

Article

Retrieval-augmented Knowledge Graph Reasoning for Commonsense Question Answering

Yuchen Sha ¹, Yujian Feng¹, Miao He², Shangdong Liu², and Yimu Ji^{2,*}

¹ School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China

² School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

* The corresponding authors are Yimu Ji (jiym@njupt.edu.cn)

Abstract: Existing Knowledge Graph (KG) models for commonsense question answering present two challenges: (i) existing methods retrieved entities related to questions from the knowledge graph, which may extract noise and irrelevant nodes, and (ii) lack of interaction representation between questions and graph entities. However, current methods mainly focus on retrieving relevant entities with some noisy and irrelevant nodes. In this paper, we propose a novel Retrieval-augmented Knowledge Graph (RAKG) model, which solves the above issues through two key innovations. First, we leverage the density matrix to make the model reason along the corrected knowledge path and extract an enhanced knowledge graph subgraph. Second, we fuse representations of questions and graph entities through a bidirectional attention strategy, in which two representations fuse and update by Graph Convolutional Network (GCN). To evaluate the performance of our method, we conduct experiments on two widely-used benchmark datasets CommonsenseQA and OpenBookQA. The case study gives insight into findings that the augmented subgraph provides reasoning along the corrected knowledge path for question answering.

Keywords: Commonsense question answering; Knowledge Graph; Graph Convolutional Network.

1. Introduction

Question Answering (QA) has become one of the most popular downstream tasks in Natural Language Processing (NLP) in recent years. QA tasks utilize Large-scale pre-trained language models (LMs) to obtain token representations, exemplified by BERT [2], GPT [3], ELMo [4], RoBERTa [5], which have achieved remarkable success. Meanwhile, commonsense as external knowledge is essential for QA systems to predict the correct answer, which is natural knowledge for humans [1]. External knowledge is often incorporated in the form of ConceptNet [7] and Freebase [8], where nodes represent entities and edges represent the relationships between two entities [6,10]. Adding a knowledge graph can indeed enhance the interpretability and credibility of prediction answers. For example, Figure 1 shows an example of the commonsense question answering based on KG. In this example, given the question "Which is a good source of nutrients for a mushroom?", correct answer entity "a cut peony", and some choice nodes. Using the KG to answer a question about "nutrients for a mushroom", we need to look for entities that involve the following concepts: "nutrients" and "mushroom". Moreover, we use entity linking to identify these entities and match them with the concepts in the Knowledge Graph (KG).

However, most existing models used the KG to obtain information increases the risk of retrieving irrelevant or noise nodes, and have difficulties in interaction representation between questions and graph entities. Therefore, extracting an enhanced Knowledge Graph subgraph has been shown to be more effective in obtaining an accurate reasoning path for question answering [9]. On one hand, when extracting relevant a subgraph, some noisy nodes are often contained by simple semantic matching [11,12]. As the reasoning paths become more complex, the noisy nodes also change continuously. When noise nodes are not discarded in a timely manner, the performance of the model in predicting answers will

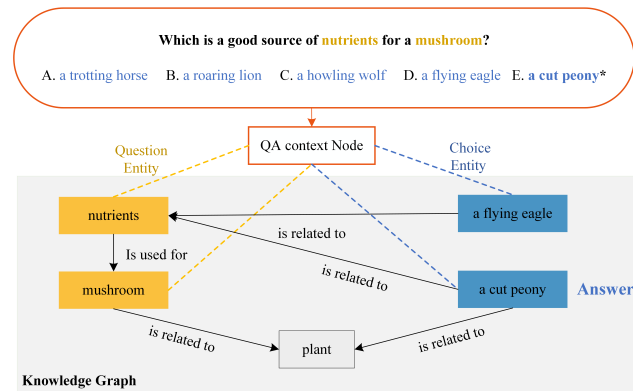


Figure 1. Given the example of QA tasks (question and five candidate choices). Our approach involves analyzing the question entity and incorporating KGs to predict the answer.

be negatively impacted [31,41]. On the other hand, in previous works [13,14], language models and knowledge graph models existed as independent components, which resulted in missing relationship between the question and graph entities. The limited interaction between language models and knowledge graph models is the major issue, which causes models to struggle with understanding complex question-knowledge relations [26,33,43].

To address the above two issues, we propose a novel Retrieval-augmented Knowledge Graph (RAKG) architecture, which make the model reason along the corrected knowledge graph for question answering. First, RAKG model extract the most relevant subgraphs by utilizing the density matrix, which affects the weights of neighborhood nodes during training. Second, our proposed model utilizing bidirectional attention strategy to fuse the representations of questions and knowledge graph entities.

