Subgraph Retrieval Enhanced by Graph-Text Alignment for Commonsense Question Answering

Boci Peng[1], Yongchao Liu[2], Xiaohe Bo[3], Sheng Tian[2], Baokun Wang[2], Chuntao Hong[2], and Yan Zhang[1(✉)]

[1] School of Intelligence Science and Technology, Peking University, China
bcpeng@stu.pku.edu.cn, zhyzhy001@pku.edu.cn
[2] Ant Group, China
{yongchao.ly,tiansheng.ts,yike.wbk,chuntao.hct}@antgroup.com
[3] School of Artificial Intelligence, Beijing Normal University, China
xiaohe@mail.bnu.edu.cn

Abstract. Commonsense question answering is a crucial task that requires machines to employ reasoning according to commonsense. Previous studies predominantly employ an extracting-and-modeling paradigm to harness the information in KG, which first extracts relevant subgraphs based on pre-defined rules and then proceeds to design various strategies aiming to improve the representations and fusion of the extracted structural knowledge. Despite their effectiveness, there are still two challenges. On one hand, subgraphs extracted by rule-based methods may have the potential to overlook critical nodes and result in uncontrollable subgraph size. On the other hand, the misalignment between graph and text modalities undermines the effectiveness of knowledge fusion, ultimately impacting the task performance. To deal with the problems above, we propose a novel framework: **S**ubgraph R**E**trieval Enhanced by Gra**P**h-**T**ext **A**lignment, named **SEPTA**. Firstly, we transform the knowledge graph into a database of subgraph vectors and propose a BFS-style subgraph sampling strategy to avoid information loss, leveraging the analogy between BFS and the message-passing mechanism. In addition, we propose a bidirectional contrastive learning approach for graph-text alignment, which effectively enhances both subgraph retrieval and knowledge fusion. Finally, all the retrieved information is combined for reasoning in the prediction module. Extensive experiments on five datasets demonstrate the effectiveness and robustness of our framework.

Keywords: Commonsense Question Answering, Pre-trained Language Models, Graph Neural Networks

1 Introduction

Commonsense question answering (CSQA) is a critical task in natural language understanding, which requires systems to acquire different types of commonsense knowledge and possess multi-hop reasoning ability [27,19,22]. Though massive pre-trained models have achieved impressive performance on this task, it is difficult to learn commonsense knowledge solely from the pre-training text corpus,

as the commonsense knowledge is evident to humans and rarely expressed explicitly in natural language. Compared with unstructured text, structured data like knowledge graphs is much more efficient in representing commonsense [26]. The incorporation of external knowledge aids PLMs in comprehending question-answer (Q-A) pairs, while the entity relations enhance the model's reasoning capabilities. Therefore, various commonsense knowledge graphs (CSKGs) (e.g., ConceptNet [25]) have been adopted in previous studies.

Existing KG-augmented models for CSQA primarily adhere to a extracting-and-modeling paradigm [36,26,29,32,28,35]. First, the knowledge subgraphs or paths related to a given question are extracted by string matching or semantic similarity, which indicate the relations between concepts or imply the process of multi-hop reasoning. Subsequently, diverse strategies emerge for the efficient representation and fusion of the extracted structural knowledge. One research path [12,8] involves elaborately crafting graph neural networks for better modeling the extracted subgraphs, whereas another [34,26] explores the efficient incorporation of knowledge from KG into language models by enhancing the interactions between PLMs and GNNs.

Despite their success, these approaches still have several limitations. First, the subgraph's quality suffers when retrieved through a simple string or semantic matching, posing limitations for subsequent operations. To obtain sufficient relevant knowledge, the number of nodes will expand dramatically with the increase of hop count, inevitably raising the burden of the model. Despite its ample size, certain crucial nodes might remain elusive, since some entities are not learned during the pre-training. Besides, the edges linked to the peripheral nodes within the subgraph are pruned, causing the message-passing mechanism of GNN to be blocked and impairing the attainment of effective representations, consequently undermining valuable information. Second, the misalignment between graph and text encoders presents a challenge for PLMs to internalize the knowledge contained in the acquired subgraph, especially in scenarios with limited data, leading to a reduced task performance [35]. Though Dragon [33] proposes a pre-training method to align GNNs and PLMs, it requires additional corpus, and the text-to-graph style to construct semantically equivalent graph-text pairs is challenging. The necessity for substantial computational resources poses another hurdle, prompting the search for a more efficient alignment method.

In this paper, we propose a novel framework: **S**ubgraph R**E**trieval Enhanced by Gra**P**h-**T**ext **A**lignment (**SEPTA**), for CSQA. To mitigate the shortcomings of the subgraph extraction process, we establish a database of subgraph vectors derived from the knowledge graph. Consequently, the challenge shifts from retrieving a pertinent subgraph to obtaining relevant subgraph vectors. A BFS-style sampling method is employed to obtain the connected graph for each node and the embedding of the subgraph is subsequently stored in the database. Drawing on the parallels between BFS and the message-passing mechanism of GNNs, the central node's representation learned from the subgraph could be closely aligned with that derived from the entire graph, with almost no information loss. Besides, to further improve the retrieval accuracy and facilitate

knowledge fusion during the prediction, we consider aligning the semantic space of the graph and text encoders, proposing an effective approach for graph-text alignment. A novel graph-to-text method is proposed to construct high-quality semantically equivalent training pairs, with no requirement of external corpus and easy to train. Finally, all the information retrieved is combined by a simple attention mechanism to facilitate the model in commonsense reasoning.

Our contributions can be summarized as follows:

- We propose a novel and effective framework SEPTA, where we convert the knowledge graph into a subgraph vector database and retrieve relevant subgraphs to facilitate commonsense reasoning.
- We design a bidirectional contrastive learning method to align the semantic space of the graph and text encoders, with a graph-to-text method to construct high-quality graph-text pairs, which facilitates subgraph retrieval and knowledge fusion.
- We propose a BFS-style subgraph sampling strategy for subgraph construction. Drawing on the parallel between BFS and the message-passing mechanism, our method can preserve complete neighbor information for each node.
- We conduct extensive experiments on five datasets. Our proposed approach achieves better results than the state-of-the-art approaches and has promising performance in weakly supervised settings.

