JointLK: Joint Reasoning with Language Models and Knowledge Graphs for Commonsense Question Answering

Anonymous ACL submission

Abstract

Existing KG-augmented models for question answering primarily focus on designing elaborate Graph Neural Networks (GNNs) to model knowledge graphs (KGs). However, they ignore (i) the effectively fusing and reasoning over question context representations and the KG representations, and (ii) automatically selecting relevant nodes from the noisy KGs during reasoning. In this paper, we propose a novel model, JointLK, which solves the above limitations through the joint reasoning of LMs and GNNs and the dynamic KGs pruning mechanism. Specifically, JointLK performs joint reasoning between the LMs and the GNNs through a novel dense bidirectional attention module, in which each question token attends on KG nodes and each KG node attends on question tokens, and the two modal representations fuse and update mutually by multi-step interactions. Then, the dynamic pruning module uses the attention weights generated by joint reasoning to prune irrelevant KG nodes recursively. We evaluate JointLK on the CommonsenseQA and OpenBookQA datasets, and demonstrate its improvements to the existing LM and LM+KG models, as well as its capability to perform interpretable reasoning.

1 Introduction

Commonsense question answering (CSQA) requires systems to acquire different types of commonsense knowledge and reasoning skills, which is normal for humans, but challenging for machines (Talmor et al., 2019). Recently, large pre-trained language models (LMs) have achieved remarkable success in many QA tasks and appear to use implicit (factual) knowledge encoded in their model parameters during fine-tuning(Liu et al., 2019; Raffel et al., 2020). Nevertheless, commonsense knowledge is self-evident to humans and is rarely expressed clearly in natural language (Gunning, 2018), which makes it difficult for LMs to learn commonsense knowledge from the pre-training text corpus alone.

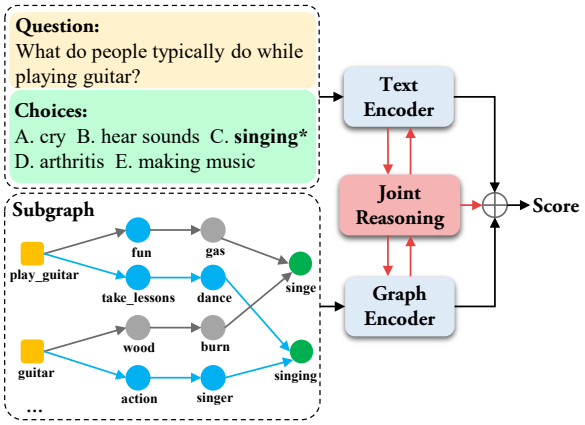


Figure 1: Our knowledge-enhanced joint reasoning model framework with an example from CommonsenseQA. The subgraph is retrieved from ConceptNet.

An extensive research path is to elaborately design graph neural networks (GNNs) (Scarselli et al., 2008) to perform reasoning over explicit structural common sense knowledge from external knowledge bases(Vrandečić and Krötzsch, 2014; Speer et al., 2017). Related methods usually follow a retrieval-and-modeling paradigm. First, the knowledge subgraphs or paths related to a given question are retrieved by string matching or semantic similarity; such retrieved structured information indicates the relation between concepts or implies the process of multi-hop reasoning. Second, the retrieved subgraphs are modeled by a well-designed graph neural network module(Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021) to perform reasoning over knowledge graphs.

However, these approaches have two main issues. First, **the retrieved knowledge subgraph contains many noisy nodes**. Whether through simple string matching or semantic matching, in order to retrieve sufficient relevant knowledge, noise knowledge graph nodes will inevitably be included. Especially

with the increase of hop count, the number of irrelevant nodes will expand dramatically, raising the burden of the model. As the example in Figure 1, some graph nodes such as "*wood*", "*burn*", and "*gas*", although related to some entities in the questions and choice, can mislead the global understanding of the question. Second, **there are limited interactions between language representation and knowledge graph representation**. Specifically, existing LM+KG methods (Lin et al., 2019; Feng et al., 2020) model question context and knowledge subgraphs in isolation by LMs and GNNs, and perform only one interaction in a shallow manner to fuse their representations at the output for prediction. We argue that the limited interaction between the two modalities is the main bottleneck that may prevent the model from understanding the complex question-knowledge relations necessary to answer the question correctly.

Based on the above consideration, we propose JointLK, a model that performs the fine-grained modal fusion and multi-layer joint reasoning between the language model and the knowledge graph. (see Figure 2). Specifically, given a question and retrieved subgraphs, JointLK first obtain the representations of the two modalities by using an LM encoder and a GNN encoder respectively. Then we design a joint reasoning module to generate fine-grained bidirectional attention maps between each question token and each KG node to fuse the information from each modality to the other. Guided by the attention generated in the interaction process, the dynamic pruning module deletes irrelevant nodes to make the model reason along the correct knowledge path. Multiple JointLK layers are stacked to form a hierarchy that supports multi-step interactions and recursive pruning. In summary, our contributions are three-fold:

- We propose JointLK, a novel model that supports multi-step joint reasoning between LM and KG. It uses dense bidirectional attention to simultaneously update query-aware knowledge graph representation and knowledge-aware query representation, bridging the gap between the two information modalities.

- We design a dynamic graph pruning module that recursively removes irrelevant graph nodes at each JointLK layer to ensure that the model reasons correctly with complete and appropriate evidence.