Specifically, RAKG model has two major steps: (i) we concatenate the given QA pair obtaining the representations. And we extract the KG in the form of ConceptNet obtaining the graph embeddings, which removes irrelevant nodes to ensure the appropriate reasoning path. We then compute the inner product of node representations and build direct neighborhoods based on the density matrix. This can be seen as a way to capture the importance of question-entities pairs in the subgraph. (ii) given the question and retrieved subgraphs, RAKG obtains initialized representations of both the question and graph entities via Graph Convolutional Network and language models, respectively. In addition, to make the model fuse the representations of question and graph entities, we further incorporate a bidirectional attention strategy between language model and knowledge graph model to bridge the gap between the question and graph entities representations.

In summary, the contributions of this work are as follows:

1) We propose a novel RAKG model with a retrieval augmented KG subgraph for Question Answering. The augmented subgraph is extracted using the density matrix, which removes irrelevant nodes at each layer of the RAKG.

2) Our model utilizes a bidirectional attention strategy to effectively integrate the representations of both language models and knowledge graphs. Moreover, we use R-dropout to prevent overfitting and improve model generalization.

3) Experimental results show that proposed RAKG achieve better performance than several baselines on CommonsenseQA and OpenbookQA benchmarks.

2. Related Work

2.1. Knowledge Graph-based Question Answering

Knowledge Graph-based Question Answering (KG-QA) is a Natural Language Processing task that predicts the answer by retrieving relevant nodes from the knowledge graph. The task has been studied in recent years [17,18]. In order to utilize external knowledge, KGs are widely used in QA tasks [15,16]. Since then, researchers have made

significant works in developing KG-QA systems. The method has shown significant improvement over previous methods in terms of both accuracy and interpretability.

However, given the KGs are usually large and complex that contain many irrelevant nodes to the question [19]. Moreover, with the large LMs, the relationships complexity of reasoning over them will increase. To address this issues, retrieving KG subgraph based on a variety of methods has been proposed [20,21]. Some works [40,41] leverage KG subgraph for commonsense question answering that obtain multi-hop relationships for reasoning paths. Furthermore, the additional node contributes to construct relationships between language model and knowledge graph model, while [31] add the question node to the KG subgraph structure. Therefore, the graph representations can learn more information from question nodes and graph entities. However, the above methods does not integrate well the semantic information between questions and knowledge graph representations, so we mainly focus on this point.

2.2. Graph Convolutional Network

Graph Convolutional Network (GCN) [22] is a type of neural network that operate on graphs and have become increasingly popular in recent years. The use of GCN allows for the modeling of complex relationships and dependencies between nodes in a graph [23]. For QA tasks, GCN have been utilized to KG complex relationships between entities and provide more accurate answers, which achieved by using GCN to encode text and construct a graph based on the entities mentioned in the KG [24]. Moreover, R-GCN is a extension of GCN that deals with a large number of relationships with multiple graphs [33].

Recent research on GCN have focused on expanding their capabilities to handle different types of graph data, such as directed GCN [25] and dynamic GCN [26]. For example, the dynamic GCN is capable of handling graphs with varying structures and sizes, as well as temporal data and time-varying graphs. Moreover, the dynamic GCN can adapt to changing graphs by learning the graph structure from the KG, enabling it to identify the most relevant nodes and edges for the QA task. Meanwhile, there has been other works on combining GCN with other methods, such as Graph Attention Network (GAT) [27] and Graph Adversarial Training [28], which use the self-attention and multi-head attention strategies to learn the node representations. In contrast to previous research, our approach utilizes a static R-GCN model, but leverages a density matrix and a bidirectional attention strategy to weight the neighborhood nodes in the KG.

3. Methodology

The Retrieval-Augmented KG model is composed of four main parts, as shown in Figure 2. These are the Language Context Encoder, Knowledge Graph Subgraph Extraction, RAKG module, Answer Prediction.

3.1. Task Formulation

Given a question q and a set of candidate answers $\{a_1, a_2, \dots, a_n\}$, the goal of the RAKG model is to identify the most plausibility answer among a set of N candidate answers a_n . Our proposed method provides the question q and an external knowledge graph (KG) as inputs to the model. A KG is denoted as $KG = (V, E)$, where V represents entities in the knowledge graph, E represents edges between entity1 and entity2.

