LMGFuse: Language Models and Graph reasoning Fuse deeply for question answering

Aoxing Wang[1], *Pengfei Duan*[1,2], *Yongbing Li*[1], *Wenyan Hu*[1], *Shengwu Xiong*[1,2]
[1]School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China;
[2]Sanya Science and Education Innovation Park, Wuhan University of Technology, Sanya 572000, China;
Email: waxzerobug@qq.com, {duanpf, 271551, 305277, Xiongsw}@whut.edu.cn

***Abstract*—The combination of pre-trained language models (LM) and knowledge graphs (KG) can enhance the reasoning ability for Question Answering. However, previous methods typically fuse the two modalities in a shallow or knowledge-draining manner, not taking full advantage of the knowledge representation of both. How to effectively fuse the different knowledge representations is still a problem of current research. In our work, a novel model is proposed that fuses LM modal knowledge representations and graph neural network (GNN) modal knowledge representations deeply over multiple layers of modality interaction operations. Specifically, the model includes an information interaction unit, through which KG and LM knowledge can be transferred between modalities to realize knowledge fusion directly, reducing information loss. In addition, we add the context node of implicit knowledge from LM encoding in the construction of the reasoning subgraph in advance for enhancing the reasoning of the GNN. We evaluate our model on two domains in the biomedical benchmark (MedQA-USMLE) and commonsense benchmarks (OpenBookQA and CommonsenseQA). Experimental results show that our model achieves a particular improvement over existing LM and LM+KG models for reasoning over both situational constraints and structured knowledge.**

Keywords – Question Answering; GNN; LM; knowledge fusion.

I. Introduction

Question Answering is a challenging task for complex questions because it often contains multiple subjects, relations, and implicit background knowledge, and currently, the hot ChatGPT model is also conducting relevant research. Generally, knowledge can be encoded implicitly in a large unstructured pre-trained language model (LM) [1], or explicitly represented in structured knowledge graphs (KGs), such as ConceptNet [2]. Recently, the fine-tuning of large pre-trained language models generated by training on large text corpora on QA datasets has made great progress and has become the dominant paradigm for question-answering tasks [3], [4]. However, previous models are flawed in structured reasoning because they rely only on simple patterns (at times spurious) to reason about answers, rather than strong, structured reasoning that fuses explicit encyclopedic knowledge with implicit knowledge [5].

DOI reference number: 10.18293/SEKE2023-023

In other words, existing pre-trained language models for fine-tuning is the lack ability to exploit unambiguous encyclopedias and commonsense knowledge for reasoning [6].

Previous studies have shown that KGs are suitable for structured reasoning and play an essential role in structured reasoning (e.g., providing background knowledge) [7]–[9]. Therefore, extensive research exists to carefully design graph neural networks (GNN) to obtain answers by retrieving knowledge subgraphs, paths related to a given question by string matching, or semantic similarity or inference after modeling the retrieved subgraphs [10], [11]. However, using these inferences to retrieve graphs can introduce noise and limit the ability of models to effectively utilize both knowledge representations for reasoning [12].

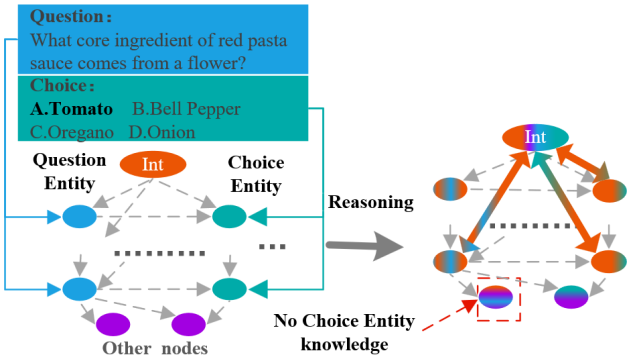

Fig. 1: The left graph is based on context entities, while the right graph shows the result after inference. "Int" is a token used for knowledge exchange. Different node types are represented by colors that alternate during knowledge fusion. However, GNN may lose information during propagation, resulting in the inability to obtain distant information, as seen in the red box lacking Choice Entity knowledge.