2 Related Work

2.1 Commonsense Question Answering

Commonsense question answering aims to evaluate the reasoning ability of models based on commonsense knowledge [7], e.g., physical commonsense [2]. To incorporate external knowledge and enhance reasoning ability, some works introduce commonsense knowledge graphs (CSKGs, e.g. ConceptNet [25]). Generally, these methods [34,37,36,26,29,12,33,8,32,28,35] extract relevant knowledge subgraphs through entity linking and adopt graph neural networks to learn knowledge representations. Among them, a category of research focuses on designing more efficient knowledge encoders. For example, SAFE [12] proposes a 2-layer MLP to improve the efficiency of graph encoding. HamQA [8] considers learning hierarchical structures in KGs with hyperbolic geometry. Another research line tries to enhance the interactions between PLMs and GNNs. For instance, QA-GNN [34] adds a QA context node to the retrieved subgraphs and incorporates relevant information from other entities. Unlike previous works, we convert the knowledge graph into a subgraph database and transform the task to a subgraph vector retrieval problem, thus bypassing the challenges inherent in the extracting-and-modeling paradigm.

2.2 Graph-Text Alignment

Aligning the embedding spaces of text encoders and graph encoders is an effective way to take the strengths of two modalities [18]. Previous alignment methods

can be roughly classified into two groups, i.e. the symmetric method and the asymmetric method, based on their training objectives. The symmetric alignments enhance each modality equally, most of which adopt a two-tower style and utilize contrastive learning techniques [5,3]. However, the asymmetric methods aim to take advantage of the capabilities of GNNs to reinforce PLMs. The predominant approaches can be categorized into two types, with the first type trying to enhance PLMs by inserting graph encoders into transformers [13,38] and the second type directly using GNNs as teacher models to generate soft labels for the PLMs [39]. In this paper, we leverage PLM as a teacher model and distill its knowledge into GNN, enabling us to retrieve related subgraphs in a manner akin to retrieving relevant text. Although DRAGON [33] also proposes a pre-training method to align PLMs and GNNs, our approach does not require additional corpus and demands lower computational costs.

3 Task Formulation

We study the multiple-choice CSQA [27,19], which can be formulated as: given a natural language question q and a set of answer candidates $C = \{c_1, c_2, \ldots, c_n\}$, the aim is to identify the optimal choice $c^* \in C$. Consistent with previous works [15], the CSQA problem is addressed in a *knowledge-aware* setting, that is, we can utilize external commonsense knowledge graphs (CSKGs) to facilitate model prediction. A CSKG can be formally described as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \boldsymbol{X})$, where $\mathcal{V}$ is the set of concept nodes (e.g., *Sun* and *Holiday*), $\mathcal{R}$ is the set of relation types (e.g., *HasProperty* and *AtLocation*), $\mathcal{E} \in \mathcal{V} \times \mathcal{E} \times \mathcal{V}$ is the link set of the knowledge graph (or fact triplets, e.g. *(House, MadeOf, Wood)*), and $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ denotes pre-trained embedddings of all concept nodes. Generally, the task can be treated as a score prediction task for each Q-A pair.

4 Methods

In this section, we will introduce the design of our SEPTA. Departing from previous extracting-and-modeling approaches, we reframe the task as a subgraph vector retrieval problem and propose a graph-text alignment method to improve the retrieval accuracy and facilitate knowledge fusion for prediction. For ease of exposition, we first introduce the graph-text alignment process in Section 4.1. Then with the aligned encoder, the subgraph vector database is constructed and retrieved, which will be presented in Section 4.2. Finally, how to combine all the structural information retrieved for answer prediction is discussed in Section 4.3. Figure 1 shows the overview of our SEPTA.

4.1 Graph-Text Alignment

To coordinate the embedding spaces of graph and text encoders and fully harness the respective strengths of text and KG, we propose an alignment process

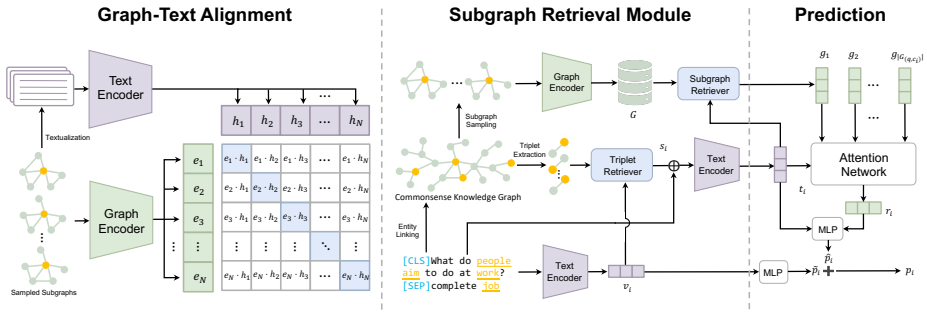


Fig. 1. The overview of our proposed SEPTA. First, a bidirectional contrastive method is proposed to align the semantic space of graph and text encoders. With the encoders aligned, we then transform the knowledge graph into a subgraph vector database and introduce a query enhancement strategy for better subgraph retrieval. Finally, all the information retrieved is combined by a simple attention mechanism to bolster the reasoning ability of PLMs for CSQA.

before downstream tasks. In our method, we initially address the challenge of generating training graph-text pairs with equivalent semantics and subsequently employ a bidirectional contrastive learning method to train the encoders of both modalities. The alignment process plays a pivotal role, as for one thing, it decides the efficacy of retrieving question-related subgraphs from the vector base, and for another, it determines the successful integration of graph information with the question context during the prediction phase.

Construction of Graph-Text Pairs The construction of high-quality semantically equivalent graph-text pairs is crucial for the alignment process, yet not that straightforward. Previous methods mostly adopt a text-to-graph approach to construct training pairs, where the goal is to discover a graph structure that corresponds to the semantics of a provided text segment. However, utilizing existing transformation tools e.g. dependency graph could not well accommodate the downstream subgraph retrieval, while rule-based methods to extract text-related subgraphs from CSKGs are challenging. It is also notably time-consuming and laborious to construct through manual annotation. Therefore, in this paper, we propose a graph-to-text approach and consider constructing synonymous text descriptions of the subgraphs.