3.2. Language Context Encoder

We have chosen RoBERTa [29] as the backbone model for our model. Our proposed end-to-end question answering system utilizes RoBERTa to perform token representations on concatenated question-answer pairs, allowing for a streamlined one-pass approach.

Given an input question q of length $|L|$, we concatenate the question q and the answer a_n together in the format of $[[CLS]; q; [SEP]; a_n]$ to form Language Context L . Where $[CLS]$ and $[SEP]$ are special tokens utilized by large-scale pre-trained language models. The

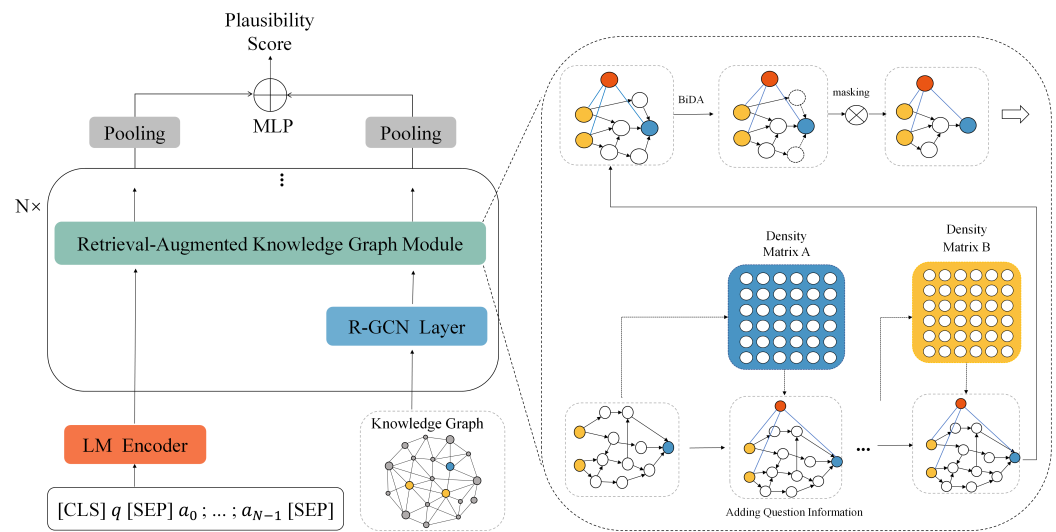


Figure 2. Our proposed RAKG model, which composed of the Language Context Encoder module, KG Subgraph Extraction module, Retrieval-Augmented KG module and Answer Prediction module. [CLS] token is placed at the beginning of the first sentence, while [SEP] token is used to separate the two input sentences. The red circle represent the question node. The yellow circle represent entity nodes mentioned in the question. The blue circle represent the answer node.

input L is provided to the encoder f_{enc} of the pre-trained LMs to generate a list of token 126
 representations. The representations of tokens Q will be sent to RAKG module in order to 127
 be further integrated with the graph entity representations, are calculated as follows: 128

$$Q = \sigma(f_s(f_{enc}(L))) \quad (1)$$

where σ is the activation function, f_s represents a linear transformation, and f_{enc} encode L 129
 into vectors. 130

3.3. KG Subgraph Extraction 131

We develop an end-to-end entity linking system, which takes a question-answer pair 132
 as input and links entities in the question to the knowledge graph. For the KG, we utilize 133
 ConceptNet [30], a general-domain knowledge graph that has multiple semantic relational 134
 edges. 135

To construct the subgraph for each example, we follow a previous approach [31] by 136
 selecting entities with an exact match between n-gram tokens and ConceptNet corpus using 137
 some normalization rules. We expand the subgraph by adding another set of entities based 138
 on the two-hop [29] reasoning path in the KG from the current entities in the subgraph. 139
 Additionally, we include the question as a separate node in the subgraph KG_{sub} and connect 140
 it to entities in the question that are relevant to the answer. 141

3.4. Retrieval-Augmented Knowledge Graph module 142

After obtaining token representations through the LM encoder, we further use the 143
 model to obtain entity representations in the subgraph. First, we utilize the RoBERTa model 144
 to obtain an initialization representation for each entity $X^0 = \{x_n^0\}_{n=0}^{|V|}$, and then we use a 145
 R-GCN [33] network to update the entity node representation through iterative message 146
 passing update, are calculated as follows: 147

$$\{\tilde{x}_1^l, \dots, \tilde{x}_{|V|}^l\} = R-GCN(\{\tilde{x}_1^{l-1}, \dots, \tilde{x}_{|V|}^{l-1}\}) \quad (2)$$

where $|V|$ represents the graph entity nodes sizes and subtracted 1 due to the question 148
 node. 149

To provide a more explicit representation of the reasoning path, we utilize the density matrix to re-evaluate the significance of neighboring nodes. Specifically, we propose RAKG model, where we use neighbor density matrix to score the relevance of neighbor nodes. For each node, we use inner product compute density matrix M_{mat} , and the density matrix change while the representations are computed at each graph layer. In our proposed RAKG model, the forward-pass message passing updates of the graph entity nodes and question nodes.