To leverage the background knowledge provided by KGs to enhance LM representations, previous approaches combine these two models' representations (i.e., expressive large language models and structured KGs) to improve inference performance [10], [13]. However, these methods typically fuse the two modalities in a shallow and non-interactive manner,

encoding both separately and combining them at the output for a prediction, or using one to augment the input of the other [11], [14], which limits the ability to exchange useful information between the two models. Latest research [14] attempted to fuse these two models' representations by tokens, but it assumed that the GNN knowledge could be learned and aggregated to fixed token nodes, and ignored the problem of information loss during iterative message passing between neighbors on the graph (see Figure.1, the nodes enclosed by the red frame may lose the knowledge of choice entity nodes after reasoning). How to effectively fuse the representations from KG and LM remains an important open question.

To address the above issues, we propose a *Language Models and Graph reasoning Fuse deeply for question answering model* (LMGFuse) as shown in Figure 2, a new model which can deeply integrate and exchange the two model representations in multi-layer architecture for multimodal fusion. Our proposed LMGFuse is mainly composed of two parts: one is the pre-trained language model to encode and understand the QA context; the other is the modal interaction of attention-based GNN and LM for joint reasoning. The former is to generate implicit background knowledge of the context, and the latter is to fuse implicit knowledge and explicit knowledge for reasoning. Referring to previous studies [11], we use the LM to encode the QA context to generate implicit knowledge, and combine this knowledge with the QA context entities to retrieve a KG subgraph following prior works [13]. After that, the LM knowledge representation and subgraph are fed into the model fusion layer, which will fuse the token node information output by the LM encoder with all node information in GNN. Through these layers, each node of the subgraph can learn the knowledge from LM directly, and the LM encoder can also learn the subgraph knowledge, reducing knowledge loss. Meanwhile, to reduce the number of parameters in modal fusion, dimension reduction of parameters is carried out by factoring, improving also the efficiency of the operation.

The contributions of this paper are three-fold: (1) an innovative approach to achieve knowledge representation fusion between LM and GNN, (2) innovative use of reasoning subgraph construction and parameter reduction techniques in question answering, (3) the experimental results on two domains with three datasets (CommonsenseQA [15], OpenBookQA [16] and MedQA-USMLE [17]) are better than the existing LM and LM+KG fusion models.

II. Related Work

There are two main research methods for QA systems under complex problems in prior work: semantic parsing-based (SP-based) and information retrieval-based (IR-based) [18], [19]. SP-based reasoning methods based on the fusion of LM and GNN knowledge representations have made great progress and become a hot research topic. Some works use two-tower to fuse the representations, but they lack contralateral information interaction or have information loss [20]. Others attempt to use one pattern to enhance the other serially, such as using the last layer of the LM knowledge representation to enhance the GNN structured representation or using the GNN structured representation to heighten the context representation [10], [13], [21]. However, in previous works, the interaction mode between the two models was limited, because the information between them could not be interacted and fused, but only flowed in one direction [14].

Several studies aim to fuse information from two models at a deeper level. Some of these works [22] use LM implicit knowledge combined with GNN model structured reasoning to construct QA data for inferencing. However, these methods focus too much on the reasoning of implicit knowledge. Recently, GREASELM [14] and QA-GNN [11] proposed updating the LM representations and GNN representations jointly through message passing. Nevertheless, their method of jointly updating knowledge representations does not handle it well: QA-GNN uses single-pool representations without deep fusion, and GREASELM updates the representation with information loss. In our work, we keep the token node representation of the LM and the graph node representations for deep fusion and use this token node to exchange information with each node in the graph reducing information loss.

In addition, some studies have explored ways to enhance the representation of LM with explicit knowledge from KG in the pre-training stage. However, this modal interaction is limited to the knowledge representation provided [23] and does not fully utilize the structured reasoning ability of the two modes.

III. Approach: LMGRFQA

As shown in Figure 2, the input to the model is the QA context [q: a] concatenated between question q and candidate answer a. LMGRFQA works as follows. First, we use LM encode representations of QA context as context nodes and retrieve KG based on QA context entities to build subgraphs containing implicit knowledge. Then, before modal fusion, we use an N-layer LM encoder to the QA context for easier knowledge modal transfer through token nodes later in the first modal interaction layer. In modal knowledge fusion, we maintain the independent structures of LM and GNN and use the designed *Exchange GNN and LM's Int token representation unit* (EXGLInt) to cross-fusion after each layer to update the knowledge representation obtained by each model. After multi-layer interaction, each node representation can learn the knowledge representation of two modalities. Finally, we make a final prediction using the LM token node representation and GNN node representations through the pooling and MLP layers.

