleSenseQA	3,510	1,021	-

Table 1: Statistics of the datasets. "-" denotes the unused or unavailable dataset split in our experiments.

5.1 Datasets

OpenBookQA (Mihaylov et al., 2018) is a 4-way multiple-choice QA dataset consisting of elementary scientific questions intended to evaluate science commonsense knowledge. We report the accuracy of our final system on the official test set (Mihaylov and Frank, 2018) and submit the test results to the leaderboard.

CommonsenseQA (Talmor et al., 2019) is a 5-way multiple-choice QA dataset. It is constructed using ConceptNet (Speer et al., 2017). CommonsenseQA has two split schemes, including in-house split (Lin et al., 2019) and official split (Talmor et al., 2019). We therefore report results in both the in-house split[4] and the official split. The test set of CommonsenseQA is not publicly available, so we submit our model's predictions to the official leaderboard to evaluate the test accuracy.

SocialIQA (Sap et al., 2019) is a 3-way multiple-choice QA dataset used to assess social commonsense knowledge comprehension. The test set is unavailable. For a fair comparison, we report results on the development set (Shwartz et al., 2020).

PIQA (Bisk et al., 2020) is a set of binary-choice questions about physical common sense. Because PIQA does not release the test set, all assessments are based on the development set.

Riddle-Sense (Lin et al., 2021) is a five-choice QA dataset regarding commonsense riddles. Since the Riddle-Sense test is hidden, evaluations are carried out on its development set.

5.2 Implementation and Training Details

For fair comparison, our MVP-Tuning method utilizes the same pretrained models as the above benchmark. We primarily seed our MVP-Tuning method with the RoBERTa-large (Liu et al., 2019b) model for all datasets. We additionally test AristoRoBERTa (Clark et al., 2020)[5] for OpenBookQA.

[4] In the in-house split, the results of test set are reported based on the model with the best performance in the dev set.

[5] OpenBookQA provides an extra corpus of scientific facts in a textual form. AristoRoBERTa uses the facts correspond-

Methods	RoBERTa-large	AristoRoBERTa	Unfreezed Param. M_{plm} + M_{kg}	Model Param. N_{plm} + N_{kg}
Fine-Tuned PLM (w/o KG)	64.80 (±2.37)	78.40 (±1.64)	355M+0	355M+0
+ RN (Santoro et al., 2017)	65.20 (±1.18)	75.35 (±1.39)	355M+210k	355M+819M
+ RGCN (Schlichtkrull et al., 2018)	62.45 (±1.57)	74.60 (±2.53)	355M+365k	355M+819M
+ GconAttn (Lin et al., 2019)	64.75 (±1.48)	71.80 (±1.21)	355M+700k	355M+819M
+ MHGRN (Feng et al., 2020b)	66.85 (±1.19)	80.6	355M+547k	355M+819M
+ QA-GNN (Yasunaga et al., 2021)	67.80 (±2.75)	82.77 (±1.56)	355M+2.85M	355M+821M
+ GreaseLM (Zhang et al., 2022)	68.80 (±1.75)*	84.80	355M+3.6M	355M+822M
+ GSC (Wang et al., 2021b)	70.33 (±0.81)	86.67 (±0.46)	355M+3k	355M+3k
+ SAFE (Jinhao Jiang and Wen, 2022)	69.2	87.13	355M+4.7k	355M+4.7k
MVP-Tuning (prefix length=120)	**71.00 (±0.21)**	**87.50 (±0.10)**	6.07M+0	355M+0

Table 2: Test accuracy comparison on OpenBookQA. Our reproduced results are denoted with *. M_{plm} and M_{kg} represent trainable parameters of PLM encoder and KG encoder respectively. N_{plm} and M_{kg} represent model size of PLM encoder and KG encoder respectively.