Specifically, we propose a BFS-style sampling strategy for subgraph construction, which initiates from the central node and proceeds to sample neighbors layer by layer. During the process, to address the challenge of an excessive number of neighbors, we set p as the probability for immediate neighbor selection. In addition, since it is sufficient to describe local neighborhoods for determining structural equivalence [10], we set a parameter d to constrain the depth of the sampled subgraphs. By restricting the search to nearby nodes, our sam-

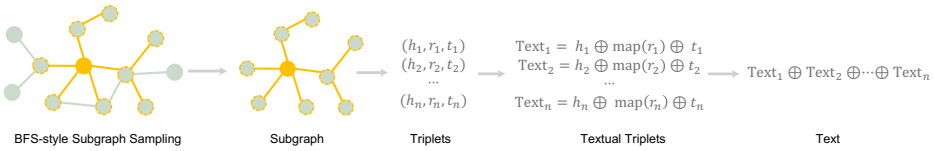


Fig. 2. The overview of the construction of graph-text pairs.

pling method achieves this characterization and obtains a microscopic view of the neighborhood of every node. Furthermore, a parameter n is established to regulate the size of the subgraphs. Once the number of sampled nodes reaches n, the layer-wise sampling is halted. Since nodes in the sampled neighbors tend to repeat many times, our method could reduce the variance in characterizing the distribution of 1-hop nodes with respect to the source node. The process can be formulated as:

$$\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{A}_i, \boldsymbol{X}_i) = \text{BFS}(v_i, p, d, n), \tag{1}$$

where $\mathcal{G}_i$ is the connected graph obtained, $\mathcal{V}_i$, $\mathcal{E}_i$ are sets of concept nodes and relation links in $\mathcal{G}_i$, $\boldsymbol{A}_i$ represents the adjacent matrix, $\boldsymbol{X}_i$ denotes embeddings of concept nodes, and v_i is the central concept node.

After that, it is necessary to textualize the subgraphs to construct synonymous text descriptions. The first step is to convert all relation links into triplet descriptions, which are later combined to compose the final description. Specifically, to transform relation links into sentences, we first map each relation type to a relation template and then concatenate the head concept, relation template, and tail concept as the description of each fact triplet. The textualization process can be denoted as:

$$s_i = \bigoplus_{e_j \in \mathcal{E}_i} \text{TEXT}(e_j), \tag{2}$$

where s_i is the description of the graph $\mathcal{G}_i$ and $\oplus$ denotes the concatenation of sentences. Therefore, the training set can be denoted as $\{(\mathcal{G}_i, s_i)\}_{i=1}^{n}$. The overview of the construction process is shown in Figure 2.

Graph-Text Contrastive Learning The graph-text alignment procedure is presented in the left part of Figure 1. First, GNN and PLM are utilized to encode the knowledge subgraphs and natural language descriptions to obtain the corresponding representation, respectively, which can be formulated as:

$$\begin{aligned} \tilde{\boldsymbol{e}}_{\boldsymbol{i}} &= \text{Pool}_G(\text{GNN}(\mathcal{G}_i)), \\ \tilde{\boldsymbol{h}}_{\boldsymbol{i}} &= \text{Pool}_T(\text{PLM}(s_i)), \end{aligned} \tag{3}$$

where $\tilde{\boldsymbol{e}}_{\boldsymbol{i}}$ is the average of all nodes' embeddings and $\tilde{\boldsymbol{h}}_{\boldsymbol{i}}$ is the representation of the [CLS] token. To project $\tilde{\boldsymbol{e}}_{\boldsymbol{i}}$ and $\tilde{\boldsymbol{h}}_{\boldsymbol{i}}$ into the same semantic space, two linear

projection layers are designed as follows:

$$\begin{aligned} \boldsymbol{e}_i &= \boldsymbol{W}_G \tilde{\boldsymbol{e}}_{\boldsymbol{i}} + \boldsymbol{b}_G, \\ \boldsymbol{h}_i &= \boldsymbol{W}_T \tilde{\boldsymbol{h}}_{\boldsymbol{i}} + \boldsymbol{b}_T, \end{aligned} \tag{4}$$

where $\boldsymbol{W}_G$, $\boldsymbol{W}_T$ and $\boldsymbol{b}_G$, $\boldsymbol{b}_T$ are the transform matrices and biases of the linear projection layers.

We then employ InfoNCE with in-batch negative sampling to align the representations of two modalities bidirectionally. The graph-to-text contrastive loss can be formulated as:

$$\mathcal{L}_{\text{G2T}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(sim(\boldsymbol{e}_i, \boldsymbol{h}_i)/\tau)}{\sum_{j=1}^{N} \exp(sim(\boldsymbol{e}_i, \boldsymbol{h}_j)/\tau)}, \tag{5}$$

where τ is a temperature coefficient and N is the number of instances in a batch. Besides, function $sim(\cdot, \cdot)$ measures the similarity between two representations, which can be calculated by:

$$sim(\boldsymbol{e}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{i}}) = \frac{\boldsymbol{e}_{\boldsymbol{i}}^T \boldsymbol{h}_i}{\|\boldsymbol{e}_{\boldsymbol{i}}\| \cdot \|\boldsymbol{h}_{\boldsymbol{i}}\|}. \tag{6}$$

Similarly, we also design a text-to-graph contrastive loss for uniformly aligning into the same semantic space, which is shown as:

$$\mathcal{L}_{\text{T2G}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(sim(\boldsymbol{h}_i, \boldsymbol{e}_i)/\tau)}{\sum_{j=1}^{N} \exp(sim(\boldsymbol{h}_i, \boldsymbol{e}_j)/\tau)}. \tag{7}$$

The final contrastive loss $\mathcal{L}_{GT}$ is defined as the average of $\mathcal{L}_{G2T}$ and $\mathcal{L}_{T2G}$:

$$\mathcal{L}_{GT} = \frac{1}{2}(\mathcal{L}_{G2T} + \mathcal{L}_{T2G}). \tag{8}$$

Remarks (1) To avoid the loss of inherent knowledge caused by over-fitting of the PLM, only the GNN and the linear projection layers are trainable during actual implementation. From another perspective, we essentially distill the semantic information from the PLM into the GNN, enabling the graph representations encoded by the GNN to encompass both structural and textual information. (2) As our ultimate goal is to obtain subgraph representations, we employ graph-level contrastive learning to align subgraph embeddings with text embeddings, yielding promising results. We also attempt other granular alignment signals, such as aligning entity node representations with text representations or applying the Masked Language Model (MLM) to text based on subgraph representations. However, with our current computational resources, these methods are unable to converge effectively. Through our contrastive learning, the model is able to rapidly converge and achieve promising performance.