A. Subgraph Construction

In the process of subgraph retrieval and construction, given a [q: a], we retrieve a KG subgraph following prior works [13], [14] and follow the settings of [11] to divide subgraph nodes into four categories: question nodes, option nodes, context nodes, and other knowledge nodes, respectively (corresponding to the node color, green, purple, pink and blue in the subgraph of Figure 2) to capture the strength of association between two nodes. We set up implicit knowledge of the QA

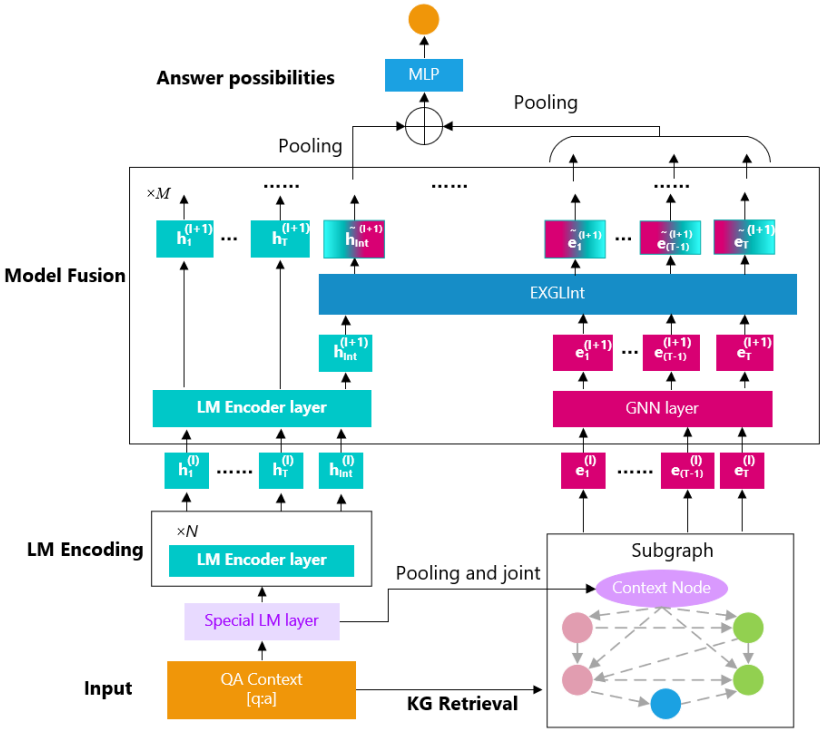

Fig. 2: Overview of LMGFuse Architecture. Given the QA context, we use LM to encode entity nodes and build subgraphs for inference. Then, LM is used to update implicit knowledge representation, and GNN is used to update knowledge representation and use a modal interaction to conduct knowledge fusion after each layer of LM and GNN.

context as the subgraph one node by pooling and joint in advance for enhancing the reasoning of GNN, where we use a special LM encoder for enhancing the representation such as Roberta-Large [24] for CommonsenseQA, AristoRoBERTa [25] for the OpenbookQA and so on. The node connects the QA context node to each question entity and answers entity nodes.

B. Language Pre-Representation

For the sequence of QA context embeddings $\{w_{int}, w_1, ..., w_T\}$, after fed through a special LM encoding layer, we use an N-layer LM encoder to encode it into a language representation $\{h^n_{int}, h^n_1, ..., h^n_T\}$, incorporating location information, etc. We opt to use the BERT [26] layer as the N-layer LM encoder due to its relatively smaller parameter size compared to other LM encoders.

$$\{h^i_{int}, h^i_1, ..., h^i_T\} = LM(\{h^{i-1}_{int}, h^{i-1}_1, ..., h^{i-1}_T\}) \quad (1)$$
$$for\ i = 1, ..., N$$

where $LM(\cdot)$ is a single-layer pre-trained language model with parameter initialization loaded in advance, h^i_{int} represents the encoded knowledge representation of the token node at layer i, which interacts with the GNN node representation to exchange information. More technical implementation details need to refer to [11].