Finally, we use to $M_{mat}^{(l+1)}$ multiple the node representation to learn the weight of edges based on density matrix between the nodes during the message passing, updating node representations are computed as follows:

$$h^{(l+1)} = \sigma \left(M_{mat}^{(l+1)} \cdot h'^{(l+1)} \cdot W_g \right) \quad (3)$$

where σ represents activation function, W_g represents a hyperparameter. Besides, we update question node representation is calculated in a similar way as above.

In addition, we employ a bidirectional attention strategy to facilitate interaction between the pre-trained language model and the knowledge graph, computing a pair-wise matching matrix M_{pair} . The computation of the attention strategy from the language model to the knowledge graph is shown as

$$S_{q_i}^l = \text{softmax}(M_{pair}) \quad (4)$$

and the computation of the attention strategy knowledge graph to language model is shown as

$$S_{x_j}^l = \text{softmax}(M_{pair}^T) \quad (5)$$

We obtain vector representations for both entities and questions, and fuse them using a bi-directional attention strategy with a concatenated matrix. The attended features are then compressed into a low-dimensional space. This approach helps to further clarify the reasoning path and minimizes the number of irrelevant entity nodes for the given question. Specifically, we use bidirectional attention strategy to choose the top K relevance nodes on the basis of above. Specifically, we define a retention ratio $\lceil K \cdot |V| \rceil K \in (0, 1]$, which represents relevance $\lceil K \cdot |V| \rceil$ nodes to be retained. We choose the top-rank nodes according to the value of bidirectional attention strategy.

3.5. Answer Prediction

In this system, the query representation and the graph representation are combined to compute a score for a given answer. The query representation is obtained after N layers of iteration, where the query information and knowledge graph information are fused together. Similarly, the graph representation is obtained from the KG subgraph representation pooled from the last graph layer. The score of a candidate answer is computed as the dot product of the mean pooled query representation and the pooled KG subgraph representation:

$$p = (a|L, KG_{sub}) = MLP([s; g]) \quad (6)$$

where s is the mean pooling of query representation, including token representations h_L , question representations h_Q , and g is the KG subgraph representation pooled from the last graph layer. The scoring function is used to determine the similarity between the query and the candidate answer, based on their representations in the knowledge graph. We get the final probability by normalize all question-choice pairs with softmax and use R-Dropout[34] to regularize the end-to-end model. R-Dropout decreases the randomness introduced by original dropout by minimizing the KL-divergence loss between the output distributions of the two sub-models sampled by the dropout.

4. Experimental Setup 191

4.1. Datasets 192

We evaluated our model on two different benchmarks for commonsense reasoning, CommonsenseQA [35] and OpenbookQA [36], which are widely used benchmarks. We used the same data split for both benchmarks and applied ConceptNet as an external knowledge graph for both benchmarks. 193
194
195
196

CommonsenseQA [35] is a benchmark for evaluating the ability of computer systems to understand and reason about commonsense knowledge. With over 32,000 questions and answers covering everyday scenarios and events, it requires systems to possess commonsense reasoning capabilities to accurately answer the questions. The dataset is unique in its focus on testing the ability of systems to reason through multiple choice questions. 197
198
199
200
201

OpenbookQA [36] is based on open text sources, with questions sourced from foundational science textbooks covering subjects such as physics, biology, and chemistry. The dataset contains over 5,000 questions and is unique in that it focuses on testing the ability of systems to answer questions that require reasoning beyond simple fact retrieval. Each question is accompanied by a multiple-choice set of answers, making it a useful benchmark for evaluating natural language understanding and reasoning models. 202
203
204
205
206
207

Table 1. Statics of CommonsenseQA (CSQA) and OpenBookQA (OBQA)

Dataset	Train	Dev	Test	Choices
CSQA	9,741	1,221	1,140	5
OBQA	4,957	500	500	4

The goal of the task for these two datasets is to select the most appropriate answer for a given question. During our training process, the data cleaning was performed to ensure that each question has at least one correct answer. Table 1 gives the experiment data scale and split of the two datasets. Two commonly used rank-based metrics, IHdev accuracy and IHtest accuracy, are used to evaluate the performance of the model on the same task and dataset. 208
209
210
211
212
213