- Experimental results show that JointLK is superior to current LM+KG methods, and the refined evidence is interpretable. Furthermore, through the multi-layer fusion of these two modalities, JointLK exhibits strong performance over previous state-of-the-art LM+KG methods in performing complex reasoning, such as solving questions with negation and complex questions with more entities.

2 Related Work

Commonsense question answering is challenging because the required commonsense knowledge is rarely given in the context of questions and answer choices or encoded in the parameters of pre-trained LMs. Therefore, many works obtain the required knowledge from external sources (e.g., KGs, corpus) to augment CSQA models. Due to the heterogeneity between structured knowledge and unstructured text questions, there are currently two main research methods. Some works(Bian et al., 2021; Lv et al., 2020; Xu et al., 2021) unify the two modalities during model input, such as transforming structured knowledge into plain text through templates or transforming question context into structured graphs. However, the original structural/textual information will inevitably be lost during the conversion process. Other works(Lin et al., 2019; Feng et al., 2020) use LM and GNN to model the two modalities separately, and perform shallow interactions in the latter model stage, such as attentive pooling or simple concatenation of the two modal representations. Although this method can retain the original information of question context and KGs, the limited interaction will affect the flow of information between the two modalities, so we mainly improve on this point.

Recently, QA-GNN(Yasunaga et al., 2021) explicitly views the QA context as an additional node, connects it and KG to form a joint graph, and mutually updates their representations through graph-based message passing. However, it pools the representation of the question context into a single node, which limits the updating of the text representation and fine-grained interaction between the LM and the GNN. Compared with prior works, we retain the individual structure of both modalities, consider fine-grained interaction between any token in question and any entity in KG through dense bidirectional attention, and perform multi-step joint reasoning by stacking several interaction

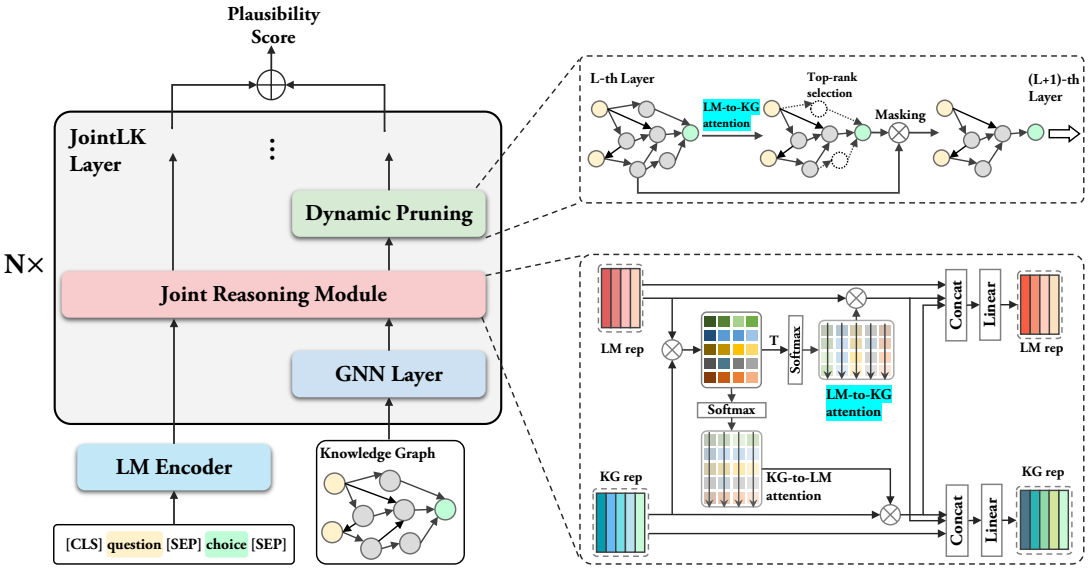


Figure 2: Overall architecture of our proposed JointLK model, which takes a query (question-choice) and a retrieved knowledge subgraph as input, and outputs a scalar that represents the plausibility score of this query. JointLK mainly consists of four modules the Query Encoder, the Graph Layer, the Joint Reasoning Module and the Dynamic Pruning Module, of which the latter three form a stack of N identical layers.

layers. Furthermore, we gradually prune the KG size in each stacked model layer under the guidance of attention weights generated in the interactions, making the reasoning path transparent and interpretable.

3 Methodology

In this section, we introduce the task definition (§ 3.1) and our JointLK model. The model framework is shown in Figure 2. JointLK takes the query and the retrieved knowledge subgraph as input, and outputs a real value as the correctness score of the answer. The model is mainly composed of four parts: query encoder, GNN layer, joint reasoning module and dynamic pruning module, of which the latter three form a stack of N identical layers. We use a pre-trained language model to learn the query representation (§ 3.2), and use the GNN layer to learn the graph representation (§ 3.3). The Joint Reasoning Module receives these two modalities' representations and then apply dense bidirectional attention to make information fusion and representation update for each token and node (§ 3.4). The LM-to-KG attention weight generated in reasoning represents the global importance of each node in the graph, so the dynamic pruning module prunes the graph layer by layer according to this weight and finally retains the most relevant nodes (§ 3.5). After N layers of iteration, the query representation and the trimmed graph representation are used to predict the answer (§ 3.6).