C. Graph Inference Representation

We take the embedding $\{e^0_1, ..., e^0_{T-1}, e^0_T\}$ of the constructed subgraph node as input which is initialized from the LM and construct the subgraph by referring to the construction method of the Graph Attention Framework (GAT) [27]. The node representation after l layer is calculated by the following formula.

$$\{e^l_1, ..., e^l_{T-1}, e^l_T\} = GNN(\{e^{l-1}_1, ..., e^{l-1}_{T-1}, e^{l-1}_T\}) \quad (2)$$
$$for\ l = 1, ..., M$$

where $GNN(\cdot)$ represents GAT, and its technical design scheme follows from [11], [14]. GNN calculates and updates the knowledge representation of each node $e_j \in \{e_1, ..., e_J\}$ through its neighbor nodes knowledge representation, node's type, and edge information.

$$e^l_j = f_n(\sum_{i \in N_j \cup \{e_j\}}^{\infty} \alpha_{ij} m_{ij}) + e^{l-1}_j \quad (3)$$

where N_j represents the set of e_j neighbor nodes, α_{ij} denotes the attention weight of message passing, m_{ij} represents the message from neighbor node i to node j, and f_n represents the multi-layer MLP. Information about the neighbor node of node j such as the relation type and node representation are aggregated to m_{ij}, and it is calculated by the following formula.

$$r_{ij} = f_r(\tilde{r}_{ij}, u_i, u_j) \quad m_{ij} = f_m(e_i, u_i, r_{ij}) \quad (4)$$

where r_{ij} is relation embedding from node i to node j, u_i and u_j are the node type embedding of nodes i and j, $\tilde{r}_{ij}$ is a relation embedding for the relation connecting e_i and e_j, f_r is a multi-layer MLP, and f_m is a linear transformation. α_{ij} reflects the importance of the neighbor node i to the message of node j, and its calculation formula is as follows.

$$q_i = f_q(e_i, u_i) \quad k_j = f_k(e_j, u_j, r_{sj}) \tag{5}$$

$$\gamma_{ij} = \frac{q_i^T k_j}{\sqrt{D}} \quad \alpha_{ij} = \frac{exp(\gamma_{ij})}{\sum_{e_s \in N_j \cup e_j} exp(\gamma_{ij})} \tag{6}$$

where u_i, u_j, r_{sj} are defined the same as above, D is graph nodes encode dimensions, and f_q, f_k are linear transformation. More technical implementation details need to refer to the GAT [27].

D. Modal Knowledge Interaction

After two independent knowledge representation layers of LM and GNN, we use an EXGLInt for modal interaction, which combines the token representation of LM with the representation of each GNN node.

$$[\tilde{h}_{int}^l; \tilde{e}_i^l] = EXGLInt([h_{int}^l; e_i^l]) \quad for\ i = 1, ..., T \tag{7}$$

where T represents the number of subgraph nodes and h_{int}^l, e_i^l are defined the same as above. In EXGLInt, we use multiple layers of MLP as information exchange units and use a two-layer pooling layer to degrade an excessive number of parameters in combination with factorization ideas [28]. LM knowledge representation does not participate in the interaction of GNN representation except with the token node directly, and the token node conducts knowledge interaction with each node representation of GNN. Through modal interaction, each node of GNN can learn LM modal knowledge, and LM can also learn GNN modal knowledge from multi-layer interaction (see Figure.2, the color fusion of node).

E. Reasoning and Prediction

After modal fusion of LM and GNN, the obtained h_{int}^M knowledge representation and graph node knowledge representation $\left\{e_1^M, ..., e_{T-1}^M, e_T^M\right\}$ are concatenated through the pooling layer. Then the representation is fed into MLP and softmax to score a given (question, answer choice) pair based.

$$p = (a|q) = MLP(h_{int}^M; e_1^M, ..., e_{T-1}^M, e_T^M) \tag{8}$$

We use the cross-entropy loss and RAdam optimizer to optimize the whole model end-to-end.