4.2. Implementation details 214

RAKG model is implemented by Pytorch. We utilized the pre-trained RoBERTa [29] model to encode our information. Our end-to-end architecture was trained using R-Dropout [34] and the RAdam optimizer [37]. The batch size was set to 16 and the maximum text input sequence length was set to 128. We employed a 5-layer graph neural module in our experiments, and its impact is described in section 5.2. The learning rate for the language model was set to 1e-5 and the learning rate for the graph module was set to 1e-3. We train our model for 100 epochs and the best model obtained in the dev set is used to evaluate in the test set. Each model was trained on a single GPU (3090Ti). 215
216
217
218
219
220
221
222

4.3. Baselines 223

To assess the effectiveness of the proposed approach, we conducted experiments using three baseline methods that rely on multiple-choice question answering tasks. A comprehensive comparison with a wide range of models is made. They include: 224
225
226

Roberta-large [38] is a common baseline for multi-choice question answering tasks. It does not rely on any extra knowledge, such as ConceptNet. 227
228

RN [39] is designed to capture the core common properties for relation reasoning. 229

KagNET [40] is a path-based model that encodes multi-hop relationships extracted from knowledge graphs using an LSTM sequence model. 230
231

MHGRN [41] is a powerful baseline. The MHGRN model proposes a scalable method for knowledge-aware question answering that uses multi-hop relational reasoning. The model uses graph neural networks to perform multi-hop reasoning on a knowledge graph and integrates language models to handle NLP tasks. 232
233
234
235

QA-GNN [31] is a novel approach, which integrates language models and knowledge graphs for multi-hop question answering. The method leverages the strengths of both models, using the language model to generate initial representations of the query and the knowledge graph to refine the representations through graph neural networks.

5. Results and Analysis

5.1. Main Results

In this section, we compare our model with the baseline models and methods, as well as the ranking results of CommonsenseQA and OpenbookQA. We also provide an analysis of the components and features of the model. Table 2 and Table 3 illustrate the main comparison results with baselines, respectively.

Table 2 shows the performance of different models on the CommonsenseQA benchmark. KagNet and RN are strong baselines. We can observe that RAKG outperforms other methods. Specifically, our method outperforms the RoBERTa-large by 3.7% and RN by 2.2% on CommonsenseQA benchmark. The boost over RN suggests that RAKG makes a better extraction of KG subgraph to perform language model and knowledge graph model. We use the same RoBERTa-large model as the backbone for training our model to ensure a fair comparison with other models. This result shows the effectiveness of our RAKG model architecture. Additionally, Table 3 shows the performance on the OpenbookQA. Our method performance improves by 8% over R-GCN model and 4.1% over QA-GNN.

Table 2. Dev accuracy and Test accuracy of various models on the CommonsenseQA benchmark.

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-large[38]	73.07	68.69
+R-GCN[33]	72.69	68.41
+GconAttn[43]	72.61	68.59
+KagNet[40]	73.47	69.01
+RN[39]	74.57	69.08
+MHGRN[41]	74.45	71.11
+QA-GNN[31]	76.50	73.40
+RAKG(Ours)	76.74	73.51

Table 3. Dev accuracy and Test accuracy of various models on the OpenbookQA benchmark.

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-large[38]	66.7	64.8
+R-GCN[33]	65.0	62.4
+GconAttn[43]	64.5	61.9
+RN[39]	66.8	65.2
+MHGRN[41]	68.1	66.8
+QA-GNN[31]	68.9	67.8
+DRGN[26]	70.1	69.6
+RAKG(Ours)	73.0	77.0

5.2. Ablation Studies

Impacts of RAKG components We conducted an ablation study on the CommonsenseQA development benchmark to evaluate the effectiveness of different components of RAKG, as presented in Table 4. First, we remove KG subgraph, our model without the KG subgraph obtains 69.6% on the CommonsenseQA. This result indicates that while the language model can provide answers to the questions, the accuracy is relatively low. Our

model, which incorporates the KG subgraph and density matrix, achieved an accuracy of 75.4%. The result demonstrate the usefulness of the density matrix in helping the model identify more relevant neighbor nodes, improve the learning of node representations, and ultimately achieve better performance. Moreover, by incorporating a bi-directional attention strategy module, our method accuracy improved to 76.2%. We further enhanced the model's generalization capability by incorporating R-Dropout, resulting in an accuracy rate of 76.7%. These results show the effectiveness of our approach and suggest that it has the potential for broader application.