4.2 Subgraph Retrieval Module

In this section, we initially present the establishment of the subgraph vector database. After that, to better accommodate the alignment process, we propose the query enhancement. Finally, we will outline the subgraph retrieval procedure.

Database Construction Previous methodologies primarily adopt an extracting-and-modeling paradigm for knowledge subgraph retrieval. However, as the cornerstone of subsequent work, the retrieval of high-quality subgraphs proves to be challenging. Consequently, we suggest transforming the knowledge graph into a subgraph vector database, thereby transitioning the focus towards retrieving pertinent subgraphs. Leveraging the analogy between BFS and the message-passing mechanism, we adopt a BFS-style subgraph sampling strategy to construct subgraphs, instead of DFS or Random Walk. On the one hand, each subgraph contains complete neighbor information for at least one node, and on the other hand, each node appears in at least one subgraph. Therefore, the information supplied for our retrieval is complete, and each subgraph vector holds fine-grained knowledge regarding the central node. Specifically, we first apply the same method as in Section 4.1 to produce the graph embedding $\boldsymbol{e}_i$ and the text embedding $\boldsymbol{h}_i$. Then we add them up as the subgraph vector $\boldsymbol{g}_i \in \mathbb{R}^d$:

$$\boldsymbol{g}_i = \frac{1}{2}\left(\frac{\|\boldsymbol{h}_i\|}{\|\boldsymbol{e}_i\|}\boldsymbol{e}_i + \boldsymbol{h}_i\right), \tag{9}$$

where the regularization coefficient $\frac{\|\boldsymbol{h}_i\|}{\|\boldsymbol{e}_i\|}$ maintains the consistency between the norm of the subgraph vectors and the norm of the text representations, preventing the prediction from relying predominantly on the features with larger norms. Finally, a subgraph vector database $\boldsymbol{G} = \{\boldsymbol{g}_i\}_{i=1}^{|G|}$ is constructed with all subgraph vectors. Note that we only need to generate subgraph vectors once before performing downstream tasks, which saves computational resources.

Query Enhancement Given a problem, we need to find the relevant subgraph vectors. An intuitive method is to apply the embedding of the question-answer pair as a query. However, there is a certain difference between such textual query and the pre-trained corpus of the aligned encoder, as the latter is constructed through triplet concatenation, which makes it difficult to ensure the quality of text encoding and reduces the accuracy of the retrieval process. Therefore, we propose to enhance the query by retrieving question-related triplets in the knowledge graph and concatenating them after the Q-A pairs. Specifically, given a pair of question-answers (q, c_i), we first apply entity linking to find all entities $E_q = \{e_q^{(1)}, e_q^{(2)}, \ldots, e_q^{(n_q)}\}$, $E_{c_i} = \{e_{c_i}^{(1)}, e_{c_i}^{(2)}, \ldots, e_{c_i}^{(n_{c_i})}\}$ appearing in question q and choice c, respectively. Then, we find all triplets in the CSKG containing the entities in E_q and E_{c_i}, which can be formulated as $T = \{(e^*, r, e), (e, r, e^*) | e \in E_q \cup E_{c_i}\}$. All fact triplets in T are serialized to natural language sentences and a pre-trained dense retriever is adopted to find the most relevant ones. We

concatenate the fact triplets retrieved together, along with the questions and options: $s_i = q \oplus c_i \oplus \text{text}_1 \oplus \text{text}_2 \oplus \cdots \oplus \text{text}_K$. The aligned PLM is then utilized to encode s_i to $\boldsymbol{t_i} = \text{PLM}(s_i) \in \mathbb{R}^d$.

Subgraph Retrieval After that, we employ the embedding of Q-A pairs concatenated with factual triples to retrieve the relevant subgraph vectors from the subgraph vector database. As the embedding space of two modalities has been aligned, the cosine similarity of $\boldsymbol{t}_i$ with each subgraph vector $\boldsymbol{g}_i$ in $\boldsymbol{G}$ is competent for the retrieval. We recall the top k subgraph vectors with the highest similarities, which is denoted as $\boldsymbol{G}_{q,c_i} \in \mathbb{R}^{k \times d}$.

4.3 Prediction

We combine all the knowledge retrieved to make the final predictions. We first integrate the retrieved subgraph vectors through multi-head attention with $\boldsymbol{t}_i$ as the query, which can be formulated as:

$$\begin{aligned} \boldsymbol{\alpha}_i^{(h)} &= \frac{(\boldsymbol{t}_i \boldsymbol{W}_Q^{(h)})(\boldsymbol{G}_{q,c_i} \boldsymbol{W}_K^{(h)})^T}{\sqrt{d}}, \\ \boldsymbol{r}_i^{(h)} &= \text{Softmax}(\boldsymbol{\alpha}_i^{(h)})(\boldsymbol{G}_{q,c_i} \boldsymbol{W}_V^{(h)}), \\ \boldsymbol{r}_i &= \text{Concat}(\boldsymbol{r}_i^{(1)}, \boldsymbol{r}_i^{(2)}, \ldots, \boldsymbol{r}_i^{(H)}) \boldsymbol{W}_O, \end{aligned} \tag{10}$$

where $\boldsymbol{W}_Q^{(h)}, \boldsymbol{W}_K^{(h)}, \boldsymbol{W}_V^{(h)} \in \mathbb{R}^{d \times d}$ are projection matrices under head h.

Subsequently, $\boldsymbol{r}_i$ and $\boldsymbol{t}_i$ are added and fed into a linear layer to predict the score of option c_i:

$$\hat{p}_i = \boldsymbol{W_1}^T (\boldsymbol{t}_i + \boldsymbol{r}_i) + b_1. \tag{11}$$

Since some questions are expected to be answered based solely on the question context, we also encode the Q-A pair (q, c_i) to infer directly:

$$\begin{aligned} \boldsymbol{v_i} &= \text{PLM}(q, c_i), \\ \tilde{p}_i &= \boldsymbol{W_2}^T \boldsymbol{v}_i + b_2. \end{aligned} \tag{12}$$

The two scores are weighted and summed to yield the final score:

$$p_i = \lambda \hat{p}_i + (1 - \lambda) \tilde{p}_i, \tag{13}$$

where λ is the hyper-parameter for the balance.