3.1 Task Definition

The CSQA task in this paper is a multiple-choice problem with five answer choices. Given a question q and a set of answer choices $\{a_1, a_2, ..., a_n\}$, our task is to measure the plausibility score between q and each answer choice a then select the answer with the highest plausibility score. In general, questions do not contain any reference to answer choices, so the external knowledge graph provides the necessary background knowledge. We extract from the external KG a subgraph $g = (V, R)$ with the guidance of question and choice. Here V is a subset of entity nodes retrieved from the external KG. $E \in V \times R \times V$ is the set of edges that connect nodes in V, where R is a set of relations types. We describe the detailed extraction process in Appendix A.

3.2 Query Encoder

We follow baselines to use pre-trained language models to encode the query $\{w_i\}_{i=1}^{M}$ (question and choice) into a sequence of vectors $\{q_i^0\}_{i=1}^{M}$:

$$\{\widetilde{q}_1^0, ..., \widetilde{q}_M^0\} = \mathbf{Encoder_{LM}}(\{w_1, ..., w_M\}) \quad (1)$$

Here $\{\widetilde{q}_i^0\}_{i=1}^{M} \in \mathbb{R}^T$ is the last hidden layer vector of each token in the query. Then we feed the representation of tokens into a non-linear layer so that

the text representation space is aligned to the entity representation space:

$$q_i^0 = \sigma(f_s(\widetilde{q}_i^0)) \tag{2}$$

where $f_s : \mathbb{R}^T \rightarrow \mathbb{R}^D$ is a linear transformation, and σ is the activation function. The representations of tokens $\mathbf{Q}^0 = \{q_i^0\}_{i=1}^M \in \mathbb{R}^D$ will be provided to the joint reasoning module for further interaction with the graph entities representations.

3.3 GNN Layer

After we obtain token representations by the query encoder, we further model the subgraph structure to obtain entity representations. First, We use the BERT model with average pooling to get the initial representation for each entity $\mathbf{X^0} = \{x_i^0\}_{i=0}^{|V|} \in \mathbb{R}^D$. Then, we apply GNN Layer to update node representation through iterative message passing between neighbors on the graph, while GNN is built on the RGAT (Wang et al., 2020a) and is a simplification of Yasunaga et al. (2021). For brevity, we formulate the entire computation in one layer as:

$$\{\widetilde{x}_1^l, ..., \widetilde{x}_{|V|}^l\} = \mathbf{GNN-Layer}(\{x_1^{l-1}, ..., x_{|V|}^{l-1}\}) \tag{3}$$

The output representation x_i^l is computed by

$$\hat{\alpha}_{ji} = (x_i^{l-1} W_q)(x_j^{l-1} W_k + r_{ji})^T, \tag{4}$$

$$\alpha_{ji} = \mathbf{softmax}(\hat{\alpha}_{ji}/\sqrt{D}), \tag{5}$$

$$\hat{x}_i^{l-1} = \sum_{j \in N_i \cup \{i\}} \alpha_{ji}(x_j^{l-1} W_v + r_{ji}), \tag{6}$$

$$\widetilde{x}_i^l = \mathbf{LayerNorm}(x_i^{l-1} + \hat{x}_i^{l-1} W_o) \tag{7}$$

where matrices $W_q, W_k, W_v, W_o \in \mathbb{R}^{D \times D}$ are trainable parameters, N_i is the neighbor of node i. $r_{ji} = \psi(e_{ji}, u_j, u_i)$ is the relation feature vector, where e_{ji} is a one-hot vector denoting the relation type of the edge (j, i) and u_j, u_i are one-hot vectors denoting the node types of j and i. The following joint reasoning module will further fuse $\widetilde{x}_i^l$ and q_i^{l-1} to obtain their updated representations.

3.4 Joint Reasoning Module

To reduce the gap of query and knowledge graph features, we fuse them in the joint reasoning module by the dense bidirectional attention mechanism that connects two encoding layers of query and knowledge graph and captures the fine-grained interplay between them.

The module takes the query and KG representations $\mathbf{Q}$ and $\mathbf{X}$ as inputs and then outputs their updated versions. We denote the inputs to the joint reasoning module in the l-st Fusion layer by $\mathbf{Q}^{l-1} = \{q_i^{l-1}\}_{i=1}^M$ and $\widetilde{\mathbf{X}}^l = \{\widetilde{x}_i^l\}_{i=1}^{|V|}$. Given q_i^{l-1} and $\widetilde{x}_i^l$, an affinity matrix is first constructed via:

$$S_{ij}^l = W_S^T [q_i^{l-1}; \widetilde{x}_j^l; q_i^{l-1} \circ \widetilde{x}_j^l] \tag{8}$$

where W_S^T is a learnable weight matrix. We normalize S_{ij}^l in row-wise to derive **KG-to-LM attention** maps on query tokens conditioned by each entity in KG as

$$S_{q_i}^l = \mathbf{softmax}(S_{ij}) \tag{9}$$

and also normalize S_{ij}^l in column-wise to derive **LM-to-KG attention** maps on entities conditioned by each query token as

$$S_{x_j}^l = \mathbf{softmax}(S_{ij}^T) \tag{10}$$

The attended representations are computed as follows:

$$\hat{q}_{ij} = q_i^{l-1} \otimes S_{q_i}^l, \hat{x}_{ij} = \widetilde{x}_j^l \otimes S_{x_j}^l \tag{11}$$

where $\otimes$ represents matrix multiplication. The attended features are fused with the original features of the other modality by concatenation and then compressed to low-dimensional space by:

$$q_i^l = W_Q[q_i^{l-1}; \hat{x}_{ij}; q_i^{l-1} \circ \hat{x}_{ij}; q_i^{l-1} \circ \hat{q}_{ij}], \tag{12}$$

$$\bar{x}_j^l = W_X[\widetilde{x}_j^l; \hat{q}_{ij}; \widetilde{x}_j^l \circ \hat{q}_{ij}; \widetilde{x}_j^l \circ \hat{x}_{ij}] \tag{13}$$

where W_Q, W_X are learnable weights. Then the updated query representation $\mathbf{Q}^l = \{q_i^l\}_{i=1}^M$ will be input to the next l-th stacked JointLK layer of to continue participating in joint reasoning, and the updated KG representation $\bar{\mathbf{X}}^l = \{\bar{x}_i^l\}_{i=1}^{|V|}$ will be input to the next module of the current JointLK layer for pruning.