Methods	Test Acc	Unfreezed Param. M_{plm} + M_{kg}	Model Param. N_{plm} + N_{kg}
AristoRoBERTa + GreaseLM	84.80	355M+3.6M	355M+822M
AristoRoBERTa + GSC	86.67 (±0.46)	355M+3k	355M+3k
AristoRoBERTa + SAFE	87.13	355M+4.7k	355M+4.7k
AristoRoBERTa + MVP-Tuning (prefix length=120)	87.50 (±0.10)	6.07M+0	355M+0
DeBERTa-xlarge + MVP-Tuning (prefix length=120)	87.63 (±0.17)	11.8M+0	900M+0
DeBERTa-xxlarge + MVP-Tuning (prefix length=120)	**91.3 (±0.10)**	17.7M+0	1.5B+0

Table 3: Test accuracy on OpenBookQA test set with different PLMs. M_{plm} and M_{kg} represent trainable parameters of PLM encoder and KG encoder respectively. N_{plm} and M_{kg} represent model size of PLM encoder and KG encoder respectively.

Methods	Test
AristoRoBERTa	77.8
KF + SIR (Banerjee and Baral, 2020)	80.0
AristoRoBERTa + PG (Wang et al., 2020)	80.2
AristoRoBERTa + MHGRN (Feng et al., 2020b)	80.6
Albert + KB	81.0
T5* (Raffel et al., 2020)	83.2
AristoRoBERTa + QAGNN	82.8
AristoRoBERTa + GreaseLM	84.8
AristoRoBERTa + GSC	87.4
UnifiedQA (Khashabi et al., 2020)	87.2
GenMC (Huang et al., 2022)	89.8
GenMC (ensemble) (Huang et al., 2022)	92.0
X-reasoner	94.2
AristoRoBERTa + MVP-Tuning (prefix length=120)	87.6
DeBERTa-xxlarge + MVP-Tuning (prefix length=120)	91.2
MVP-Tuning (ensemble)	95.2

Table 4: Test accuracy on OpenBookQA leaderboard.

To evaluate the effectiveness of our method, we test MVP-Tuning with larger PLMs, such as DeBERTa-xlarge and DeBERTa-xxlarge (He et al., 2020). Detailed hyperparameter setting can be found in Appendix A.1.

5.3 Baselines

Fine-tuned PLMs We fine-tune RoBERTa-large to study the impact of vanilla fine-tuned PLM, which does not use any KG and is only fed with the question and choices. For the OpenBookQA, we also fine-tune AristoRoBERTa.

PLM+KG Models combine PLMs with extra GNN-based KG encoders. With the same fine-tuned PLM, we evaluate eight KG encoder variants, including RN (Santoro et al., 2017), RGCN (Schlichtkrull et al., 2018), GconAttn (Lin et al., 2019), MHGRN (Feng et al., 2020a), QAGNN (Yasunaga et al., 2021), GSC (Wang et al., 2021b), GreaseLM (Zhang et al., 2022) and SAFE (Jinhao Jiang and Wen, 2022). Details can be seen in Appendix A.2.

5.4 Main Results

Results on OpenBookQA According to Table 2, MVP-Tuning outperforms the current PLM+KG methods in either RoBERTa-large or AristoRoBERTa setting. Although this improvement seems to be minor, it is achieved with no more than 2% trainable parameters (6.02M for MVP-Tuning vs. 355M for Fine-tuned PLM). MVP-Tuning allows us to use a larger PLM with a low training cost. Table 3 shows that the test performance of MVP-Tuning with DeBERTa-xxlarge is 4% better than the best PLM+KG model, while having 20× fewer trainable parameters (17.7M vs. 355M). Compared to other systems on the leaderboard of OpenbookQA (Table 4), our MVP-Tuning with DeBERTa-xxlarge ranks 3rd with only

ing to each question, prepared by Clark et al. (2020), as an additional input to the QA context.