Table 4. Ablation study on model components using RoBERTa-large as the text encoder. We report the IHdev Accuracy on CommonsenseQA.

Methods	IHdev-Acc. (%)
RAKG (N=5) w/o KG subgraph	69.6
+KG subgraph	72.6
+density matrix	75.4
+bidirectional attention	76.2
+R-Dropout	76.7

Impact of number of RAKG Layers In our RAKG model, each layer of the graph is composed of numerous nodes that represent various features or patterns of the input data. However, not all nodes are equally important for predicting the output. The number of graph layers is impact of RAKG architecture because the density matrix is based on GCN. As shown in Figure 3, we investigate the impact of the number of RAKG layers. The results indicate that as the number of graph layers increases, the accuracy rate also increases, peaking at fifth layers. However, after the fifth layer, the accuracy gradually decreases due to model overfitting. Therefore, our model sets the number of graph layers to 5.

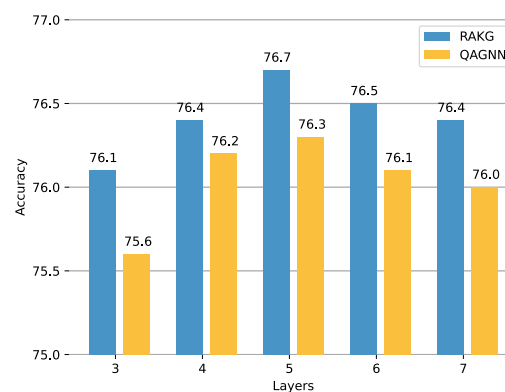
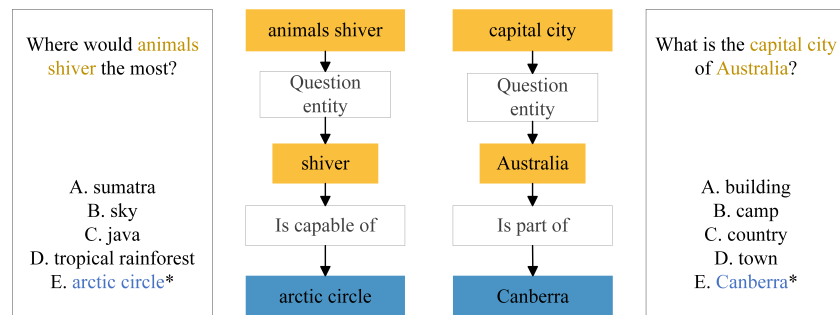


Figure 3. Impact of number of RAKG Layers. We show IHdev Accuracy of RAKG on CommonsenseQA.

Impact of KL-divergence loss weight in R-Dropout We evaluate the influence of KL-divergence loss weight. As shown in Table 5, we conducted an experiment where we set the weights of kl weight to take the values of 0.5, 0.7, 0.9, 1, and 5. Through experimentation, we found that as the weights get larger, the peak value of 76.7% is reached at the weight value of 0.7, but then the accuracy starts to decrease. By adjusting the KL-divergence loss weight, we can effectively control the complexity of our model and prevent model overfitting. It is worth noting that the optimal weight value for kl-divergence loss may vary depending on the specific task, dataset size, and model architecture. Therefore, it is important to choose the KL-divergence loss weight in R-Dropout carefully to avoid model overfitting and achieve better performance. Therefore, we use $\alpha=0.7$ to ensure the best performance of our model.

Table 5. Impact of the KL-divergence loss weight in R-Dropout.

α	RAKG+R-Dropout (%)
$\alpha=0.5$	74.5
$\alpha=0.7$	76.7
$\alpha=0.9$	76.3
$\alpha=1.0$	75.6
$\alpha=5.0$	75.3

**Figure 4.** Case study on model interpretability. We present two examples from CommonsenseQA with the reasoning paths in RAKG.

Model Interpretability In order to better understand how our model arrives at its answers, we can analyze the reasoning path it takes. Fig. 4 provides two examples from the CommonsenseQA, where our model provides the correct reasoning path and answers the question accurately. Each example consists of a question and 5 candidate answers. Our method links the entities in the question to those in the answer options, and uses these entities to find the most likely reasoning path between the question and the correct answer. Our model is able to capture the potential relationships between the candidate answers and the question. By analyzing the relationships between entities and making connections across them, our model can effectively reason and make accurate predictions. This is particularly important in question answering tasks where the answer may not be immediately obvious and requires the ability to reason and draw connections between different pieces of information. In summary, our model's ability to identify and analyze reasoning paths between entities is a critical component of its success in answering questions. By leveraging this capability, our model can accurately reason and arrive at the correct answers even when faced with complex and ambiguous questions.