3.5 Dynamic Pruning Module

In Equation 10, the LM-to-KG attention value implies the importance of different nodes in the subgraph for question answering. Inspired by SAGPool (Lee et al., 2019), under the guidance of query, we retain relevant nodes and cut out irrelevant nodes according to the LM-to-KG attention. Then, We define a hyperparameter, the Retention ratio $K \in (0, 1]$, which determines the number of nodes

to be retained. We choose the top $\lceil K \cdot |V| \rceil$ nodes according to the value of LM-to-KG attention:

$$idx = \mathbf{top-rank}\left(Z, \lceil K \cdot |V| \rceil\right), \quad (14)$$

$$Z_{mask} = Z_{idx} \quad (15)$$

where top-rank is a function that returns the index of top $\lceil K \cdot |V| \rceil$ value, $\cdot_{idx}$ is an indexing operation, and Z_{mask} is corresponding attention mask. Next, the subgraph is formed by pooling out the less essential entity nodes as:

$$\begin{aligned} \mathbf{X}^l &= \bar{\mathbf{X}}^l_{idx,:} \odot Z_{mask}, \\ \mathbf{A}^l &= \bar{\mathbf{A}}^l_{idx,idx} \end{aligned} \quad (16)$$

where $\bar{\mathbf{X}}^l_{idx,:}$ is the row-wise indexed representation matrix of $\bar{\mathbf{X}}^l$, $\odot$ is the broadcasted elementwise product, and $\bar{\mathbf{A}}^l_{idx,idx}$ is the row-wise and col-wise of indexed adjacency matrix. $\mathbf{X}^l = (x^l_1, x^l_2, \ldots, x^l_{\lceil k|V| \rceil})$, $\mathbf{A}^l$ and $\lceil K \cdot |V| \rceil$ are the representation matrix, the adjacency matrix and the number of graph nodes in the next JointLK layer.

3.6 Answer Prediction

After N layers of iteration, we finally obtain the query representation $\mathbf{Q}^N$ that fuses knowledge information and the graph representation $\mathbf{X}^N$ that fuses question information. We compute the score of a being the correct answer as:

$$p = (a|q) = \mathbf{MLP}\left([s; g]\right) \quad (17)$$

where s is the mean pooling of $\mathbf{Q}^N$, and g is the attention-based pooling of $\mathbf{X}^N$. We get the final probability by normalize all question-choice pairs with softmax.

4 Experimental Setup

4.1 Datasets

We evaluate our model on two typical commonsense question answering datasets CommonsenseQA(Talmor et al., 2019) and OpenBookQA(Mihaylov et al., 2018). **CommonsenseQA** is a 5-way multiple-choice QA task that requires commonsense for reasoning and contains 12,102 questions. The test set of CommonsenseQA is not publicly available, and model predictions can only be evaluated every two weeks via the official leaderboard. Hence, we experiment and report the accuracy on the in-house dev (IHdev) and test (IHtest) splits used by Lin et al. (2019), and report the accuracy of our final system on the official test set. **OpenBookQA** is a 4-way multiple choice QA task that requires reasoning with elementary science knowledge, containing 5,957 questions. We use the official data split.

4.2 Implementation Details

We use ConceptNet (Speer et al., 2017), a commonsense knowledge graph, as our structured knowledge source for both of the above tasks. Given each query, we follow the preprocessing steps described in Feng et al. (2020) to retrieve the subgraph from ConceptNet, and the max hop size is 3 (see Appendix A for the detail). We use ROBERTA-LARGE (Liu et al., 2019) as the backbone. We use cross-entropy loss and RAdam (Liu et al., 2019a) optimizer. In training, we set the maximum input sequence length to text encoders to 100, batch size to 128, and perform early stopping. We set the dimension (D = 200) and number of layers (N = 5) of our GNN module, with dropout rate 0.2 applied to each layer (Srivastava et al., 2014). We use separate learning rates for the LM encoder and the graph encoder. We choose the LM encoder learning rate from$\{1 \times 10^{-5},\ 2 \times 10^{-5},\ 3 \times 10^{-5}\}$, and choose the graph encoder learning rate from$\{1 \times 10^{-3},\ 2 \times 10^{-3}\}$. Each model is trained using one GPU (Tesla_v100-sxm2-16gb), which takes 20 hours on average.

4.3 Compared Method

Although text corpus can provide complementary knowledge except for knowledge graphs, our model focuses on improving the use of KG and the joint reasoning between LM and KG, so we choose LM and LM+KG as the comparison methods.