IV. Experiments Settings

Following previous work [11], [14], we set the batch size to 128 and use mini-batch training. We set separate learning rates for GNN and LM, where the learning rate for GNN is chosen from $\left\{1 \times 10^{-3}, 2 \times 10^{-3}\right\}$ and the learning rate for LM is chosen from $\left\{1 \times 10^{-5}, 3 \times 10^{-5}\right\}$ [12]. The pre-trained language model uses parameters provided by the Pytorch interface in advance for parameter initialization. Given each query, we set the number of subgraph retrieval hops to 2 according to [13] and the number of nodes reserved for each subgraph to 200 following previous work [11], where we set the node dimension of the graph to 200 and number of layers (N = 5) of our GNN module [14]. We use one GPU (GeForce RTX 3090 Ti-24g) for our experiments, and each task takes about 10 hours.

A. DataSets

We evaluate our model LMGFuse on three standard QA benchmarks: CommonsenseQA [15], OpenBookQA [16], and MedQA-USMLE [17], which come from different domains (commonsense and medical).

CommonsenseQA is a 5-option commonsense question answering dataset of 12,102 questions that requires common sense knowledge for reasoning. We conduct experiments on the in-house data split of [10] to compare to baseline methods since the CommonsenseQA test set is not publicly available.

OpenbookQA is a 4-option question commonsense answering dataset of 5957 questions that require scientific facts knowledge for reasoning. We use the official data-splitting method [16].

MedQA-USMLE is a 4-option medical question answering dataset of 12723 questions that require biomedical and clinical knowledge for reasoning. The data segmentation method refers to the official paper [17].

B. Knowledge Graph

We use *ConceptNet* [2] as the knowledge source for CommonsenseQA and OpenBookQA which is a general-domain knowledge graph and better suited to commonsense reasoning tasks. It has 799,273 nodes and 2,487,810 edges in total. For MedQA-USMLE, We use the knowledge graph constructed by [11], [14]. It contains 9,958 nodes and 44,561 edges.

C. Language Models

We set up different language models for different domain tasks to better reason the implicit knowledge of each domain task. We use the Roberta-Large [24] in CommonsenseQA, and AristoRoBERTa [25] in OpenbookQA, which are commonsense pre-trained language models. We use the SapBERT [29] on MedQA-USMLE, which is a biomedical pre-trained language model. These language models selected demonstrate LMGFuses generality concerning for to language model initializations.

V. Results and Analysis

A. Main Results

Our experimental results on CommonsenseQA and OpenBookQA datasets are presented in Table 1, respectively. From the table comparison results of the previous two datasets, our model has improved performance compared to the fine-tuned LM and the existing LM+KG model, on CommonsenseQA, compared to Roberta +5.7% and the previous best LM+KG model GREASELM compared to +0.2%, and on OpenBookQA, obvious with 7.6% higher than fine-tuned LMs

and 1.2% higher than LM+KG models. Improvements over QA-GNN and GREASELM show that our model LMGFuse outperforms the LM+KG approach in transferring information between text and KG representations. QA-GNN does not integrate continuous interactions between two modalities, and GREASELM uses more expressive labeled interactions to fuse interactions that limit the ability of GNNs to propagate information. The results on the OpenbookQA leaderboard are shown in Table 2. UnifiedQA (11B params) and T5 (3B) are about 30x and 8x larger than our model.

TABLE I: Evaluation of our models on CommonsenseQA and OpenBookQA datasets under the same random seed. For CommonsenseQA, as the official test is hidden, so here we report the in-house Test (IHtest) accuracy and use the same data set as [11], [14].

Methods	CommonsenseQA Acc.(%)	OpenBookQA Acc.(%)
RoBERTa-Large (w/o KG)	68.7	-
AristoRoBERTa (no KG)	-	78.4
RGCN [30]	68.4	74.6
GconAttn [20]	68.6	71.8
MHGRN [13]	71.1	80.6
QA-GNN [11]	73.4	82.8
GREASELM [14]	74.2	84.8
LMGFuse (Ours)	**74.4**	**86.0**

In addition, results on public datasets show that our model exhibits superior performance in modal fusion, for which we study performance on MedQA-USMLE datasets from other domains. Table 3 shows that our model also has a better performance compared to classical LM methods (e.g. SapBERT [29]) in the biomedical domain compared with QA-GNN and GREASELM, with a 4.3% improvement over fine-tuned LMs and a 2.5% improvement over the LM+KG model.

Stacked of Modal Fusion Layer We test the effect of the number of modal interaction layers on model performance. As shown in Figure 3, increasing the number of modal interaction layers continues to bring benefits until the number of layers N = 3, when N > 3, the performance begins to degrade. As the number of layers increases, the model changes from underfitting to overfitting.