During the training phase, we employ the softmax function to normalize the score for each choice and optimize the model by cross-entropy loss. For inference, we determine the prediction by selecting the choice with the highest score.

Table 1. Statistics of the datasets. '-' denotes the unavailable dataset split.

Task	Train	Dev	Test
CommonsenseQA official split	9,741	1,221	1,140
CommonsenseQA 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	-

5 Experiments

5.1 Datasets

We conduct experiments to evaluate our method on five CSQA datasets, which are shown in Table 1:

• **CommonsenseQA** [27] is a 5-way multiple-choice QA dataset, which is created based on ConceptNet [25]. Due to the dual split of CommonsenseQA: the official split [27] and the in-house (IH) split [15], we report the results for both settings. For the official split, the ground truth of the test set is not publicly available, so we submit our model's predictions to the official leaderboard[4] to evaluate the test accuracy.

• **OpenBookQA** [19] is a 4-choice dataset about elementary science questions to evaluate the science commonsense knowledge. We also submit the predictions of the test set to the official leaderboard[5].

• **SocialIQA** [22] is a 3-choice dataset to evaluate the understanding of commonsense social knowledge. Due to the unavailability of the test set, consistent with prior works [24], we report the accuracy of the development set.

• **PIQA** [2] is a 2-choice QA dataset regarding physical commonsense. Since the test set is hidden, evaluations are conducted on the development set.

• **RiddleSenseQA** [16] is a 5-choice QA dataset about commonsense riddles. Because the test set is not released, we only report the validation accuracy.

5.2 Baselines

We compare with the mainstream RoBERTa-Large + GNN methods, including RN [21], RGCN [23], GconAttn [31], MHGRN [9], QA-GNN [34], DGRN [37], GreaseLM [36], JointLK [26], GSC [29], SAFE [12], DRAGON [33], HamQA [8], and DHLK [32]. Among them, DRAGON introduces BookCorpus to joint-train GNN and PLM, and DHLK additionally retrieves paraphrases of key entities in WordNet and Wiktionary.

[4] https://www.tau-nlp.sites.tau.ac.il/csqa-leaderboard
[5] https://leaderboard.allenai.org/open_book_qa/submissions/public

5.3 Implementation Details

According to the previous works, we use RoBERTa-Large [17] as the text encoder and use GraphGPS [20] for the graph encoder. We also test AristoRoBERTa [6] for OpenBookQA. We use ConceptNet [25], including 799,273 nodes and 2,487,003 edges, as the commonsense knowledge graph. For each node, we treat it as the center and employ BFS to obtain the subgraph, which is then translated into the corresponding natural language description.

In the graph-text alignment phase, we randomly sample 64,000 graph-text pairs to train, and sample 16,000 pairs to evaluate two encoders. We fix the learning rate to 1e-3, the number of GNN layers to 2, and the dimensions of all embeddings to 1024. During the fine-tuning stage, we set the number of fact triplets to 10, tune the number of retrieved subgraph vectors k in $\{10, 30, 50, 70, 100\}$, the batch size in $\{4, 8, 16\}$, the balance coefficient λ from 0.1 to 1.0, and the learning rate in {2e-5, 1e-5, 5e-6, 2e-6, 1e-6}. The parameters of the model are optimized by RAdam. We train 30 epochs until the performance does not improve on the development sets for 3 consecutive epochs. We use the default parameter settings as their original implementations for the baseline methods. We conduct all experiments on NVIDIA A100-40GB GPUs.

5.4 Main Results

Following previous works [34,29,12,11], we compare our method with different baselines on CommonsenseQA and OpenBookQA as main results, which are shown in Table 2. The best and runner-up results in each column are highlighted in bold and underlined, respectively.

From the results, we can observe: (1) Our method can contribute performance gains to LMs, which improves 6.54% and 6.09% on IHdev and IHtest of CommonsenseQA compared to fine-tuned RoBERTa. (2) SEPTA outperforms all baselines without additional corpus on both datasets. For example, compared to the GSC method, our method improves by 2.00% and 0.70% on OpenBookQA using RoBERTa and AristoRoBERTa, respectively. (3) Compared to baselines incorporating additional corpus, our method also achieves comparable performance. Specifically, we surpass DHLK on both datasets and DRAGON on OpenBookQA and slightly lag behind DRAGON on CommonsenseQA. It should be noted that the DRAGON undergoes MLM training on the BookCorpus dataset and requires training on 8×A100 GPUs for a week [33,32]. By eliminating the MLM, our SEPTA model demonstrates a definitive enhancement.

In Table 3, we evaluate SEPTA on the official CommonsenseQA and OpenBookQA leaderboards (as of March 22, 2024). Our method achieves results surpassing all baselines based on the same PLM and exhibits comparative performance compared with methods with larger-scale parameters (e.g., UnifiedQA).

To comprehensively evaluate the efficiency of SEPTA, we extend our comparative analysis to other commonsense reasoning datasets originating from diverse domains or tasks. As shown in Table 4, our SEPTA consistently achieves superior

Table 2. Evaluation on CommonsenseQA (in-house split) and OpenBookQA. We use RoBERTa-Large as the text encoder in CommonsenseQA, and use RoBERTA-Large and AristoRoBERTa in OpenBookQA. Methods with AristoRoBERTa use the textual evidence by [6] as an additional input to the QA context. The baselines incorporating extra corpus are marked with *.