Methods	Official Dev Acc.	In-house Dev Acc.	In-house Test Acc.	Unfreezed Param. M_{plm} + M_{kg}	Model Param. N_{plm} + N_{kg}
Fine-Tuned PLM (w/o KG)	77.15 (±0.35)*	73.07 (±0.45)	68.69 (±0.56)	355M+0	355M+0
+ RN (Santoro et al., 2017)	76.00 (±0.65)*	74.57 (±0.91)	69.08 (±0.21)	355M+210k	355M+819M
+ RGCN (Schlichtkrull et al., 2018)	77.07 (±0.14)*	72.69 (±0.19)	68.41 (±0.66)	355M+365k	355M+819M
+ GconAttn (Lin et al., 2019)	77.56 (±0.27)*	72.61 (±0.39)	68.59 (±0.96)	355M+700k	355M+819M
+ MHGRN (Feng et al., 2020b)	79.52 (±0.15)*	74.45 (±0.10)	71.11 (±0.81)	355M+547k	355M+819M
+ QA-GNN (Yasunaga et al., 2021)	78.77 (±0.16)*	76.54 (±0.21)	73.41 (±0.92)	355M+2.85M	355M+821M
+ SAFE (Jinhao Jiang and Wen, 2022)	78.97 (±0.29)*	76.93 (±0.37)*	74.03 / 73.68 (±0.43)*	355M+4.7k	355M+4.7k
+ GreaseLM (Zhang et al., 2022)	79.44 (±0.43)*	78.5 (±0.5)	74.2 (±0.4)	355M+3.6M	355M+822M
+ GSC (Wang et al., 2021b)	80.43 (±0.21)*	79.11 (±0.22)	74.48 (±0.41)	355M+3k	355M+3k
MVP-Tuning (prefix length=100)	**83.29 (±0.13)**	**81.13 (±0.11)**	**75.89 (±0.19)**	4.92M+0	355M+0

Table 5: Performance comparison on CommonsenseQA in both official split setting and in-house split setting. Our reproduced results are denoted with *. M_{plm} and M_{kg} represent trainable parameters of PLM encoder and KG encoder respectively. N_{plm} and M_{kg} represent the size of PLM encoder and KG encoder respectively.

Methods	Single	Ensemble
RoBERTa (Liu et al., 2019a)	72.1	72.5
RoBERTa+FreeLB (Zhu et al., 2019) (ensemble)	72.2	73.1
RoBERTa+HyKAS (Ma et al., 2019)	73.2	-
RoBERTa+KE (ensemble)	-	73.3
RoBERTa+KEDGN (ensemble)	72.5	74.4
RoBERTa+MHGRN (Feng et al., 2020b)	75.4	-
RoBERTa + QA-GNN(Yasunaga et al., 2021)	76.1	-
RoBERTa + GSC(Wang et al., 2021b)	76.2	-
Albert (Lan et al., 2019)	-	76.5
Albert+PG (Wang et al., 2020)	75.6	78.2
ALBERT+HGN(Yan et al., 2020)	77.3	80.0
XLNet+GraphReason (Lv et al., 2020b)	75.3	-
UnifiedQA (11B) (Khashabi et al., 2020)	**79.1**	-
RoBERTa-large + MVP-Tuning	78.4	-

Table 6: CommonsenseQA leaderboard result.

Methods	SocialIQA	PhysQA	RiddleSense
Fine-Tuned PLM	78.25	77.53	60.72
+ GcoAttn	78.86	78.24	61.77
+ MHGRN	78.11	77.15	63.27
+ QAGNN	78.10	78.24	63.39*
+ GreaseLM	77.89*	78.02*	63.88*
+ GSC	78.61*	78.40*	64.07*
+ SAFE	78.86	**79.43**	63.78*
MVP-Tuning	**79.12**	78.94	**64.54**

Table 7: Performance comparison on SocialIQA, PhysQA, and RiddelSense (Dev accuracy). Our reproduced results are denoted with *.