To investigate the role of KG, we compare with the benchmark model ROBERTA-LARGE, which does not use KG. For LM+KG methods, they share a similar high-level framework with our methods, that is, LM is used as a text encoder, GNN or RN is used as an external KG encoder, but the way of using knowledge or reasoning is different:(1) Relationship network (RN) (Santoro et al., 2017), (2) RGCN (Schlichtkrull et al., 2018), (3) GconAttn (Wang et al., 2019), (4)KagNet (Lin et al., 2019) and (5)MHGRN (Feng et al., 2020), (6) QA-GNN (Yasunaga et al., 2021). (1), (2) and (3) are the relational perception GNNs for KGs, and (4), (5) and (6) are further model paths in KGs. To be fair, we use the same LM for all comparison methods.

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-large(w/o KG)	73.07 (±0.45)	68.69 (±0.56)
+ RGCN	72.69 (±0.19)	68.41 (±0.66)
+ GconAttn	71.61 (±0.39)	68.59 (±0.96)
+ KagNet	73.47 (±0.22)	69.01 (±0.76)
+ RN	74.57 (±0.91)	69.08 (±0.21)
+ MHGRN	74.45 (±0.10)	71.11 (±0.81)
+ QA-GNN	76.54 (±0.21)	73.41 (±0.92)
+ JointLK (Ours)	**77.88** (±0.25)	**74.43** (±0.83)

Table 1: Performance comparison on CommonsenseQA in-house split. We follow the data division method of Lin et al. (2019) and report the in-house Dev (IHdev) and Test (IHtest) accuracy(mean and standard deviation of four runs).

Methods	Test
RoBERTa (Liu et al., 2019)	72.1
Albert (Lan et al., 2019) (ensemble)	76.5
RoBERTa + FreeLB (Zhu et al., 2019) (ensemble)	73.1
RoBERTa + HyKAS (Ma et al., 2019)	73.2
RoBERTa + KE (ensemble)	73.3
RoBERTa + KEDGN (ensemble)	74.4
XLNet + GraphReason (Lv et al., 2020)	75.3
RoBERTa + MHGRN (Feng et al., 2020)	75.4
Albert + PG (Wang et al., 2020b)	75.6
RoBERTa + QA-GNN (Yasunaga et al., 2021)	76.1
Roberta + JointLK (Ours)	**76.6**

Table 2: Performance comparison on the CommonsenseQA official leaderboard. Our model has achieved state-of-the-art under the setting of ROBERTA-LARGE encoder.

Methods	RoBERTa-Large	AristoRoBERTa
Fine-tuned LMs (w/o KG)	64.80 (±2.37)	78.40 (±1.64)
+ RGCN	62.45 (±1.57)	74.60 (±2.53)
+ GconAttn	64.75 (±1.48)	71.80 (±1.21)
+ RN	65.20 (±1.18)	75.35 (±1.39)
+ MHGRN	66.85 (±1.19)	80.6
+ QA-GNN	67.80 (±2.75)	82.77 (±1.56)
+ JointLK (Ours)	**70.34** (±0.75)	**84.92** (±1.07)

Table 3: Test accuracy on OpenBookQA. Methods with AristoRoBERTa use the textual evidence by Clark et al. (2019) as an additional input to the QA context.

Methods	Test
Careful Selection (Banerjee et al., 2019)	72.0
AristoRoBERTa	77.8
KF + SIR (Banerjee and Baral, 2020)	80.0
AristoRoBERTa + PG (Wang et al., 2020c)	80.2
AristoRoBERTa + MHGRN (Feng et al., 2020)	80.6
ALBERT + KB	81.0
AristoRoBERTa + QA-GNN (Yasunaga et al., 2021)	82.8
T5* (Raffel et al., 2020)	83.2
UnifiedQA(11B)* (Khashabi et al., 2020)	87.2
AristoRoBERTa + JointLK (Ours)	**85.6**

Table 4: Test accuracy on OpenBookQA leaderboard. All listed methods use the provided science facts as an additional input to the language context. The previous top 2 systems, UnifiedQA (11B params) and T5 (3B params) are 30x and 8x larger than our model.

5 Results and Analysis

5.1 Main Results

The results on CommonsenseQA in-house split dataset and official test dataset are shown in Table 1 and Table 2. The results on OpenBookQA test dataset and leaderboard are shown in Table 3 and Table 4. We can observe that JointLK performs best among all fine-tuned LMs and existing LM+KG models. On CommonsenseQA, our model's test performance improves by 5.74% over fine-tuned LMs and 1.02% over the prior best LM+KG model, QA-GNN. On OpenbookQA, our model's test performance improves by 6.52% over fine-tuned AristoRoBERTa, and 2.15% over QA-GNN. Additionally, we also submit our best model to the leaderboards, and our JointLK (with the text encoder being ROBERTA-LARGE) ranks first among comparable approaches. Compared with the previous best model MHGRN and QA-GNN, the boost over them suggests the effectiveness of our proposed joint reasoning between LM and KG and the dynamic pruning mechanism.

In particular, we do not compare with the higher ranking models on the leaderboard, such as unified QA(Khashabi et al., 2020), Albert + DESC-KCR (Xu et al., 2021), because they either use a stronger text encoder or use additional data resources, while our model focuses on improving the joint reasoning between LM and KG.

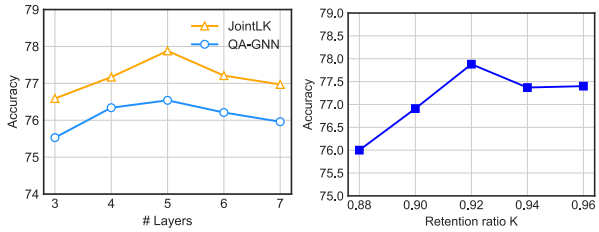
5.2 Ablation Studies

We further conduct in-depth analyses to investigate the effectiveness of different components in our model. We show the accuracy of JointLK on the CommonsenseQA IHdev set.