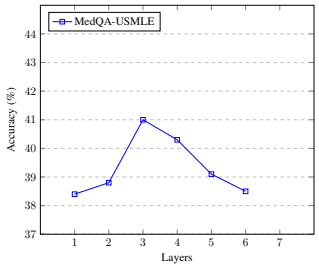

Fig. 3: The effect of layers on model performance

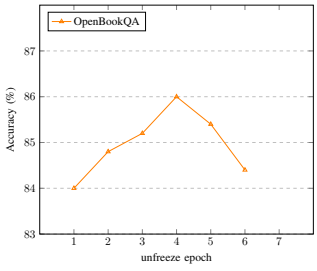

Fig. 4: The effect of unfreeze epoch on model performance

TABLE II: Test accuracy on OpenBookQA leaderboard

Methods	Acc.(%)
AristoRoBERTaV7	77.8
QAGNN-DeBERTa	79.0
QA-GNN [11]	82.8
GREASELM [14]	84.8
T5 11B + KB	85.4
JointLK [12]	85.6
UnifiedQA (11B) [4]	87.2
LMGFuse (Ours)	**86.0**

TABLE III: Performance on MedQA-USMLE.

Methods	Acc.(%)
BIOROBERTA-BASE	36.1
BIOBERT-LARGE	36.7
SapBERT-Base (w/o KG)	37.2
QA-GNN	38.0
GREASELM	38.5
LMGFuse (Ours)	**41.0**

unfreeze epoch We researched the unfreeze epoch, a hyperparameter that affects the parameter updates of the LM model during backpropagation. We found that freezing the LM parameters for a certain number of epochs can improve the performance of the model. As shown in Figure 4, on OpenBookQA, the performance of the model increases by about 2% when the unfreeze epoch is set from 0 to 4, and when it is set from 4 to 6, it decreases by about 1.6%.

Furthermore, we do not compare with models on higher leaderboards on OpenBookQA, such as unified QA [4], and Albert+DESC-KCR [31], because they either use stronger text encoders or use additional data resources, while our model focuses on improving joint reasoning between KG and LM.

B. Ablation studies

Through ablation experiments, we analyze the effectiveness of different model components on the MedQA-USMLE dataset, which includes rich background information. We evaluate the effect of fusing retrieval subgraphs with QA contextual information on model performance.

TABLE IV: The performance of LMGFuse with and without modal fusion on MedQA-USMLE.

Methods	Acc.(%)
GREASELM (No QA context)	38.5
GREASELM (Join QA context)	38.8
LMGFuse (No QA context)	39.3
LMGFuse (Join QA context)	**41.0**

Graph Construction with QA context We perform experiments to test whether adding QA context nodes to the model can improve the performance of the model under the condition that other environments such as random seeds are consistent. We do not consider splicing the QA context node with all

nodes at the beginning, because the subsequent modal fusion process is similar to this operation. The results in Table 4 show that graph reasoning with QA context nodes can bring certain improvements.

VI. Conclusion

In this paper, we propose the LMGFuse model, a new model that realizes the multi-level deep interaction and fusion of LM knowledge representations and GNN knowledge representations in a novel way. In this model, we design a deep interaction and fusion module, so that information can be transferred and updated between the two knowledge models. In addition, we also added the context node of the implicit knowledge generated from LM encoding in the construction of the reasoning subgraph in advance, so that the GNN can learn the implicit knowledge during the first message-passing process and enhance the reasoning ability of the GNN. We conduct experiments on multiple domains (commonsense and medical) datasets, and the results show that our model outperforms the previous KG+LM and LM-only baselines, demonstrating the models' generality with respect to language model initializations.

ACKNOWLEDGEMENTS

This work was in part supported by NSFC (Grant No.62176194), the Science and Technology Innovation 2030 (Grant No. 2022ZD0160604) Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No.2020KF0057) and Fundamental Research Funds for the Central Universities (WUT: 2021III054JC).

References

[1] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "Comet: Commonsense transformers for automatic knowledge graph construction," in *ACL*, 2019, pp. 4762–4779.

[2] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *AAAI*, 2017.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, vol. 21, pp. 1–67, 2020.