17.7M trainable parameters, while most of the other QA systems are built on the T5 model with 11B trainable parameters. Moreover, our ensembled MVP-Tuning[6] rank top-1 to date. We note that the runner-up with a public technical report, GENMC-ensemble (Huang et al., 2022), combines 7 fine-tuned T5-11B models and has 4000 times more trainable parameters than ours. Table 4 also indicates that our MVP-Tuning with AristoRoBERTa performs better than the current GNN-based QA methods with the same scale PLMs.

Results on CommonsenseQA We compared our MVP-Tuning with existing PLM+KG models and fine-tuned PLMs. All of them are based on the RoBERTa-large model. As we can see in Table 5, MVP-Tuning shows a constant improvement under three evaluation settings, with 2.04% higher mean accuracy on the official dev split, 2.02% higher mean accuracy on in-house dev split, and 1.41% higher mean accuracy on the in-house test split, all without a KG encoder and with no more than 2% (4.92M vs. 355M) trainable parameters of PLM. Moreover, the variance of MVP-Tuning is smaller than the baselines, which implies the robustness of our method. We also submit our MVP-Tuning model based on RoBERTa-large to CommonsenseQA's official leaderboard. As can be seen from Table 6, MVP-Tuning offers a non-trivial advantage over every other GNN-based QA system with a comparable scale PLM.

Results on Other QA Datasets To further assess the effectiveness of the proposed MVP-Tuning, we also compare our method to the aforementioned baselines on other commonsense reasoning datasets from different domains or tasks. As shown in Table 7, our MVP-Tuning obtains the best performance in most cases, which indicates that our approach is generally effective for various commonsense reasoning datasets or tasks in a unified and parameter-efficient way.

6 Analysis

6.1 Low-Resource Setting

To test the robustness of MVP-Tuning, we examine its performance in low-resource settings, with three different proportions of training data, i.e., 5%, 10% and 20%, in CommonsenseQA and OpenBookQA. For the CommonsenseQA, we still use the in-house split setting. We follow SAFE (Jinhao Jiang and Wen, 2022) setting to report the average test per-

[6]We apply MVP-Tuning to DeBERTaV3-large, AristoRoBERTa, DeBERTa-xxlarge, and UniMC-DeBERTa-xxlarge (Yang et al., 2022) and ensemble their predictions.

Methods (%, shots)	CommonsenseQA (5%, 425)	CommonsenseQA (10%, 850)	CommonsenseQA (20%, 1700)	OpenBookQA (5%, 298)	OpenBookQA (10%, 498)	OpenBookQA (20%, 991)
RoBERTa-large	29.66	42.84	58.47	37.00	39.4	41.47
+ RGCN	24.41	43.75	59.44	38.67	37.53	43.67
+ GconAttn	21.92	49.83	60.09	38.60	36.13	43.93
+ RN	23.77	34.09	59.90	33.73	35.93	41.40
+ MHGRN	29.01	32.02	50.23	38.00	36.47	39.73
+ QA-GNN	32.95	37.77	50.15	33.53	35.07	42.40
+ GreaseLM	22.80*	56.16*	63.09*	39.00*	39.60*	42.20*
+ GSC	31.02*	35.07*	65.83*	29.60*	41.80*	42.40*
+ SAFE	36.45	56.51	65.16	38.80	41.20	44.93
+ MVP-Tuning	**48.99**	**61.16**	**67.12**	**39.60**	**49.00**	**56.00**

Table 8: Performance comparison on CommonSenseQA and OpenBookQA with different proportions of training data. * indicates the results reproduced by us.

Methods	CommonSenseQA	OpenbookQA
Input Text	76.82	80.04
+ Self-View Know.	80.42	86.8
+ Consensus-View Know.	78.46	85.4
+ Multi-Hop Know	79.11	84.0
+ Multi-View Know.	**83.29**	**87.6**

Table 9: Performance of different knowledge in RoBERTa-large on the CommonSenseQA official dev set and the OpenBookQA test set.