6. Conclusion

In this paper, we propose a novel Retrieval-augmented Knowledge Graph (RAKG) model, which retrieve entities from the external source of knowledge in the form of a Knowledge Graph. Our key innovations include (i) Density matrix, where we compute the relevance of KG neighborhood relationships to remove irrelevant nodes at each layer of RAKG. (ii) The bidirectional attention strategy, where we integrate the representations of both questions and knowledge graphs to obtain semantic information. Moreover, R-dropout can prevent our model overfitting and improve model generalization. Through both quantitative and qualitative analyses, our results on CommonsenseQA and OpenBookQA demonstrate the superiority of RAKG over baseline methods using the KG, as well as its strong performance in performing complex reasoning paths.

Author Contributions: Ideas, writing-review, editing, Y.S. and Y.F.; derivation and simulation experiments, writing-original draft, Y.S. and M.H.; funding acquisition, Y.J and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62176264), Natural Science Foundation of Jiangsu Province (Higher Education Institutions) (20KJA520001), Open Research Project of Zhejiang Lab (No. 2021KF0AB05).

Data Availability Statement: The data supporting the findings of this study will be available in the github.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

KG	Knowledge Graph
QA	Question Answering
GCN	Graph Convolutional Network
KG-QA	Knowledge Graph-based Question Answering

References

- Seo J, Oh D, Eo S, et al. PU-GEN: Enhancing generative commonsense reasoning for language models with human-centered knowledge[J]. *Knowledge-Based Systems*, 2022, 256: 109861.
- Li J, Niu L, Zhang L. From Representation to Reasoning: Towards both Evidence and Commonsense Reasoning for Video Question-Answering[C]. *IEEE Conference on Computer Vision and Pattern Recognition*. 2022: 21273-21282.
- Liu J, Hallinan S, Lu X, et al. Rainier: Reinforced knowledge introspector for commonsense question answering[J]. *arXiv:2210.03078*, 2022.
- Peters M.E, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep Contextualized Word Representations[C]. *North American Chapter of the Association for Computational Linguistics*.2018: 2227-2237.
- Zhan X, Huang Y, Dong X, et al. PathReasoner: Explainable reasoning paths for commonsense question answering[J]. *Knowledge-Based Systems*, 2022, 235: 107612.
- Bhargava P, Ng V. Commonsense knowledge reasoning and generation with pre-trained language models: a survey[C]. *AAAI Conference on Artificial Intelligence*. 2022, 36(11): 12317-12325.
- Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *AAAI conference on artificial intelligence*. 2017, 31(1):1-8.
- Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]. *ACM SIGMOD international conference on Management of data*. 2008: 1247-1250.
- Zhang Q, Chen S, Fang M, et al. Joint reasoning with knowledge subgraphs for Multiple Choice Question Answering[J]. *Information Processing & Management*, 2023, 60(3): 103297.
- Qiao Z, Ye W, Zhang T, et al. Exploiting Hybrid Semantics of Relation Paths for Multi-hop Question Answering Over Knowledge Graphs[J]. *arXiv:2209.00870*, 2022.
- Zhang J, Zhang X, Yu J, et al. Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering[C]. *Association for Computational Linguistics*. 2022: 5773-5784.
- Cui L, Zhu W, Tao S, et al. Mining non-lattice subgraphs for detecting missing hierarchical relations and concepts in SNOMED CT[J]. *Journal of the American Medical Informatics Association*. 2017, 24(4): 788-798.
- Zhang Y, Jin L, Li X, et al. Edge-Aware Graph Neural Network for Multi-Hop Path Reasoning over Knowledge Base[J]. *Computational Intelligence and Neuroscience*. 2022 (2022): 4734179-4734179.
- Ren H, Dai H, Dai B, et al. Smore: Knowledge graph completion and multi-hop reasoning in massive knowledge graphs[C]. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022: 1472-1482.
- Abacha A B, Zweigenbaum P. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies[J]. *Information processing & management*, 2015, 51(5): 570-594.