Methods	CommonsenseQA		OpenBookQA	
	IHdev-Acc (%)	IHtest-Acc (%)	RoBERTa-Large (%)	AristoRoBERTa (%)
Fine-tuned LMs	73.07 (±0.45)	68.69 (±0.56)	64.80 (±2.37)	78.40 (±1.64)
+ RN	74.57 (±0.91)	69.08 (±0.21)	65.20 (±1.18)	75.35 (±1.39)
+ RGCN	72.69 (±0.19)	68.41 (±0.66)	62.45 (±1.57)	74.60 (±2.53)
+ GconAttn	72.61 (±0.39)	68.59 (±0.96)	64.75 (±1.48)	71.80 (±1.21)
+ MHGRN	74.45 (±0.10)	71.11 (±0.81)	66.85 (±1.19)	80.60
+ QA-GNN	76.54 (±0.21)	73.41 (±0.92)	67.80 (±2.75)	82.77 (±1.56)
+ DGRN	78.20	74.00	69.60	84.10
+ GreaseLM	78.50 (±0.50)	74.20 (±0.40)	68.80 (±1.75)	84.80
+ JointLK	77.88 (±0.25)	74.43 (±0.83)	70.34 (±0.75)	84.92 (±1.07)
+ GSC	79.11 (±0.22)	74.48 (±0.41)	70.33 (±0.81)	86.67 (±0.46)
+ SAFE	76.93 (±0.37)	74.03 (±0.43)	69.20	<u>87.13</u>
+ HamQA	76.88	73.91	71.12	84.59
+ DRAGON*	-	**76.00**	72.00	-
+ DRAGON (w/o MLM)*	-	73.80	66.40	-
+ DHLK*	<u>79.39</u> (±0.24)	74.68 (±0.26)	<u>72.20</u> (±0.40)	86.00 (±0.79)
+ SEPTA (**Ours**)	**79.61** (±0.17)	<u>74.78</u> (±0.23)	**72.33** (±0.35)	**87.37** (±0.51)

Table 3. Performance comparison on CommonsenseQA (left) and OpenBookQA (right) official leaderboard.

Methods	**Test-Acc (%)**
RoBERTa [17]	72.1
RoBERTa+FreeLB	72.2
RoBERTa+HyKAS	73.2
RoBERTa+KE	73.3
RoBERTa+KEDGN	74.4
RoBERTa+MHGRN [9]	75.4
RoBERTa+QA-GNN [34]	76.1
RoBERTa+GSC [29]	76.2
Albert	73.5
ALBERT+Path Generator [30]	75.6
ALBERT+HGN [9]	77.3
UnifiedQA (11B) [14]	**79.1**
RoBERTa+SEPTA (**Ours**)	76.6

Methods	**Test-Acc (%)**
Careful Selection [1]	72.0
AristoRoBERTa [6]	77.8
KF+SIR	80.0
AristoRoBERTa+PG [30]	80.2
AristoRoBERTa+MHGRN [9]	80.6
AristoRoBERTa+QA-GNN [34]	82.8
AristoRoBERTa+GreaseLM [36]	84.8
AristoRoBERTa+GSC [29]	87.4
AristoRoBERTa+MVP-Tuning [11]	87.6
ALBERT + KB	81.0
T5	83.2
UnifiedQA (11B) [14]	87.2
AristoRoBERTa+SEPTA (**Ours**)	**87.8**

performance. This observation underscores the overall effectiveness of SEPTA in addressing various commonsense reasoning datasets or tasks, demonstrating a unified methodology.

5.5 Ablation Study

We conduct an ablation study on CommonsenseQA and OpenBookQA to explore the effectiveness of each component of SEPTA. We remove the alignment process (w/o alignment), retrieved subgraph vectors (w/o subgraph), fact triplets (w/o

Table 4. Performance comparison on SocialIQA, PIQA, and RiddleSenseQA.

Methods	SocialIQA	PIQA	RiddleSenseQA
RoBERTa-Large	78.25	77.53	60.72
+ GconAttn	78.86	78.24	61.77
+ RN	78.45	76.88	62.17
+ MHGRN	78.11	77.15	63.27
+ QA-GNN	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
+ SEPTA (**Ours**)	**79.21**	**80.85**	**67.62**

Table 5. Ablation study on CommonsenseQA (IHtest) and OpenBookQA datasets. The values in parentheses denote the extent of performance decline.

Ablation	CommonsenseQA	OpenBookQA
SEPTA	74.78	72.33
w/o alignment	69.83 (-4.95)	67.20 (-5.13)
w/o subgraph	72.34 (-2.44)	70.23 (-2.10)
w/o triplets	71.25 (-3.53)	69.67 (-2.66)
$\lambda = 1.0$	74.13 (-0.65)	70.47 (-1.86)

triplets), and scores predicted based on Q-A pairs (i.e. set $\lambda = 1.0$), respectively.

As shown in Table 5, four components are all crucial for SEPTA, and removing any part will result in a decrease in performance. Specifically, the performance drops the most significantly when we remove the graph-text alignment. This is because if the representations of graphs and texts are not semantically aligned, then during the knowledge retrieval stage, the retrieved subgraph vectors may be irrelevant and situated in different latent spaces from the textual information. Moreover, removing either fact triplets or subgraph vectors will affect the performance. On one hand, they represent different aspects of information, with the former providing more specific knowledge and the latter describing more comprehensive relationships between entities. On the other hand, fact triplets also play an auxiliary role in retrieving relevant subgraph vectors. Furthermore, only using knowledge-enhanced representations for predictions (i.e. $\lambda = 1.0$) cannot achieve optimal results. This is because some questions do not require additional knowledge, or relevant information cannot be found in CSKGs, which may instead become interference.

5.6 Low-Resource Setting

To evaluate the robustness of SEPTA, we conduct extensive experiments in low-resource settings, with different proportions of training data, including 5%, 10%, 20%, 50%, and 80%, in CommonsenseQA (IHtest) and OpenBookQA.

Table 6. Performance with different proportions of training data.

Methods	CommonsenseQA						OpenBookQA					
	5%	10%	20%	50%	80%	100%	5%	10%	20%	50%	80%	100%
RoBERTa-large	29.66	42.84	58.47	66.13	68.47	68.69	37.00	39.4	41.47	53.07	57.93	64.8
+ RGCN	24.41	43.75	59.44	66.07	68.33	68.41	38.67	37.53	43.67	56.33	63.73	62.45
+ GconAttn	21.92	49.83	60.09	66.93	69.14	68.59	38.60	36.13	43.93	50.87	57.87	64.75
+ RN	23.77	34.09	59.90	65.62	67.37	69.08	33.73	35.93	41.40	49.47	59.00	65.20
+ MHGRN	29.01	32.02	50.23	68.09	70.83	71.11	38.00	36.47	39.73	55.73	55.00	66.85
+ QA-GNN	32.95	37.77	50.15	69.33	70.99	73.41	33.53	35.07	42.40	54.53	52.47	67.80
+ GreaseLM	22.80	56.16	63.09	70.56	73.41	74.20	39.00	39.60	42.20	57.87	65.13	68.80
+ GSC	31.02	35.07	65.83	70.94	73.82	74.48	29.60	41.80	42.40	58.03	65.97	70.33
+ SAFE	36.45	56.51	65.16	70.72	73.22	74.03	38.80	41.20	44.93	58.33	65.60	69.20
+ SEPTA(**Ours**)	**50.69**	**62.37**	**68.09**	**71.80**	**74.05**	**74.78**	**45.63**	**54.80**	**58.10**	**66.57**	**68.30**	**72.33**

Table 7. Effect of different graph encoders.