Initialization Strategy	CommonSenseQA	OpenbookQA
Random Init.	82.47	87.0
Relations Augmentation Init.	82.39	86.2

Table 10: Performance of two prefix initialization strategies with RoBERTa-large on the CommonSenseQA official dev set and the OpenBookQA test set. The prefix length here is 34, as there are 34 relation types in CSKG.

formance of three runs, and the best results are highlighted in bold. According to Table 8, our MVP-Tuning consistently outperforms other approaches on different training data sizes, which shows the remarkable low-resource capability of our method. And we observe that our MVP-Tuning performs the best when the number of shots is approximately between 500 and 1000, which obtains an improvement of over 5% accuracy.

6.2 Ablation Study

We conduct the ablation study on the proposed MVP-Tuning. For the multi-view knowledge retrieval, we augment the input text with self-view knowledge, consensus-view knowledge, and multi-view knowledge separately, then evaluate their performance on various datasets. In addition, we also examine the influence of the number of retrieved consensus-view knowledge. For the prompt-tuning module, we explore the influence of the prefix initialization strategy and prefix length.

Effect of Different Types of Knowledge According to Table 9, multi-view knowledge can provide the most comprehensive and diverse information for commonsense reasoning QA tasks, and thus achieve the best result. Consensus-view knowledge performs worse than self-view knowledge, suggesting that although consensus-view knowledge is complementary to self-view knowledge, it still misses some important knowledge. We further evaluate the performance of multi-hop knowledge. Our findings reveal that multi-hop knowledge exhibits inferior performance not only in comparison to multi-hop knowledge but also when compared to self-view knowledge. These comparative results demonstrate the efficacy of multi-view retrieval as a retrieval technique.

Effect of the Quantity of Retrieved Consensus-View Knowledge Figure 3 shows the impact of the quantity of consensus-view knowledge retrieved in MVP-Tuning. The performance generally improves with more consensus-view knowledge, but too much consensus-view information introduces noises that ultimately hurt performance.

Effect of Prefix Initialization Strategies We compare two prompt tuning module initialization strategies in Table 10. Random initialization slightly outperforms relation augmentation initialization, indicating that the basic prompt tuning is already a good baseline for MVP-Tuning.

Effect of the Number of Soft Prefix Tokens We studied the effect of the number of soft prefix tokens. Figure 4 indicates that our system is not sensitive to the length of soft prefix.

Case Study We also provide some examples in Appendix A.4 to illustrate the effectiveness of our multi-view knowledge retrieval.

7 Conclusion

In this work, we propose MVP-Tuning, a simple and effective approach to building a strong com-

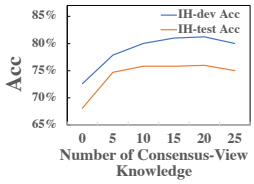

Figure 3: Effect of the number of consensus-view knowledge.

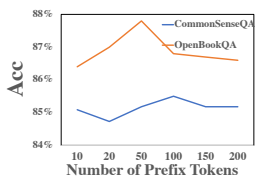

Figure 4: Effect of the number of soft prefix tokens.

monsense reasoning QA system. It strengthens the conventional knowledge retrieval results via multi-view knowledge and unifies the modeling of input text and retrieved knowledge in a single prompt-tuned PLM. Extensive experiments show the superiority of MVP-Tuning, as it beats other sophisticated approaches in 4 out of 5 popular commonsense QA benchmarks while having less than 2% trainable parameters. MVP-Tuning achieves a new state-of-the-art performance in OpenBookQA and wins first place in the leaderboard.