Impact of JointLK components We assess the impact of the joint reasoning module (§ 3.4) and the dynamic pruning module (§ 3.5), shown in Table 5. Disabling the dynamic pruning module results in 0.5% drop in performance, showing that some nodes in subgraph are not conducive to reasoning. Especially, when we disable the joint reasoning module, the corresponding dynamic pruning module will also be removed, because the latter depends

Methods	IHdev-Acc. (%)
JointLK (N=5)	**77.88**
- Dynamic Pruning Module	77.38
- Joint Reasoning Module	76.61

Table 5: Ablation study on model components using ROBERTA-LARGE as the text encoder. We report the IHdev accuracy on CommonsenseQA.



(a) (b)

Figure 3: Ablation study on stacked of JointLK layers (a) and the retention ratio in pruning (b).

on the attention value in the former. Then the results have a significant drop: $77.88\% \rightarrow 76.61\%$, suggesting that the joint reasoning between LM and KG is critical.

Impact of stacked of JointLK Layers We investigate the impact of the number of JointLK layers (shown in Figure 3 (a)). The increase of layers continues to bring benefits until layers $N = 5$. However, performance begins to drop when $N > 5$. As the number of layers increases, the model changes from underfitting to overfitting.

Impact of the Retention Ratio in Pruning The retention ratio K is a hyperparameter of the dynamic pruning module. Since it is recursively pruning in each stacked layer of JointLK, the percentage of graph nodes that the model ultimately retains is also related to the number of layers of JointLK, that is, K^N, where $N = 5$. Experiments show that if the retention ratio is too high, there may be almost no pruning effect (for example, K=0.98, 90% of the nodes are retained in the last layer); otherwise, useful nodes may be deleted. As shown in Figure 3 (b), when the number of JointLK layers $N = 5$, $K = 0.92$ (about 66% of the original nodes remain in the last layer) works the best on the CommonsenseQA dev set.

5.3 Quantitative Analysis

Considering the overall performance improvement of our model on these two datasets, we analyze whether the improvement is reflected in questions that require more complex reasoning, such as questions with negation and complex questions with more entities. We compare our model with the prior best LM+KG model, QA-GNN in Table 6.

Methods	IHdev-Acc (Overall)	IHdev-Acc (Questions w/ negation)	IHdev-Acc (Questions w/ ≤7 entities)	IHdev-Acc (Questions w/ >7 entities)
Number	*1221*	*133*	*723*	*498*
QA-GNN	76.99	72.18	76.63	77.51
JointLK(**Ours**)	78.38	**75.18** (↑**3.00**)	**77.59** (↑**0.96**)	**79.52** (↑**2.01**)

Table 6: Performance on the CommonsenseQA IHdev set on questions with negation terms and fewer/more entities.

Questions with negation Large LMs do well due to memorizing subject and filler co-occurrences but are easily distracted by elements like negation(Zagoury et al., 2021). To investigate the reasoning ability of the model on negation, we retrieved 133 questions with negation terms (e.g., no, not, nothing, never, unlikely, don't, doesn't, didn't, can't, couldn't) from the CommonsenseQA IHdev set. JointLK exhibits a big boost (↑3.00%) over QA-GNN, suggesting its strength in negation reasoning. The fine-grained joint inference of LM and GNN allows the model to pay attention to the semantic nuances of language expressions.

Questions with fewer/more entities When the question contains many entities, the size and noise of the retrieved KG may limit the model's performance because the model needs to understand the complex relationship between entities. According to statistics (see Appendix A), questions contain an average of 7 entities, so we divide the question into two categories: containing fewer entities (≤7) and more entities(>7). Compared with QA-GNN, JointLK has a bigger boost on questions with more entities (↑2.01%) than those with fewer entities (↑0.96%), suggesting that our model can reduce the reasoning difficulty of complex questions because it can remove irrelevant nodes in reasoning.

5.4 Interpretability: A Case Study

We aim to interpret JointLK's reasoning process by analyzing the pruning of the knowledge subgraph. Figure 4 shows an example from CommonsenseQA where our model correctly answers the question and finally retains reasonable reasoning paths by pruning the subgraph. The flow from (a) to (b) to (c) represents the recursive pruning of the subgraph according to the LM-to-KG attention weight at each GNN update layer. From (a) to (b), although the nodes *wood* and *burn* bridge the reason-

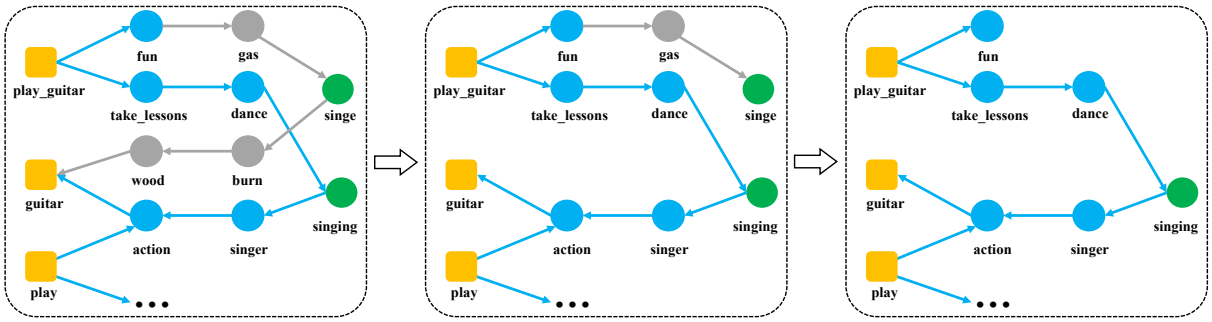


Figure 4: Case study of our model reasoning and pruning process. The question and answer choices corresponding to this case are: "What do people typically do while playing guitar? A.cry B. hear sounds C. **singing** D. arthritis E. making music".