GNN	**CommonsenseQA**	**OpenBookQA**
GraphGPS	74.78	72.33
FILM-GNN	74.67	72.17
RGCN	74.51	71.87

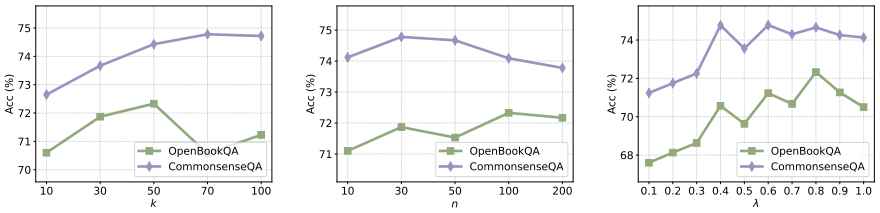
From the results in Table 6, we can observe that our SEPTA achieves promising performance in all settings, and it exhibits a trend where the performance improvement relative to other baselines is more significant with fewer training data. This is because we align text representations with graph representations before fine-tuning on downstream tasks, enabling retrieved subgraph vectors to integrate well with text representations even in low-resource settings. In contrast, other baselines make it hard to project subgraph representations and text representations into the same semantic space when training data is limited, resulting in structure embeddings becoming a noise that interferes with PLM reasoning.

5.7 Evaluation with Other GNNs

To demonstrate the generality of SEPTA, We employ GraphGPS [20], FILM-GNN [4], and RGCN [23] as the graph encoders, respectively. Table 7 illustrates the results on CommonsenseQA and OpenBookQA. From the results, we can observe that different graph encoders achieve competitive results on both datasets, with their performances being relatively close (within a difference of around 0.5%), which demonstrates the effectiveness and robustness of SEPTA.

5.8 Hyper-parameter Analysis

We further conduct in-depth analyses to investigate the impact of hyper-parameters. With other parameters fixed, we compare the effect of the number of retrieved subgraph vectors k, the maximum number of nodes n in each subgraph, and



(a) Effect of k. (b) Effect of n. (c) Effect of λ.

Fig. 3. Hyper-parameter analysis.

the balance coefficient λ. The results on CommonsenseQA (IHtest) and OpenBookQA datasets are reported in Figure 3.

Effect of subgraph vectors number k From the results, we observe that the accuracy initially ascends with the increase in the number of retrieved subgraph vectors, achieving its peak before subsequently declining. This is because fewer subgraph vectors may lose crucial commonsense, while an excess of subgraph vectors could introduce irrelevant information.

Effect of maximum number of nodes n Based on the results, the performance of the model initially increases with the increment of n, reaching a peak, then decreases. It might be attributed to the fact that when n is relatively small, subgraphs are unable to fully encompass the neighbor information of the central nodes, leading to the inability to acquire sufficient relevant knowledge during the subgraph retrieval phase. Conversely, when n is excessively large, each subgraph may contain a significant amount of information irrelevant to the central node, resulting in overall information redundancy.

Effect of the balance coefficient λ λ controls the proportion of inference based on the retrieved knowledge. When λ is small, the model primarily relies on its own knowledge for inference, which may lead to a lack of relevant information for some questions. However, when λ is large, the model heavily depends on retrieved knowledge to derive answers, although many questions need to be resolved according to the question context. Therefore, in terms of results, the accuracy generally increases initially with the increase in λ and then decreases.

6 Ethical Considerations and Limitations

6.1 Ethical Considerations

Our work proposes a novel and effective framework to combine PLMs and external knowledge graphs for commonsense question answering. However, potential

issues may arise from the utilization of PLMs and CSKGs. On one hand, PLMs tend to encapsulate certain biases present in the pre-training data. On the other hand, CSKGs may harbor biased concepts stemming from human annotations. To alleviate these biases, the implementation of appropriate screening rules offers a promising approach, e.g., filter biased concepts during the subgraph extraction process from the CSKG. Although a comprehensive analysis of such biases is not included in our work, it is imperative to implement supplementary measures before deploying the system in real-world scenarios.

6.2 Limitations

We propose a subgraph retrieval enhanced by a graph-text alignment framework named SEPTA for commonsense question answering. However, there are still limitations that demand resolution. Firstly, the corresponding text generated by rules from knowledge subgraphs still exhibits disparities from natural language. One possible solution is to reorganize the text using LLMs, but the cost is prohibitively high. Therefore, acquiring large-scale, high-quality graph-text pairs remains an ongoing challenge. Secondly, the number of retrieved subgraph vectors is required to tune according to the accuracy of development sets, which is time-consuming. Designing a module to automatically select the number may be a solution worth exploring. Thirdly, due to considerations of fairness in comparison and limited computational resources, we do not employ other PLMs, especially LLMs, as text encoders, which will be considered in our future work.

7 Conclusion

We propose an effective framework: subgraph retrieval enhanced by graph-text alignment, named SEPTA, for commonsense question answering. In our method, we reframe the task as a subgraph vector retrieval problem and introduce a graph-text alignment method to enhance retrieval accuracy and facilitate knowledge fusion for prediction. Subsequently, all the structural information retrieved is then combined by a simple attention mechanism to bolster the reasoning capabilities of PLMs. Extensive experiments on five benchmarks demonstrate the effectiveness of SEPTA.

In the future, our work will focus on the following aspects. First, we will explore more efficacious pre-training tasks for semantic alignment. Second, if there are sufficient computational resources, we intend to apply our approach to larger language models. Third, we will try SEPTA on relevant tasks, e.g., node classification and link predictions on text-attributed graphs.