Limitation

This paper presents the MVP-Tuning framework, which combines multi-view knowledge retrieval with prompt tuning and incorporates retrieved knowledge in a simple KG-encoder-free paradigm. However, there are limitations to our approach. Firstly, multi-view knowledge consists of self-view and consensus-view knowledge, which are one-hop triplets in the knowledge graph. However, not all question-choice pairs have one-hop triplets, leading to null knowledge being retrieved. Additionally, excessive consensus-view knowledge can lead to noisy retrieved knowledge. Therefore, our knowledge retrieval system needs further improvement to obtain sufficient, high-quality knowledge. Secondly, we focus on the empirical study of prompt tuning in commonsense reasoning tasks. Although we conduct extensive experiments, including initialization schemes and prefix token length, we do not fully understand the mechanism behind prompt tuning and sometimes experience unstable performance. Although prompt tuning has been proven to be an efficient tuning paradigm for commonsense reasoning tasks, it requires further exploration.

Acknowledgements

Liwei Wang is also a Principal Investigator of Centre for Perceptual and Interactive Intelligence Limited (CPII). This work is supported in part by CPII, in part by the UGC under Research Matching Grant Scheme and Direct Grant at CUHK.

References

Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Pratyay Banerjee and Chitta Baral. 2020. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *arXiv preprint arXiv:2005.00660*.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2020. From 'f'to 'a'on the ny regents science exams: An overview of the aristo project. *AI Magazine*, 41(4):39–53.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *CoRR*, abs/2205.12548.

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*.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020a. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1295–1309.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Zixian Huang, Ao Wu, Jiaying Zhou, Yu Gu, Yue Zhao, and Gong Cheng. 2022. Clues before answers: Generation-enhanced multiple-choice QA. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3272–3287, Seattle, United States. Association for Computational Linguistics.

Wayne Xin Zhao Jinhao Jiang, Kun Zhou and Ji-Rong Wen. 2022. Great truths are always simple: A rather simple knowledge encoder for enhancing the commonsense reasoning capacity of pre-trained models. In *North American Chapter of the Association for Computational Linguistics-Findings(NAACL-Findings)*.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of EMNLP*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

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

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. 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. 2021c. 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. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020a. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020b. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8449–8456.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*.

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 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. *arXiv preprint arXiv:1805.07858*.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2020. How additional knowledge can improve natural language commonsense question answering?

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

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 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4967–4976.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472.

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 - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843, pages 593–607.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

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

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Olga Vechtomova and Ying Wang. 2006. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4):324–333.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. 2021a. GNN is a counter? revisiting GNN for question answering. *CoRR*, abs/2110.03192.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. 2021b. Gnn is a counter? revisiting gnn for question answering. *arXiv preprint arXiv:2110.03192*.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. *arXiv preprint arXiv:2005.00691*.

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng, Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng, and Xuedong Huang. 2021a. Human parity on commonsenseqa: Augmenting self-attention with external attention. *arXiv preprint arXiv:2112.03254*.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021b. Fusing context into knowledge graph for commonsense question answering. In *Association for Computational Linguistics (ACL)*.

Jun Yan, Mrigank Raman, Aaron Chan, Tianyu Zhang, Ryan Rossi, Handong Zhao, Sungchul Kim, Nedim Lipka, and Xiang Ren. 2020. Learning contextualized knowledge structures for commonsense reasoning. *arXiv preprint arXiv:2010.12873*.

Ping Yang, Junjie Wang, Ruyi Gan, Xinyu Zhu, Lin Zhang, Ziwei Wu, Xinyu Gao, Jiaxing Zhang, and Tetsuya Sakai. 2022. Zero-shot learners for natural language understanding via a unified multiple choice perspective. *arXiv preprint arXiv:2210.08590*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 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 for question answering. *arXiv preprint arXiv:2201.08860*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*.

A Appendix

A.1 Hyperparameter Settings for Datasets and Models

Table 11 shows hyperparameter settings for datasets and models.