ing gap between question entity and answer entity, their semantics are very different from the question. From (b) to (c), "$play_guitar \xrightarrow{usedfor} fun$" and "$fun \xrightarrow{relatedto} gas \xrightarrow{relatedto} sing$" are both reasonable, but the former is related to the semantics of the question, and the latter is not. Two paths are reserved in (c), "$play_guitar \xrightarrow{hassubevent} take_lessons \xrightarrow{hassubevent} dance \xrightarrow{relatedto} sing$" and "$play_guitar \xrightarrow{relatedto} action \xrightarrow{relatedto} singer \xrightarrow{relatedto} sing$". These two paths describe two possible scenarios that support answering the question.

5.5 Error Analysis

In order to understand why our model fails in some cases, we randomly select 100 error cases and group them into several categories. There are three main types of errors, and we show some examples in the Appendix C.

Miss important evidence (39/100) Although we can retrieve many nodes related to questions and choices from ConceptNet, due to the incompleteness of the knowledge graph, there may be missing essential evidence nodes in the reasoning paths to answer the question. For example, although *"eating_dinner"* will cause *"sleepiness"* or *"indigestion"*, knowledge such as "lactose intolerance causes indigestion" is essential to answer the question (Wikipedia: *Lactose intolerance is a common condition caused by a decreased ability to digest lactose, a sugar found in dairy products.*). However, ConceptNet does not cover such knowledge or not is retrieved.

Indistinguishable knowledge (25/100) Several choices of the question may be correct, difficult to distinguish, and which one is correct may vary from person to person. For example, *"human"* and *"cat"* may be at location *"bed"* or *"comfortable chair"*, and the knowledge provided by ConceptNet is also the same. The model may choose bed because the bed appears more frequently in the pre-trained corpus.

Incomprehensible questions (23/100) This type of error often occurs when the question is particularly long, involving various events and changes in the characters' emotions. The model is difficult to understand the scene described by the question. Some questions may require reasoning based on events, but the knowledge in ConceptNet is more based on entities and attributes.

The above three types of errors show that selecting complete, accurate, and context-sensitive knowledge is vital for more effective KG-enhanced models.

6 Conclusion

In this work, we propose JointLK and provide a set of experiments to prove that (i) LM and KG interactive fusion can reduce the semantic gap between the two information modalities and make better use of KG for joint reasoning with LM. (ii) Dynamic pruning module can recursively delete irrelevant subgraph nodes at each layer of JointLK to provide fine appropriate evidence. Our results on CommonsenseQA and OpenBookQA demonstrate the superiority of JointLK over other methods using external knowledge and the strong performance in performing complex reasoning. In addition, our research results can be broadly extended to other tasks that require KGs as additional background knowledge to augment LMs, such as entity linking, KG completion and the recommendation system.

Ethical Impact

This paper proposes a general approach to fuse language models and external knowledge graphs for commonsense reasoning. We worked within the purview of acceptable privacy practices and strictly followed the data usage policy. In all the experiments, we use public datasets and consist of their intended use. We neither introduce any social/ethical bias to the model nor amplify any bias in the data, so we do not foresee any direct social consequences or ethical issues.

References

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

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12574–12582.

Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2019. From'f'to'a'on the ny regents science exams: An overview of the aristo project. *arXiv preprint arXiv:1909.01958*.

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

David Gunning. 2018. Machine common sense concept paper. *arXiv preprint arXiv:1810.07528*.

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

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. In *International Conference on Learning Representations*.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*, pages 3734–3743. PMLR.

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)*, pages 2829–2839.

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

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

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. *EMNLP 2019*, page 22.

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*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

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. *Journal of Machine Learning Research*, 21:1–67.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *Advances in Neural Information Processing Systems*, 30.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, pages 4149–4158.

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

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020a. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020b. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4129–4140.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020c. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

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

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207, Online. Association for Computational Linguistics.

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*, pages 535–546.

Avishai Zagoury, Einat Minkov, Idan Szpektor, and William W Cohen. 2021. What's the best place for an ai conference, vancouver or _: Why completing comparative questions is difficult. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14292–14300.

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 Extracting subgraph from External KG

We choose ConceptNet as the external knowledge base, and we follow the process of Feng et al. (2020) and Yasunaga et al. (2021) to retrieve the knowledge subgraph.

Given the question and choice, we use *spaCy* to identify the entities that appear in ConceptNet in question and choice, respectively, and get the initial node set V_q and V_a, which form the initial node set $V_{q,a}$. For example, in the question "*What do people typically do while playing guitar?*" and choice "*singing*", V_q = {guitar, people, play, play_guitar, playing, playing_guitar, typically}, V_a = {singe, singing}. Then, in order to extract the subgraph related to question and choice, we add the bridge entities on the 1 and 2 hop paths between any pair of entities in $V_{q,a}$, thus obtaining the retrieved entity set V.

There may be many nodes in V, especially long questions contain many concepts. We follow the preprocessing method of Yasunaga et al. (2021), connect the nodes with question + choice, and calculate the relevant scores of the nodes through a pre-trained LM. We only retain the top 200 scoring nodes (It is worth noting that this is the preprocessing of the retrieval process, which is different from the dynamic pruning in section 3.5. The former is to score only one node and separate from the whole subgraph where the node is located, while the latter is recursive pruning in the updating process of the modeling subgraph).

Finally, we get the relation set R by merging the relation types in ConceptNet and adding reverse relation. We retrieve all the edges in R of any two nodes in V. In addition, we add question as a node q to V, and add the bidirectional edges of q to V_q and q to V_a. The relation types are shown in Table 7, and the statistics of the retrieved nodes are shown in Table 8.

Relation	Merged Relation
AtLocation LocatedNear	AtLocation
Causes CausesDesire *MotivatedByGoal	Causes
Antonym DistinctFrom	Antonym
HasSubevent HasFirstSubevent HasLastSubevent HasPrerequisite Entails MannerOf	HasSubevent
IsA InstanceOf DefinedAs	IsA
PartOf *HasA	PartOf
RelatedTo SimilarTo Synonym	RelatedTo
CapableOf	CapableOf
CreatedBy	CreatedBy
Desires	Desires
UsedFor	UsedFor
HasContext	HasContext
HasProperty	HasProperty
MadeOf	MadeOf
NotCapableOf	NotCapableOf
NotDesires	NotDesires
ReceivesAction	ReceivesAction
$q \rightarrow V_q$	$q \rightarrow V_q$
$q \rightarrow V_a$	$q \rightarrow V_a$

Table 7: Relation types after preprocessing. *RelationX indicates the reverse relation of RelationX. There are 19 kinds of merged relations. We consider the reverse edge of each relation during training and testing, so there are 38 relation types in total.

B Node Initialization

For each entity in the subgraph, we need to obtain its feature representation. Following (Feng et al., 2020), we first use the template to convert the knowledge triples in ConceptNet into sentences, and feed them into the pre-trained BERT-LARGE, obtaining a sequence of tokens embeddings from the last layer. For each entity, we perform mean pooling over the tokens of the entity's occurrences across all the sentences to form the initial embeddings x_i^0.

C Error Types and Examples

In Table 9, we present examples for each error type in the Commonsense IHdev set. Because the average number of knowledge subgraph nodes corresponding to each case is about 100, we cannot list them all, so only some important nodes are shown here.

Datesets	Split	Average $\|V_q\|$	Average $\|V_a\|$	Average $\|V\|$
CommomsenseQA	Train set	7.43	2.07	107.96
	Dev set	7.20	2.05	106.55
	Test set	7.38	2.05	106.22
OpenBookQA	Train set	6.59	2.85	100.14
	Dev set	6.48	3.41	108.15
	Test set	6.42	3.08	101.60

Table 8: Statistics on the number of retrieved subgraph nodes corresponding to each piece of data. V_q is the set of entities included in a question. V_a is the set of entities included in a choice. V contains V_q, V_a, and any bridging entity with no more than two hops between any pair of entities in V_q and V_a.

Error type	**Example**	
Missing important evidence (39/100)	Question	He has lactose intolerant, but was **eating dinner** made of cheese, what followed for him?
	Answer choices	digestive×| feel better ×| **sleepiness**×| **indigestion** ✓| illness×
	Subgraph for correct answer	$eating_dinner \xrightarrow{causes} indigestion$, $intolerant \xrightarrow{relatedto} pain \xrightarrow{isa} symptom \xleftarrow{isa} indigestion$, ...
	Subgraph for predicted answer	$lactose \xrightarrow{relatedto} food \xrightarrow{hassubevent} eating_dinner \xrightarrow{causes} sleepiness$, $intolerant \xrightarrow{relatedto} bear \xrightarrow{relatedto} sleep$, ...
Indistinguishable knowledge (25/100)	Question	Where would a cat snuggle up with their human?
	Answer choices	floor×| humane society×| **bed**×| **comfortable chair**✓| window sill×
	Subgraph for correct answer	$cat \xrightarrow{atlocation} chair$, $human \xrightarrow{atlocation} chair$, ...
	Subgraph for predicted answer	$cat \xrightarrow{atlocation} bed$, $human \xrightarrow{atlocation} bed$, ...
Incomprehensible questions (23/100)	Question	The man tried to **break** the glass in order to make his escape in time, but he could not. The person in the car, trying to kill him, did what?
	Answer choices	**accelerate**✓| putting together×| **working**×| construct×| train×
	Subgraph for correct answer	$escape \xleftarrow{isa} break \xrightarrow{antonym} accelerate$, $kill \xrightarrow{relatedto} attack \xrightarrow{relatedto} accelerate$, $man \xrightarrow{relatedto} break \xrightarrow{relatedto} falling \xrightarrow{hassubevent} accelerate$, ...
	Subgraph for predicted answer	$break \xrightarrow{isa} action \xrightarrow{relatedto} work$, $escape \xleftarrow{isa} break \xleftarrow{hassubevent} work$, $kill \xrightarrow{causes} die \xleftarrow{hassubevent} work$, ...

Table 9: Several error cases of JointLK model on CommonsenseQA dev dataset. Because there are many nodes in the subgraph, we represent some nodes and relationships in the subgraph in the form of links.