Acknowledgments. This work is supported by Ant Group through Ant Research Intern Program.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Banerjee, P., Pal, K.K., Mitra, A., Baral, C.: Careful selection of knowledge to solve open book question answering. In: ACL (2019)
2. Bisk, Y., Zellers, R., Bras, R.L., Gao, J., Choi, Y.: PIQA: reasoning about physical commonsense in natural language. In: AAAI (2020)
3. Brannon, W., Fulay, S., Jiang, H., Kang, W., Roy, B., Kabbara, J., Roy, D.: Congrat: Self-supervised contrastive pretraining for joint graph and text embeddings. CoRR **abs/2305.14321** (2023)
4. Brockschmidt, M.: Gnn-film: Graph neural networks with feature-wise linear modulation. In: ICML (2020)
5. Chandra, S., Mishra, P., Yannakoudakis, H., Nimishakavi, M., Saeidi, M., Shutova, E.: Graph-based modeling of online communities for fake news detection. CoRR **abs/2008.06274** (2020)
6. Clark, P., Etzioni, O., Khot, T., Khashabi, D., Mishra, B.D., Richardson, K., Sabharwal, A., Schoenick, C., Tafjord, O., Tandon, N., Bhakthavatsalam, S., Groeneveld, D., Guerquin, M., Schmitz, M.: From 'f' to 'a' on the N.Y. regents science exams: An overview of the aristo project. AI Mag. **41**(4), 39–53 (2020)
7. Davis, E., Marcus, G.: Commonsense reasoning and commonsense knowledge in artificial intelligence. Commun. ACM **58**(9), 92–103 (2015)
8. Dong, J., Zhang, Q., Huang, X., Duan, K., Tan, Q., Jiang, Z.: Hierarchy-aware multi-hop question answering over knowledge graphs. In: WWW (2023)
9. Feng, Y., Chen, X., Lin, B.Y., Wang, P., Yan, J., Ren, X.: Scalable multi-hop relational reasoning for knowledge-aware question answering. In: EMNLP (2020)
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD (2016)
11. Huang, Y., Li, Y., Xu, Y., Zhang, L., Gan, R., Zhang, J., Wang, L.: Mvp-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning. In: ACL (2023)
12. Jiang, J., Zhou, K., Wen, J., Zhao, W.X.: Great truths are always simple: A rather simple knowledge encoder for enhancing the commonsense reasoning capacity of pre-trained models. In: NAACL (2022)
13. Jin, B., Zhang, W., Zhang, Y., Meng, Y., Zhang, X., Zhu, Q., Han, J.: Patton: Language model pretraining on text-rich networks. In: ACL (2023)
14. Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., Hajishirzi, H.: Unifiedqa: Crossing format boundaries with a single QA system. In: EMNLP (2020)
15. Lin, B.Y., Chen, X., Chen, J., Ren, X.: Kagnet: Knowledge-aware graph networks for commonsense reasoning. In: EMNLP (2019)
16. Lin, B.Y., Wu, Z., Yang, Y., Lee, D., Ren, X.: Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In: ACL (2021)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019)
18. Mao, Q., Liu, Z., Liu, C., Li, Z., Sun, J.: Advancing graph representation learning with large language models: A comprehensive survey of techniques. CoRR **abs/2402.05952** (2024)
19. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? A new dataset for open book question answering. In: EMNLP (2018)

20. Rampásek, L., Galkin, M., Dwivedi, V.P., Luu, A.T., Wolf, G., Beaini, D.: Recipe for a general, powerful, scalable graph transformer. In: NeurIPS (2022)
21. Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P.W., Lillicrap, T.: A simple neural network module for relational reasoning. In: NeurIPS (2017)
22. Sap, M., Rashkin, H., Chen, D., Bras, R.L., Choi, Y.: Social iqa: Commonsense reasoning about social interactions. In: EMNLP (2019)
23. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC (2018)
24. Shwartz, V., West, P., Bras, R.L., Bhagavatula, C., Choi, Y.: Unsupervised commonsense question answering with self-talk. In: EMNLP (2020)
25. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: AAAI (2017)
26. Sun, Y., Shi, Q., Qi, L., Zhang, Y.: Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In: NAACL (2022)
27. Talmor, A., Herzig, J., Lourie, N., Berant, J.: Commonsenseqa: A question answering challenge targeting commonsense knowledge. In: NAACL (2019)
28. Taunk, D., Khanna, L., Kandru, S.V.P.K., Varma, V., Sharma, C., Tapaswi, M.: Grapeqa: Graph augmentation and pruning to enhance question-answering. In: WWW (2023)
29. Wang, K., Zhang, Y., Yang, D., Song, L., Qin, T.: GNN is a counter? revisiting GNN for question answering. In: ICLR (2022)
30. Wang, P., Peng, N., Ilievski, F., Szekely, P.A., Ren, X.: Connecting the dots: A knowledgeable path generator for commonsense question answering. In: EMNLP (2020)
31. Wang, X., Kapanipathi, P., Musa, R., Yu, M., Talamadupula, K., Abdelaziz, I., Chang, M., Fokoue, A., Makni, B., Mattei, N., Witbrock, M.: Improving natural language inference using external knowledge in the science questions domain. In: AAAI (2019)
32. Wang, Y., Zhang, H., Liang, J., Li, R.: Dynamic heterogeneous-graph reasoning with language models and knowledge representation learning for commonsense question answering. In: ACL (2023)
33. Yasunaga, M., Bosselut, A., Ren, H., Zhang, X., Manning, C.D., Liang, P., Leskovec, J.: Deep bidirectional language-knowledge graph pretraining. In: NeurIPS (2022)
34. Yasunaga, M., Ren, H., Bosselut, A., Liang, P., Leskovec, J.: QA-GNN: reasoning with language models and knowledge graphs for question answering. In: NAACL (2021)
35. Ye, Q., Cao, B., Chen, N., Xu, W., Zou, Y.: Fits: Fine-grained two-stage training for knowledge-aware question answering. In: AAAI (2023)
36. Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C.D., Leskovec, J.: Greaselm: Graph reasoning enhanced language models. In: ICLR (2022)
37. Zheng, C., Kordjamshidi, P.: Dynamic relevance graph network for knowledge-aware question answering. In: COLING (2022)
38. Zhu, Y., Wang, Y., Shi, H., Tang, S.: Efficient tuning and inference for large language models on textual graphs. CoRR **abs/2401.15569** (2024)
39. Zou, T., Yu, L., Huang, Y., Sun, L., Du, B.: Pretraining language models with text-attributed heterogeneous graphs. In: EMNLP (2023)