A.2 Details of PLM+KG Baselines

- RN (Santoro et al., 2017) utilizes a relational reasoning structure in order to incorporate information from a commonsense knowledge graph (CSKG).
- RGCN (Schlichtkrull et al., 2018) uses a graph concept attention model to gather entity data from the CSKG.
- GconAttn (Lin et al., 2019) enhances the GCN (Kipf and Welling, 2016) by adding relation-specific weights.
- MHGRN (Feng et al., 2020a) is a GNN architecture that uses both GNNs and path-based models to reason over the CSKG.
- QAGNN (Yasunaga et al., 2021) employs a GAT (Veličković et al., 2017) to jointly reason over the CSKG and incorporate information from the CSKG into its processing.
- GSC (Wang et al., 2021b) utilizes a simple graph neural counter as the KG encoder in order to incorporate knowledge from the CSKG.
- GreaseLM (Zhang et al., 2022) combines encoder representations from a pre-trained language model (PLM) and KG encoder through the use of multiple modality interaction layers, allowing for the integration of knowledge from the CSKG into the PLM's processing.
- SAFE (Jinhao Jiang and Wen, 2022) merely utilize MLP-based KG encoder to extract features from relation paths in the retrieved multi-hop knowledge subgraph.

A.3 Training Curve Analysis

We additionally investigate the learning of our MVP-Tuning. We compare the training curves of prompt-tuning and fine-tuning with multi-view knowledge retrieval and a backbone PLM Roberta-large. Figure 5 demonstrates that the fine-tuning approach converges rapidly and starts to overfit soon, where the val loss rises with fluctuations. On the other hand, prefix-tuning converges more slowly and smoothly due to its fewer trainable parameters.

A.4 Case Study

In Table 12, we provide two examples from CSQA to illustrate how the model may reason using retrieved multi-view knowledge to arrive at the correct answer. For the first question, self-knowledge helps eliminate the incorrect answer *be dismembered by a chainsaw*, as "child" is incapable of doing so. The consensus-view knowledge verifies the "Desires" relationship between"kids" and "play", indicating that "play tag" is the right response. Again, self-view knowledge excludes *hurt* from the second question, as there is no link between "hurt" and "having fun" in the CSKG. The consensus-view knowledge contains triplets whose tail entity is a synonym of "pleasure" such as "happiness" and "enjoyment", which helps to affirm the correct answer. This suggests that multi-view knowledge is essential for obtaining the correct answer. Multi-view knowledge retrieval facilitates model reasoning to choose the right candidate.

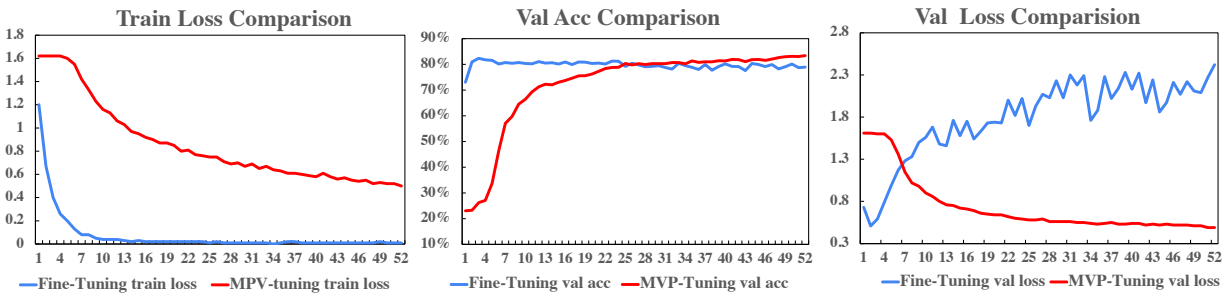

Figure 5: MVP-Tuning vs. Fine-Tuning with Multi-View Knowledge on CommonsenseQA dataset.