[4] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi, "Unifiedqa: Crossing format boundaries with a single qa system," in *EMNLP*, 2020, pp. 1896–1907.

[5] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[6] D. Yin, L. Dong, H. Cheng, X. Liu, K.-W. Chang, F. Wei, and J. Gao, "A survey of knowledge-intensive nlp with pre-trained language models," *arXiv preprint arXiv:2202.08772*, 2022.

[7] H. Ren, W. Hu, and J. Leskovec, "Query2box: Reasoning over knowledge graphs in vector space using box embeddings," in *ICLR*, 2020.

[8] H. Ren, H. Dai, B. Dai, X. Chen, M. Yasunaga, H. Sun, D. Schuurmans, J. Leskovec, and D. Zhou, "Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs," in *ICML*. PMLR, 2021, pp. 8959–8970.

[9] H. Ren and J. Leskovec, "Beta embeddings for multi-hop logical reasoning in knowledge graphs," *NeurIPS*, 2020.

[10] B. Y. Lin, X. Chen, J. Chen, and X. Ren, "Kagnet: Knowledge-aware graph networks for commonsense reasoning," in *EMNLP-IJCNLP*, 2019, pp. 2829–2839.

[11] M. Y. H. R. A. Bosselut and P. L. J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," *NAACL*, 2021.

[12] Y. Sun, Q. Shi, L. Qi, and Y. Zhang, "Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering," *arXiv preprint arXiv:2112.02732*, 2021.

[13] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren, "Scalable multi-hop relational reasoning for knowledge-aware question answering," in *EMNLP*, 2020, pp. 1295–1309.

[14] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. Manning, and J. Leskovec, "Greaselm: Graph reasoning enhanced language models for question answering," in *ICLR*, 2022.

[15] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," in *NAACL-HLT*, 2019, pp. 4149–4158.

[16] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering," in *EMNLP*, 2018, pp. 2381–2391.

[17] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," *Appl. Sci*, vol. 11, no. 14, p. 6421, 2021.

[18] S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao, "Answering natural language questions by subgraph matching over knowledge graphs," *ICDE*, vol. 30, no. 5, pp. 824–837, 2017.

[19] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, and Y. Su, "Beyond iid: three levels of generalization for question answering on knowledge bases," in *WWW*, 2021, pp. 3477–3488.

[20] X. Wang, P. Kapanipathi, R. Musa, M. Yu, K. Talamadupula, I. Abdelaziz, M. Chang, A. Fokoue, B. Makni, N. Mattei *et al.*, "Improving natural language inference using external knowledge in the science questions domain," in *AAAI*, vol. 33, no. 01, 2019, pp. 7208–7215.

[21] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, and S. Hu, "Graph-based reasoning over heterogeneous external knowledge for commonsense question answering," in *AAAI*, vol. 34, no. 05, 2020, pp. 8449–8456.

[22] J. D. Hwang, C. Bhagavatula, R. Le Bras, J. Da, K. Sakaguchi, A. Bosselut, and Y. Choi, "(comet-) atomic 2020: on symbolic and neural commonsense knowledge graphs," in *AAAI*, vol. 35, no. 7, 2021, pp. 6384–6392.

[23] T. Shen, Y. Mao, P. He, G. Long, A. Trischler, and W. Chen, "Exploiting structured knowledge in text via graph-guided representation learning," in *EMNLP*, 2020, pp. 8980–8994.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] P. Clark, O. Etzioni, T. Khot, D. Khashabi, B. Mishra, K. Richardson, A. Sabharwal, C. Schoenick, O. Tafjord, N. Tandon *et al.*, "From fto aon the ny regents science exams: An overview of the aristo project," *AI Magazine*, vol. 41, no. 4, pp. 39–53, 2020.

[26] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *ICLR*, 2018.

[28] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *ICLR*, 2019.

[29] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, and N. Collier, "Self-alignment pretraining for biomedical entity representations," in *NAACL-HLT*, 2021, pp. 4228–4238.

[30] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*. Springer, 2018, pp. 593–607.

[31] Y. Xu, C. Zhu, R. Xu, Y. Liu, M. Zeng, and X. Huang, "Fusing context into knowledge graph for commonsense question answering," in *ACL-IJCNLP*, 2021, pp. 1201–1207.