16. Li Z, Zhong Q, Yang J, et al. DeepKG: an end-to-end deep learning-based workflow for biomedical knowledge graph extraction, optimization and applications[J]. *Bioinformatics*, 2022, 38(5): 1477-1479. 369-371
17. Zhang, Q, Chen, S, Xu, D, Cao, Q, Chen, X, Cohn, T, & Fang, M. (2022). A survey for efficient open domain question answering. *ArXiv:2211.07886*. 372-373
18. Zhang Y, Dai H, Kozareva Z, et al. Variational reasoning for question answering with knowledge graph[C]. *AAAI conference on artificial intelligence*. 2018, 32(1). 374-375
19. Tiddi I, Schlobach S. Knowledge graphs as tools for explainable machine learning: A survey[J]. *Artificial Intelligence*, 2022, 302: 103627. 376-377
20. Chen M, Zhang W, Zhu Y, et al. Meta-knowledge transfer for inductive knowledge graph embedding[C]. *ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022: 927-937. 378-380
21. Howard P, Ma A, Lal V, et al. Cross-Domain Aspect Extraction using Transformers Augmented with Knowledge Graphs[C]. *ACM International Conference on Information & Knowledge Management*. 2022: 780-790. 381-383
22. Gao H, Wang Z, Ji S. Large-Scale Learnable Graph Convolutional Networks[J]. *arXiv:1808.03965*, 2018. 384-385
23. Parisot S, Ktena S I, Ferrante E, et al. Spectral graph convolutions for population-based disease prediction[C]. *Medical Image Computing and Computer Assisted Intervention*. 2017: 177-185. 386-387
24. Therasa M, Mathivanan G. ARNN-QA: Adaptive Recurrent Neural Network with feature optimization for incremental learning-based Question Answering system[J]. *Applied Soft Computing*, 2022, 124: 109029. 388-389
25. Ma L, Zhang P, Luo D, et al. Syntax-based graph matching for knowledge base question answering[C]. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2022: 8227-8231. 390-393
26. Zheng C, Kordjamshidi P. Dynamic Relevance Graph Network for Knowledge-Aware Question Answering[C]. *International Conference on Computational Linguistics*. 2022: 1357-1366. 394-395
27. Shan Y, Che C, Wei X, et al. Bi-graph attention network for aspect category sentiment classification[J]. *Knowledge-Based Systems*, 2022, 258: 109972. 396-397
28. Li J, Peng J, Chen L, et al. Spectral Adversarial Training for Robust Graph Neural Network[J]. *IEEE Transactions on Knowledge Data Engineering*, 2022 (01): 1-14. 398-399
29. Feng Y, Chen X, Lin B Y, et al. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering[C]. *Empirical Methods in Natural Language Processing*. 2020: 1295-1309. 400-401
30. Speer R, Chin J, Havasi C. Conceptnet 5.5: An open multilingual graph of general knowledge[C]. *Association for the Advancement of Artificial Intelligence conference on artificial intelligence*. 2017, 31(1). 402-404
31. Yasunaga M, Ren H, Bosselut A, et al. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering[C]. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021: 535-546. 405-407
32. Feng Y, Chen X, Lin B Y, et al. Scalable multi-hop relational reasoning for knowledge-aware question answering[J]. *arXiv preprint arXiv:2005.00646*, 2020. 408-409
33. Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. 2018: 593-607. 410-411
34. Wu L, Li J, Wang Y, et al. R-drop: Regularized dropout for neural networks[J]. *Advances in Neural Information Processing Systems*, 2021, 34: 10890-10905. 412-413
35. Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2018: 4149-4158. 414-417
36. Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *Empirical Methods in Natural Language Processing*. 2018: 2381-2391. 418-420
37. Liu L, Jiang H, He P, et al. On the variance of the adaptive learning rate and beyond[J]. *arXiv:1908.03265*, 2019. 421-422
38. Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. *arXiv:1907.11692*, 2019. 423-424
39. Santoro A, Raposo D, Barrett D G, et al. A simple neural network module for relational reasoning[J]. *Advances in neural information processing systems*, 2017, 30. 425-426

-
40. Lin B Y, Chen X, Chen J, et al. Kagnet: Knowledge-aware graph networks for commonsense reasoning[J]. arXiv:1909.02151, 2019. 427
428
 41. Feng Y, Chen X, Lin B Y, et al. Scalable multi-hop relational reasoning for knowledge-aware question answering[J]. arXiv:2005.00646, 2020. 429
430
 42. Zhang Z, Xu Z Q J. Implicit regularization of dropout[J]. arXiv:2207.05952, 2022. 431
 43. Wang X, Kapanipathi P, Musa R, et al. Improving natural language inference using external knowledge in the science questions domain[C]. Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. 2019, 33(01): 7208-7215. 432
433
434

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content. 435
436
437
438