Models	Hyperparameter	OpenBookQA	CommonsenseQA	Other QA Datasets
Both **RoBERTa-large** and **AristoRoBERTa**	Batch Size	4	8	8
	Number of epochs	100	100	100
	Learning Rate	1e-3	1e-3	1e-3
	Optimizer	Adam	Adam	Adam
	Prefix Token Length	120	100	100
Both **DeBERTa-xlarge** and **DeBERTa-xxlarge**	Batch Size	2		
	Number of epochs	100		
	Learning Rate	2e-4		
	Optimizer	Adam		
	Prefix Token Length	100		

Table 11: Hyperparameter settings for different datasets and models. Other QA datasets include SocialIQA, PIQA and Riddle-Sense.

Question	A child wants to play, what would they likely want?
Candidates	*A) fall down*, B) breathe, **C) play tag**, D) be dismembered by a chainsaw, E) become adult
Self-View Knowledge	child CapableOf {fall down, breathe, play tag, become adult}
Consensus-View Knowledge	children Desires play sports; child CapableOf play video games; children Desires play ball
Question	What is the feeling of one having fun?
Candidates	A) *smiling*, **B) pleasure**, C) hurt, D) injuries, E) laughter
Self-View Knowledge	having fun HasSubevent {smiling, laughter}; having fun Causes {pleasure, injuries}
Consensus-View Knowledge	having fun Causes being happy; having fun Causes happiness; having fun Causes enjoyment; having fun Causes feeling happy}

Table 12: Case study of the effect of multi-view retrieval of knowledge. We list two questions from the CommonsenseQA dataset using the multi-view knowledge we have retrieved. The correct response is displayed in **bold**, and our MVP-Tuning strategy selected the correct answer for both questions. Choices in *italic* are the incorrect options that a RoBERTa-large model makes.

Question	What are candles good for eliminating?
Candidates	A) shelf, B) board, C) church, D) table, **E) dark**
Multi-View Knowledge Retrieval	Single View Knowledge: candle at location dark Consensus View Knowledge: candle CapableOf {light house, **emit light**}, candle AtLocation dimly lit room, light source AtLocation candle, lighting match Causes illumination
Multi-Hop Knowledge Retrieval	One hop knowledge: candle at location dark Two Hop knowledge: light antonym {dark, heavy}, candle isa light, good antonym evil, dark isa illumination
Question	What happens if someone kisses too long?
Candidates	A) strong feelings , B) herpes, **C) shortness of breath**, D) excitement, E) arousal
Multi-View Knowledge Retrieval	Single View knowledge: kissing causes shortness of breath Consensus View knowledge: kissing Causes {shyness, pleasurable, **sexual excitement**, happiness}, being in love CausesDesire kiss, person Desires passionate kisses
Multi-Hop Knowledge Retrieval	One Hop knowledge: kissing causes shortness of breath Two Hop knowledge: long antonym {short, brief}, shortness isa {length, duration}, kissing hassubevent kiss

Table 13: Comparing two knowledge retrieval schemes: multi-view knowledge retrieval and multi-hop knowledge retrieval. We list two questions from the CSQA dataset to compare our retrieved multi-view knowledge and with multi-hop knowledge.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The last setction

☒ A2. Did you discuss any potential risks of your work?
This paper does not have such risk since it is a multi-choice question answering setting.

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

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

B ☑ Did you use or create scientific artifacts?

Section 5 experiments

☑ B1. Did you cite the creators of artifacts you used?
Section 5 experiments

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5 experiments

☑ 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)?
Section 5 experiments

☑ 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?
Section 5 experiments

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5 experiments

☑ 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.
Section 5 experiments

C ☑ Did you run computational experiments?

Appendix A.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 5 experiments

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?
Appendix A.1

☑ 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 5 experiments

☑ 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 5 experiments

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

This paper does not involve human annotation or research with human subjects:

☒ 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.?
This paper does not involve human annotation or research with human subjects:

☒ 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)?
This paper does not involve human annotation or research with human subjects:

☒ 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?
This paper does not involve human annotation or research with human subjects:

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
This paper does not involve human annotation or research with human subjects:

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
This paper does not involve human annotation or research with human subjects: